https://doi.org/10.37815/rte.v34n3.955
Artículos originales

Análisis comparativo de Sistemas de Detección de Intrusos (IDS) en entornos universitarios

Comparative analysis of Intrusion Detection Systems (IDS) in university environments

Alex Caizapanta González[1] https://orcid.org/0000-0001-8112-7086

[1]*Universidad Central del Ecuador*, Quito, Ecuador
aecaizapanta@uce.edu.ec

Esta obra está bajo una licencia internacional
Creative Commons Atribución-NoComercial 4.0.

Enviado: 2022/07/03
Aceptado: 2022/09/16
Publicado: 2022/11/30

Resumen

En la actualidad, los diferentes ataques de tipo cibernético han obligado a las organizaciones a buscar diversos equipos que las protejan, sin embargo, los patrones de ataque no son iguales en todas las organizaciones ya que su núcleo de trabajo no es el mismo. Por lo que, en este trabajo se propone un análisis de los distintos patrones de ataque más recurrentes en entornos universitarios, basándose en una comparativa de las funcionalidades y resultados obtenidos mediante tres tipos de Sistemas de Detección de Intrusos (IDS), dos de ellos utilizando software libre y el otro licenciado, así también, efectuar ataques a los mismos en un entorno controlado. Este artículo contiene los siguientes apartados: introducción, metodología de trabajo, resultados – discusión y conclusiones. Con el análisis comparativo se espera obtener estadísticas acerca de los patrones de ataque más críticos a los que se enfrenta este tipo de ambiente, para concluir se emiten las directrices de configuraciones más óptimas para mitigar dichos patrones de ataque.

Palabras clave: Ataque cibernético, entorno controlado, patrones de ataque, software libre, vulnerabilidad.

Abstract

Nowadays, the different cyber-attacks have forced organizations to find different equipment to protect them; however, the attack patterns are not the same in all organizations since their core work is not identical. Therefore, this paper proposes an analysis of the most recurrent attack patterns in university environments based on a comparison of the functionalities and results

Como citar: Caizapanta, A. (2022). Análisis comparativo de Sistemas de Detección de Intrusos (IDS) en entornos universitarios. *Revista Tecnológica - Espol, 34(3), 118-138.*
http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/955

obtained by three types of Intrusion Detection Systems (IDS): two of them using free software and the third one using licensed software, as well as to carry out attacks on them in a controlled environment. This article contains the following sections: introduction, methodology, results, discussion, and conclusions. With the comparative analysis, it is expected to obtain statistics about the most critical attack patterns faced by this type of environment. Finally, guidelines are issued for the most optimal configurations to mitigate such attack patterns.

Keywords: Cyber-attack, controlled environment, attack patterns, free software, vulnerability.

Introducción

En la actualidad todo tipo de organización posee infraestructura de servidores y servicios que se hallan publicados en internet, por lo que, existe innumerable información confidencial que se encuentra expuesta a potenciales ataques cibernéticos. Debido a esto, en las organizaciones se deben tomar las respectivas medidas a nivel tecnológico y de gestión.

Una de las bases fundamentales en la protección del perímetro de una infraestructura de red de una organización son los Sistemas de Detección de Intrusiones IDS (por sus siglas en ingles), los cuales son sistemas encargados de detectar y bloquear ataques cibernéticos mediante la ejecución de un análisis de tráfico en la red y comparándolo con firmas de ataques conocidos o comportamiento sospechoso. Por esta razón, en este trabajo se propone verificar cuáles son los patrones más comunes de ataques a infraestructuras de servidores y servicios en entornos universitarios utilizando tres IDS.

El interés de aplicar este análisis dentro de entornos universitarios se debe a la diversidad de perfiles de usuarios de red existentes dentro de estos ambientes, ya que se pueden encontrar desde artistas, pasando por filósofos hasta tener expertos en informática, donde cada perfil tiene sus propias tendencias de navegación, aplicativos, software utilitario, entre otros. Cada una de estas tendencias puede traer riesgos detrás de dichas conexiones.

Por lo antes mencionado se debe tomar en cuenta que en las universidades tanto servidores como servicios ofrecidos están expuestos a ataques internos y externos a la infraestructura de red, principalmente debido a la diversidad de dispositivos móviles de docentes, estudiantes y trabajadores.

