TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol. 17. No. 2, mayo-agosto, 2018, p. 71- 80 ISSN 1729-3804

MODELO PARA LA DETECCIÓN DE ATAQUES A LAS APLICACIONES WEB E INTERCAMBIO DE CIBERAMENAZAS

Ing. Dennis Barrera Pérez[1], MSc. Henry Raúl González Brito[2], Lic. Yailin Sánchez Borrell [3]

[1,2,3] Universidad de las Ciencias Informáticas (UCI), Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana, Cuba.
[1]dbperez@uci.cu , [2]henryraul@uci.cu, [3]ysanchezb@uci.cu

RESUMEN

En el trabajo se presentan los resultados de la implementación de un modelo para la detección y prevención de ciberataques en aplicaciones web a partir del análisis de patrones de comportamiento. Como componente fundamental del modelo se emplea el sistema SIEM Open Source Security Information Management (OSSIM) mediante una arquitectura distribuida y basada en sensores. Se describe el proceso de análisis de las diferentes fases de un ataque realizado a aplicaciones web y la toma de decisiones a partir de reglas de correlación lógica utilizando el sistema SIEM OSSIM. El modelo propuesto permite la creación de un entorno colaborativo para la detección y prevención de ataques a las aplicaciones web a través del intercambio de ciberamenazas. Como aporte fundamental de la investigación, se encuentra el incremento en la detección y prevención de ataques nuevos a las aplicaciones web con respecto a mecanismos tradicionales como Sistemas de Detectores de Intrusos (IDS) y Firewall de Aplicaciones Web (WAF). Los resultados de la aplicación del modelo en el entorno real de la infraestructura tecnológica de la Universidad de las Ciencias Informáticas, validan la pertinencia y factibilidad de este tipo de propuestas a nivel nacional.

PALABRAS CLAVES: ataques, aplicación web, intercambio de ciberamenazas.

MODEL FOR DETECTING ATTACKS ON WEB APPLICATIONS AND CYBER THREAT EXCHANGE

ABSTRACT

In the paper we present obtained results of a proposed model to detect and prevent cyber-attacks on web applications based on the analysis of behavioral patterns. Main component of the model is provided by the SIEM Open Source Security Information Management (OSSIM) system. This is used through a distributed and sensorbased architecture. It describes the process of analysis of different phases of attacks made to web applications and the decision making from logical correlation rules using the SIEM OSSIM system. The proposed model also allows the creation of a collaborative environment for the detection and prevention of attacks on web applications through the exchange of threats. As a fundamental contribution of research, there is an increase in the detection and prevention of new attacks on web applications with respect to traditional mechanisms such as Intrusion Detection Systems (IDS) and Web Application Firewall (WAF). The results of the application of the model in the real environment of the technological infrastructure of the University of Computer Science, validate the relevance and feasibility of this type of proposals to generalize their use on our country.

KEY WORDS: attack, web application, threat exchange.

1. INTRODUCCIÓN

La constante evolución de las Tics ha propiciado que las empresas quieran mantener su presencia en Internet a través de aplicaciones web. Algunos de sus principales usos son el comercio online, la prestación de servicios, las redes sociales y el intercambio de información en general. Por tal motivo las aplicaciones web se han

convertido en el punto de mira de los ciberdelincuentes, para el robo de información sensible, la propagación de malware, la afectación de la disponibilidad de los servicios, entre otros [1].
Un análisis del panorama actual del software, arroja como resultado que la implementación de medidas de seguridad mientras se desarrolla un software, no es económicamente factible, debido a los costos y los recursos necesarios, por lo cual un enfoque común es implementar una capa adicional encima de las aplicaciones web [2]. Uno de los mecanismos utilizados como parte de esta capa adicional son los Detectores de Intrusos de Red (NIDS) como es el caso de Snort [3].
Un NIDS se basa en la inspección de paquetes a nivel de red, usualmente utiliza dos enfoques para la detección de ataques, uno basado en firmas [3] y otro basado en anomalías. Sin embargo, cuando se trata de aplicaciones web, los mismos carecen de efectividad por las siguientes razones:

- Si el tráfico HTTP está encriptado, los NIDS no pueden inspeccionarlo.
- Están diseñados para operar sobre las Capas 3 y 4 del Modelo OSI, mientras que las aplicaciones web se encuentran en la Capa 7 (Aplicación).
- Los atacantes pueden utilizar técnicas de evasión de IDS como codificación HTTP, desfragmentación, etc.

Por otra parte, se encuentran los denominados Firewall de Aplicaciones Web (WAF). Los WAF actúan como intermediario entre la aplicación y el visitante que navega por un sitio web, interceptando y eliminando solicitudes maliciosas antes de que puedan causar daños [4]. Su principal desventaja radica en la alta tasa de falsos positivos.

2. IMPLEMENTACIÓN DEL MODELO

En el presente trabajo se muestran los resultados de la implementación de un modelo (denominado Aon) para la detección y prevención de ciberataques en aplicaciones web a partir del análisis de patrones de comportamiento. Dicho modelo emplea como componente principal, el sistema SIEM Open Source Security Information Management (OSSIM) mediante una arquitectura distribuida y basada en sensores [5]. Se describe el proceso de análisis de las diferentes fases de un ataque realizado a aplicaciones web y la toma de decisiones a partir de reglas de correlación lógica utilizando el sistema SIEM OSSIM. El modelo propuesto permite, además, la creación de un entorno colaborativo para la detección y prevención de ataques a las aplicaciones web a través del intercambio de amenazas.

Técnicas de protección contra ataques a las aplicaciones web existentes

Numerosas son las técnicas de protección de aplicaciones web propuesta por investigadores en los últimos años. Las mismas abarcan desde soluciones específicas para la protección contra ataques como: Inyección SQL, Cross Site Scripting (XSS), etc. [6], hasta soluciones genéricas para cualquier tipo de ataques, incluyendo técnicas basadas en ontología [7] y ML. Una propuesta novedosa de clasificación de los diferentes métodos de protección es realizada por V. Prhokhorenko et al. y se presenta en la siguiente tabla.

Tabla 1. Técnicas de protección de Aplicaciones Web. [2]

Table 1
Web application protection techniques classification criteria (Authors' compilation).

Decision base	A. Statistics (anomalies – probabilistic view)	B. Policy (protection developer/administrator knowledge or wishes-rule-based approaches)	C. Intent (application developer intentions)
Subject of observation	1. Inputs (HTTP GET/POST parameters, Cookies, Headers, URLs requested – user behaviour)	2. Application (source code and layout-white box approaches)	3. Outputs (generated page, database/network/file system activity – application behaviour)

A partir de la clasificación descrita anteriormente se concibió que el modelo Aon utilizara como Objetivo de observación un análisis de las Entradas (1) y las Salidas (3). La toma de decisiones se realiza mediante Políticas (B) basadas en reglas establecidas por los especialistas de seguridad e implementadas en el sistema SIEM OSSIM. Por lo cual una clase que define de manera general el comportamiento del modelo es {B 1 3}.

Arquitectura y componentes del modelo.

A continuación, se presenta la arquitectura general del modelo.

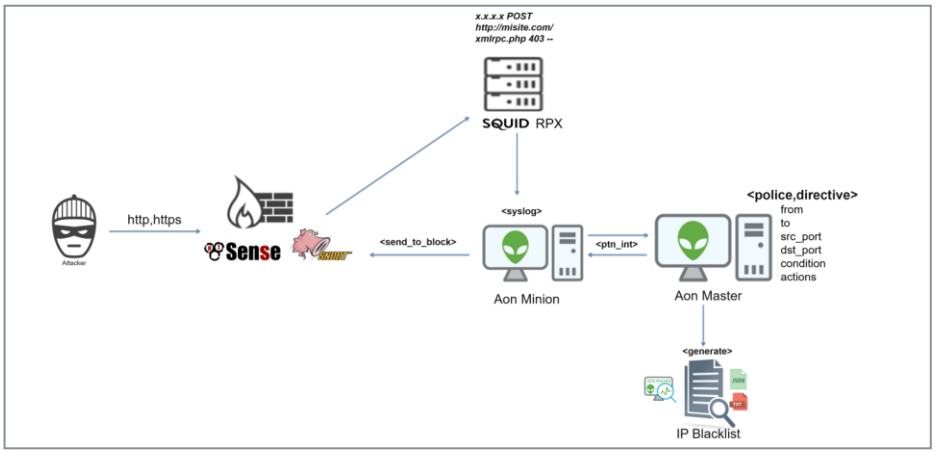

Figura 1. Arquitectura del Modelo Aon. Elaboración propia.

Aunque el esquema presentado está orientado a proteger múltiples aplicaciones web que se encuentran detrás de un Proxy Inverso, es importante destacar, que el modelo es adaptable a las diferentes fuentes de datos debido a su arquitectura orientada a plugins. Aon básicamente cuenta con cuatro componentes fundamentales: La fuente de datos (Logs), el Sensor de OSSIM (Aon Minion), el Mecanismo de Bloqueo (Snort + pfSense) y el Servidor de OSSIM (Aon Master).

Durante el estudio del arte sobre el tema en cuestión se observó que el análisis de logs para la detección de ataques a las aplicaciones web ha sido abordado en los últimos tiempos por varias investigaciones [9-12]. Por lo cual se considera que los logs del servidor web o proxy constituyen una gran fuente de información para ser incorporada al modelo. Las trazas seleccionadas como Fuente de Datos se encuentran en el formato (Apache Combined Log Format) [8] con los datos de las peticiones realizadas a las aplicaciones web. A continuación, se muestra un ejemplo con la explicación de cada uno de los parámetros:

X.X.X.X - - [17/Oct/2017:01:56:03 +0000] "GET http://my.app.com/24h/status.xhtml? HTTP/1.1" 200 4407 "http://my.app.com/category/android/page/3/" "Mozilla/5.0 (compatible; SemrushBot/1.2~bl; +http://www.semrush.com/bot.html)" TCP_MISS:FIRSTUP_PARENT

Tabla 2. Descripción detallada de los campos de una petición HTTP.

Parámetro	**Descripción**
X.X.X.X	Dirección IP del cliente que realiza la petición.
[17/Oct/2017:01:56:03 +0000]	La fecha en que el servidor terminó de procesar la información.
"GET http://my.app.com/24h/status.xhtml? HTTP/1.1"	La petición realizada por el cliente comenzando por el método HTTP utilizado.
200	El código de estado que el servidor devuelve al cliente.
4407	El tamaño del objeto enviado al cliente sin incluir la cabecera de la respuesta.
"http://my.app.com/category/android/page/3/"	El “Referer”
"Mozilla/5.0 (compatible; SemrushBot/1.2~bl; +http://www.semrush.c om/bot.html)	El user-agent de la cabecera HTTP

Las trazas generadas a partir de las peticiones son enviadas en tiempo real al Sensor (Aon Minion), el cual se encarga de aplicarle las reglas correspondientes a partir de los patrones de comportamiento anómalo definidos. Una vez identificado un ataque, el sistema automáticamente selecciona el IP identificado como atacante y lo envía para que sea bloqueado.

Por otra parte, se encuentra el sistema Snort + pfSense, el cual se encarga de bloquear el IP recibido por un período definido por los especialistas de seguridad. Para el análisis de la gran diversidad de ataques que se pueden presentar, se realizó una clasificación de tres fases fundamentales. Dicha clasificación parte del framework creado por Lockheed Martin denominado Cyber Kill Chain. El mismo brinda visibilidad ante un ataque y permite enriquecer el entendimiento de un analista de las técnicas y procedimientos de un ciberatacante para complementar sus objetivos [13]:

Fase 1: Reconocimiento. Está fase está compuesta por las acciones de un atacante para obtener la información necesaria antes de lanzar su ataque. Incluye el uso de herramientas para la realización de escaneos de vulnerabilidades como OWASP ZAP, Nikto y wpScan. Además, comprende un grupo de acciones de exploración como la búsqueda de ficheros específicos en el servidor web que brinden información sobre las versiones de las tecnologías utilizadas, la estructura de sitio web, etc. Ejemplo de los ficheros a explorar son: Readme.txt, Robots.txt, Install.php y readme.html.

Fase 2: Explotación: Constituye el ataque en sí, parte de los resultados de la fase anterior para la explotación de una vulnerabilidad del sistema objetivo. Se puede realizar de forma manual o utilizando alguna herramienta como Metasploit o Sqlmap. Dentro de los principales ataques a las aplicaciones web se encuentran los ataques de XSS, Inyección SQL, RFI, LFI, CSRF, un listado completo se puede observar en la Guía de OWASP v.4. [14]

Fase 3: Acciones sobre el Objetivo: Una vez comprometida una aplicación web o servidor, el atacante puede comenzar a realizar un conjunto de acciones dentro de las que se destaca: implantación de puerta trasera para el control del servidor, desfiguración de la página principal del sitio web o los sitios alojados en el servidor, subida de ficheros con código malicioso (webshell, malware, etc.), uso del servidor como parte de una botnet para la ejecución de Ataques de Denegación de Servicio Distribuidos (DDOS), entre otros.
Para cada una de estas fases se establecieron un conjunto de patrones, que permiten detectar la existencia de comportamiento anómalo y generar un Evento de Seguridad. El manejo de estos eventos es realizado por el Servidor (Aon Master) que es otro de los componentes fundamentales del modelo propuesto. Este componente es el encargado de realizar la correlación lógica para el análisis de los eventos recibidos del Sensor y emitir alertas en correspondencia con el nivel de gravedad del evento. Algunas de las funciones principales son:

- Realizar un seguimiento de los IP detectados y clasificados dentro de la fase de Reconocimiento.
- Hacer un seguimiento a los ataques compuestos por varias peticiones semejantes para lograr su objetivo, como son los ataques de Fuerza Bruta (ver Figura 2) y Denegación de Servicio (DOS).
- Permite ejecutar un conjunto de acciones a partir de los eventos generados como son el bloqueo automático y la notificación por correo.
- Generar una Lista Negra de los IP detectados como atacantes lo cual constituye la reputación.
- Generar en una estructura de syslog un registro histórico de cada ataque realizado, efectivo o no.
- Creación de reportes ejecutivos personalizables.
- Correlación lógica para la detección de ataques, ver Figura 3.

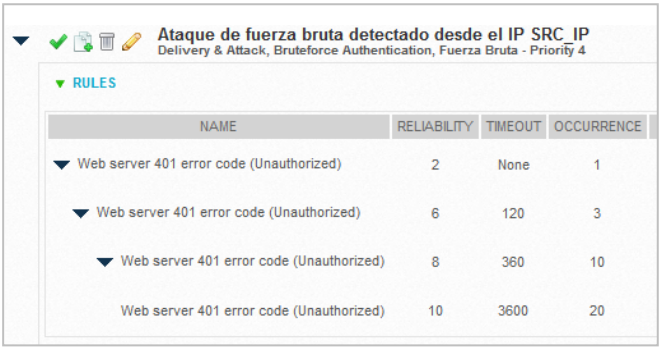

Figura 2. Mecanismo de correlación para un ataque de fuerza bruta.

Figura 3. Conjunto de directivas para dar seguimiento a los eventos.

Por lo planteado anteriormente, es considerado por los autores que dichas funcionalidades, brindan una ventaja significativa con respecto a los mecanismos de protección como NIDS y WAF.

Ejemplo de funcionamiento del Sistema Aon

Cuando se realiza una petición mediante el protocolo HTTP o HTTPS hacia alguna de las aplicaciones web a proteger, la misma es filtrada por el proxy inverso y una traza en formato Apache Combine Log es generada. El sistema Aon Minion se encarga de procesar la traza generada y aplicarle una regla, en correspondencia con los patrones definidos. En caso de coincidencia con alguna de las fases definidas se ejecuta la acción de seguimiento o bloqueo automático en dependencia del tipo de ataque. Seguidamente se genera una traza con los detalles del ataque y el IP detectado se agrega a la Lista Negra formando parte del Intercambio de Amenazas.

Implementación del modelo en un entorno real

En este epígrafe se presenta el resultado de la implementación del modelo en la Universidad de las Ciencias Informáticas. Para ello se muestran algunas de las vulnerabilidades recientes a las aplicaciones web que han sido objeto de ataque por los ciberdelincuentes y un ejemplo de cómo el modelo fue capaz de detectarlas.

Vulnerabilidad 0-Day en Joomla

En diciembre del 2015 se publicó una vulnerabilidad de ejecución remota de comandos que afectaba a todas las versiones de Joomla desde 1.5 hasta 3.4. Inmediatamente se emitió el parche de seguridad que corregía dicha vulnerabilidad. Sin embargo, no pasó mucho tiempo para que fuera explotada por los ciberdelincuentes [15] y [16]. La forma de explotación está compuesta por peticiones HTTP del tipo:

X.X.X.X – – [12/Dec/2015:16:49:40 -0500] "GET /contact/ HTTP/1.1" 403 5322 "http://google.com/" "}__test|O:21:\x22JDatabaseDriverMysqli\x22:3:..{s:2:\x22fc\x22;O:17:\x22JSimplepieFactory\x22:0:{}s:21:\x22\x5C0\x5C0\x5C0disconnectHandlers\5:..{s:8:\x22sanitize\x22;O:20:\x22JDatabaseDriverMysql\x22:0:{}s:8:\x22feed_url\x22…

Como se puede observar, en el campo del User-agent se muestra el código malicioso a ejecutar. El resultado del modelo AON para dicha vulnerabilidad es el que se muestra a en las siguientes figuras:

Figura 4. Detección de la vulnerabilidad 0-day para Joomla

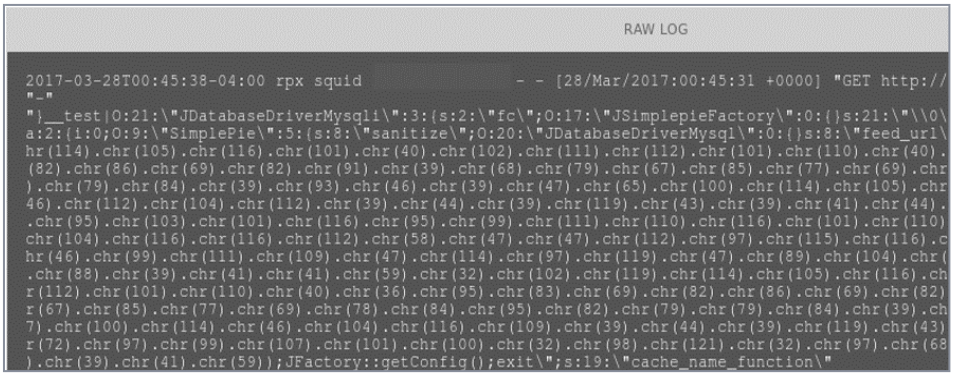

Figura 5. Detalles de la petición realizada.

Vulnerabilidad en el Plugin Revolution Slider de Wordpress

En agosto del 2014 se emitió una alerta sobre la existencia de una vulnerabilidad en el plugin Revolution Slider el cual viene incluido en varios temas comerciales de CMS Wordpress. La vulnerabilidad fue clasificada como Inclusión de Ficheros Locales (LFI) y permitía acceder a cualquier fichero alojado en el servidor web. [17], [18]. Inmediatamente se desató una campaña de ataques, dentro de los sitios web afectados por los ciberdelincuentes se encontraba el sitio del bufete de abogados panameño Mossack Fonseca (MF). Dicho sitio poseía la versión vulnerable del plugin Revolution Slider (2.1.7) y al ser comprometido propició la mayor violación de datos a periodistas en la historia, con un peso de 2,6 terabytes y 11,5 millones de documentos. Esta brecha es conocida como La Brecha de Panamá Papers [19].

Según la fuente Sucuri [20], el método de explotación se basaba en una petición con la siguiente estructura:

http://victima.com/wp-admin/adminajax.php?action=revslider_show_image&img=../wpconfig.php

Una estructura similar fue detectada por el modelo Aon y clasificada como Reconocimiento, en la cual se intentaba acceder a la versión del plugin Revolution Slider.

Figura 6. Captura de la petición HTTP en la fase de Reconocimiento.

Vulnerabilidad en el API-REST de Wordpress 4.x (CVE-2017-5487)

El 1ro de febrero del 2017, la empresa Sucuri emitió una alerta sobre una vulnerabilidad grave (9/10, impacto muy alto) de inyección de contenido (escalamiento de privilegios) que afecta a la API REST de Wordpress.

Esta vulnerabilidad permite a un usuario no autenticado modificar el contenido de cualquier post o página de un sitio web realizado con este CMS. Dicha vulnerabilidad fue reportada al equipo de seguridad de Wordpress y el parche fue agregado de forma silenciosa en la versión 4.7.2. [21]
Aun así, se desató una campaña global de defacement arrojando como resultado un gran número de sitios web comprometidos y desfigurados [21]. La campaña 1 contaba con más de 66.000 páginas comprometidas por el ciberdelincuente o grupo denominado w4l3XzY3. La segunda campaña no tuvo tanto éxito, ya que Google solo mostraba 500 páginas comprometidas por el grupo de ciberdelincuentes denominado Cyb3r-Shia. Un ejemplo de una petición intentando explotar dicha vulnerabilidad se muestra a continuación:

Figura 7. Estructura de una petición HTTP con la firma del ciberdelincuente MuhmadEmad

En la siguiente figura se presenta un fragmento de la alerta emitida por Aon para dicho ataque:

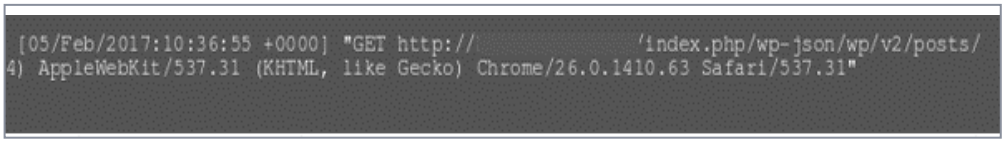

Figura 8: Acceso al recurso wp-json/wp/v2/posts en la fase de Reconocimiento.

Para cada una de estos ataques que fueron detectados en la fase de Reconocimiento, el modelo emitió una alerta y el IP detectado fue bloqueado automáticamente como medida preventiva.

Análisis comparativo

A continuación, se muestra un análisis comparativo entre el modelo propuesto y varios mecanismos existentes. El criterio a evaluar es la efectividad en la detección de cuatro vulnerabilidades conocidas y que fueron descritas anteriormente.

Tabla 3. Análisis comparativo entre Aon y otros mecanismos existentes.

Vulnerabilidad	WAF Mod Security	Nids Snort	Modelo Aon
Vulnerabilidad 0 Day en Joomla CV E-2015-8562	-	X	X
Vulnerabilidad en el Plugin Revolution Slider de WordPress	-	X	X
Vulnerabilidad en el API-REST de Wordpress 4.x	-	-	X
Vulnerabilidad de Inclusión de Ficheros en OJS 6.0	-	-	X

Intercambio de amenazas

Uno de los principales retos actuales de la ciberseguridad es el intercambio de información de ciberamenazas. Su objetivo principal es la recopilación, el almacenamiento y la distribución de la información necesaria para actuar de forma homogénea, rápida y eficaz contra las ciberamenazas, generando un conocimiento común y compartido. Existes varios estándares que implementan estos conceptos, algunos ejemplos son OpenIOC, CybOX, STIX y TAXII [22]. El modelo descrito propone un mecanismo basado en estos estándares que

permita la creación de un entorno colaborativo para intercambio de información de ciberamenzas a nivel nacional. Dicho mecanismo tiene como objetivo la detección temprana de ataques, así como el mejoramiento de la respuesta ante incidentes.

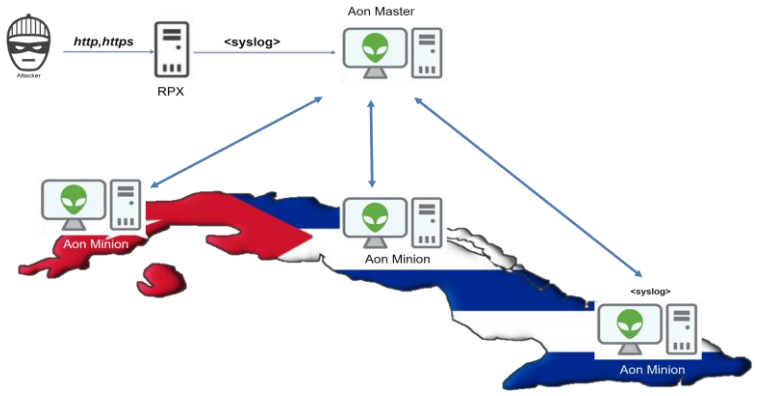

Figura 9: Entorno colaborativo para el intercambio de amenazas.

3. CONCLUSIONES

Debido al gran auge que ha tenido el uso de las aplicaciones web en los últimos tiempos, es notable que se hace necesario la existencia de técnicas de protección cada vez más efectivas. Mecanismos como el propuesto en el presente trabajo persiguen como objetivo la detección y prevención de los ataques a las aplicaciones web en sus fases más tempranas. Es notable que los ciberdelincuentes usan cada vez técnicas más sofisticadas para encubrir sus ataques, por lo cual, una importante línea a investigar lo constituye la incorporación de técnicas de Inteligencia Artificial al modelo propuesto, para el mejoramiento de la detección de los ciberataques. La creciente avalancha de ciberataques demuestra la necesidad de mecanismos que permitan el intercambio de información de ciberamenazas, por lo cual la implementación de los mismos constituye un paso de avance para la detección de ataques y rapidez en la respuesta a incidentes. Se considera que los resultados de la aplicación del modelo en el entorno real de la infraestructura tecnológica de la Universidad de las Ciencias Informáticas, validan la pertinencia y factibilidad de este tipo de propuestas a nivel nacional.

RECONOCIMIENTOS

El presente trabajo constituye un resultado importante de la labor realizada en la Dirección de Seguridad Informática de la Universidad de las Ciencias Informáticas. Los autores desean agradecer especialmente a los colegas Msc. Ramón Alexander Angla Martínez y al Ing. Yaisel Hurtado González que formaron parte del equipo de trabajo y que contribuyeron significativamente en el desarrollo e implementación del modelo propuesto.

REFERENCIAS

[1] Huang, C., et al., A study on Web security incidents in China by analyzing vulnerability disclosure platforms. Computers & Security, 2016. 58: p. 47-62.
[2] PROKHORENKO, Victor; CHOO, Kim-Kwang Raymond; ASHMAN, Helen. Web application protection techniques: A taxonomy. Journal of Network and Computer Applications, 2016, vol. 60, p. 95-112.
[3] Snort - Network Intrusion Detection. [en línea]. Disponible en: https://www.snort.org/. [Consulta: 19 junio 2017].
[4] Web Application Firewall - OWASP. [en línea]. [Consulta: 19 mayo 2017]. Disponible en: https://www.owasp.org/index.php/Web_Applica tion_Firewall.
[5] "OSSIM: The Open Source SIEM | AlienVault." [en línea]. Disponible en: https://www.alienvault.com/products/ossim. [Consulta: 20 mayo 2017].
[6] G.Swapna, R.Pavani Srivatsav. Securing Web Applications By Analyzing The Logs Of The Database Server Or Web Server/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.comVol. 2, Issue 6, November - December 2012, pp.432-435

[7] RAZZAQ, Abdul, et al. Ontology for attack detection: An intelligent approach to web application security. computers & security, 2014, vol. 45, p. 124-146.
[8] Apache HTTP Server, (2007). Combined Log Format. [en línea]. Disponible en: http://httpd.apache.org/docs/1.3/logs.html#combined. [Consulta: 02 junio 2017].
[9] Kent, C., M. Tanner, and S. Kabanda. How South African SMEs address cyber security: The case of web server logs and intrusion detection. in Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech), IEEE International Conference on. 2016. IEEE.
[10] Müller, Jens, Jörg Schwenk, and Ing Mario Heiderich. "Web Application Forensics." (2012).
[11] Roger Meyer and Carlos Cid (advisor). Detecting Attacks on Web Applications from Log Files. In Information Security Reading Room, January 2008. SANSInstitute 2008, As part of the Information Security Reading Room.
[12] PRODROMOU, A., 2016. Using logs to investigate a web application attack. Acunetix [en línea]. Disponible en: http://www.acunetix.com/blog/articles/usinglogs-to-investigate-a-web-application-attack/. [Consulta: 07 junio 2017].
[13] Cyber Solutions · Lockheed Martin. [en línea]. Disponible en: http://www.lockheedmartin.com/us/what-wedo/aerospace-defense/cyber.html. [Consulta: 12 junio 2017].
[14] OWASP Testing Project - OWASP. [en línea]. Disponible en: https://www.owasp.org/index.php/OWASP_Tes ting_Project. [Consulta: 20 junio 2017].
[15] Critical 0-day Remote Command Execution Vulnerability in Joomla. Sucuri Blog [en línea], 2015. Disponible en: https://blog.sucuri.net/2015/12/remotecommand-execution-vulnerability-injoomla.html. [Consulta: 20 junio 2017].
[16] Hackers actively exploit critical vulnerability in sites running Joomla | Ars Technica. [en línea Disponible en: https://arstechnica.com/security/2015/12/hackers-actively-exploit-critical-vulnerability-in-sitesrunning-joomla/], [Consulta: 20 junio 2017].
[17] Vulnerability in Slider Revolution Responsive plugin for WordPress Could Allow for ArbitraryFile Download | New York State Office of Information Technology Services. [en línea], Disponible en: https://its.ny.gov/security-advisory/vulnerabilityslider-revolution-responsive-plugin-wordpresscould-allow-arbitrary. [Consulta: 20 junio 2017].
[18] CVE-2014-9735: The ThemePunch Slider Revolution (revslider) plugin before 3.0.96 for WordPress. [en línea]. [Consulta: 20 junio 2017]. Disponible en: http://www.cvedetails.com/cve/CVE-20149735/.
[19] Mossack Fonseca Breach - WordPress Revolution Slider. Wordfence [en línea], 2016. Disponible en: https://www.wordfence.com/blog/2016/04/moss ack-fonseca-breach-vulnerable-sliderrevolution/. [Consulta: 20 junio 2017].
[20] Sucuri Security. Sucuri Security [en línea]. Disponible en: https://sucuri.net. [Consulta: 20 junio 2017].
[21] Content Injection Vulnerability in WordPress. Sucuri Blog [en línea], 2015. Disponible en: https://blog.sucuri.net/2017/02/contentinjection-vulnerability-wordpress-rest-api.html, [Consulta: 20 junio 2017].
[22] FERNÁNDEZ, Miguel Angel Rego; GARCÍA, Pedro Pérez. El intercambio de información de ciberamenazas. Cuadernos de estrategia, 2017, no 185, p. 139-17

SOBRE LOS AUTORES

Ing. Dennis Barrera Pérez (dbperez@uci.cu). Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2014. Ha presentado varios trabajos relacionados con temas de Seguridad Informática y Ciberseguridad en varios eventos nacionales e internacionales. Posee una certificación de Security Essential WhiteHat otorgada por la Corporación Bkav Internet Security de Vietnam. Posee la Categoría Docente de Instructor y presta servicios docentes en la carrera de ciclo corto Administración de Redes y Seguridad Informática. Es profesor del curso de Postgrado de Automatización de Controles de Seguridad Informática perteneciente a la Escuela de Invierno y Verano de la Universidad de las Ciencias Informáticas. Actualmente se encuentra cursando la Especialidad de Seguridad Informática y se desempeña como Especialista Superior y Jefe del Grupo de Ciberseguridad de la Dirección de Seguridad Informática de la UCI.

MSC. Henry Raúl González Brito: Graduado de Ingeniería Informática por la Universidad de Camagüey y la CUJAE en el 2005. Es Máster en Gestión de Proyectos Informáticos. Es coordinador de la Especialidad de Posgrado en Seguridad Informática. Es profesor del curso de postgrado de Pruebas de Penetración en Aplicaciones Web que se ha impartido a especialistas de diversas instituciones del país y ha contado con participación internacional. Es coordinador del Grupo de Investigación de Seguridad Informática de la UCI.

Editor de un blog especializado en Seguridad en Informática (https://henryraul.WordPress.com). Miembro del comité académico de la carrera de Administración de Redes y Seguridad Informática. Participante en cursos de diversas maestrías y especialidades impartiendo temáticas de Seguridad Informática. Ha impartido conferencias en eventos nacionales e internacionales en cuba sobre la seguridad en aplicaciones web.

Lic. Yailin Sánchez Borrell (ysanchezb@uci.cu). Graduada de Ciencia de la Computación en la Universidad de Oriente en 2013. Ha presentado varios trabajos en eventos relacionados con la Seguridad Informática. Actualmente se desempeña como Especialista General en la Dirección de Seguridad Informática y se dedica a la gestión y respuesta a incidentes en el Grupo de Gestión de Incidentes de la Dirección de Seguridad Informática de la UCI. Posee un amplio dominio del trabajo con el Sistema de Gestor de Incidentes (OTRS), Sistema de gestión de reportes de la DSI (Libélula) y la investigación de incidentes mediante el sistema SIEM OSSIM. Actualmente se encuentra cursando la Especialidad de Seguridad Informática.