La importancia de este proyecto se basa en proponer y validar herramientas y configuraciones óptimas en las instituciones universitarias para la prevención de ataques y la protección de las infraestructuras de red, servidores y servicios críticos como los son las bases de datos, los sistemas académicos, la información relacionada con investigaciones y publicaciones, entre otros.

Para la ejecución del prototipo experimental en un entorno emulado se toma como base la infraestructura de red, servicios y servidores de la Universidad Central del Ecuador campus principal ubicado en la ciudad de Quito, Ecuador. En este lugar, conviven diariamente un aproximado de 45000 personas entre docentes, estudiantes y personal administrativo lo que proporciona una gran variedad de dispositivos conectados a la red, la mayoría de ellos no pertenecen a la institución y, por lo tanto, existe una alta probabilidad de sufrir ataques a la infraestructura tecnológica.

Para la implementación del prototipo se ejecuta un análisis de la topología de red existente en la Universidad, posteriormente se procede a la revisión de los logs y patrones de ataque proporcionados por el módulo de IPS del equipo de seguridad perimetral instalado en

dicha institución. Con ello se pretende reconocer a qué criterios se rigen estos patrones de ataques y a base de esto examinar las reglas configuradas actualmente en el equipo antes mencionado.

Para la aplicación del trabajo se tiene contemplado la ejecución de pruebas de concepto con dos Sistemas de Detección de Intrusiones de Red (NIDS) mediante software libre como lo son Snort y Suricata haciendo uso de un entorno emulado en laboratorio. En el momento que la topología se encuentre implementada se ejecutarán diversas pruebas de evasión de IDS utilizando herramientas basadas en la distribución Kali Linux con el objeto de evaluar las soluciones antes mencionadas. Con cada una de estas herramientas se realiza el respectivo análisis de los logs generados y los patrones de ataque recolectados, para luego ser comparados con los resultados anteriormente obtenidos con el IDS existente en la infraestructura universitaria. Con este procedimiento se mide la efectividad en la detección de patrones de ataques y la respuesta de dichos sistemas a los mismos.

Finalmente, se ejecuta un análisis comparativo de los resultados obtenidos para determinar las debilidades y fortalezas de cada sistema, a base de esto se realiza las recomendaciones para mejorar las reglas del equipo en producción, a fin de tener un patrón de configuración basado en las mejores prácticas de implementación de este tipo de plataformas.

El objetivo general de la investigación es realizar un análisis exhaustivo de los distintos patrones de ataque que se presentan en las universidades mediante la utilización de sistemas de detección de intrusiones en software libre y licenciados. Con estos resultados, emitir recomendaciones para la configuración de reglas apropiadas dentro de este tipo de entornos.

Los objetivos específicos del proyecto son:

- Analizar los patrones de ataques que se producen en la infraestructura de red, servicios y servidores en un entorno universitario mediante los logs registrados por el IDS existente.
- Establecer un ambiente emulado para la ejecución de pruebas experimentales con los NIDS de software libre planteados.
- Identificar los tipos de patrones de ataques captados por los sistemas de detección de intrusiones IDS experimentales previamente implementados en el entorno emulado.
- Comparar los resultados obtenidos en los diferentes sistemas probados y obtener estadísticas acerca del comportamiento de los patrones de ataques en este tipo de entornos.
- Establecer un conjunto de buenas prácticas de configuración que optimice el funcionamiento de los sistemas de detección de intrusiones IDS dentro de entornos universitarios.

El artículo cuenta con tres secciones: metodología de trabajo, resultados – discusión y conclusiones.

Metodología de Trabajo

El presente trabajo es un proyecto basado en un prototipo experimental el cual se ha dividido en diferentes fases para alcanzar los objetivos previamente planteados.

Como primera fase se realizó una revisión de la topología de red perteneciente a la Universidad Central del Ecuador donde se identificó cada uno de sus equipos, segmentos de red, servidores y seguridad perimetral. Específicamente se identificó dentro del equipo de seguridad perimetral dónde se encuentra y cómo funciona el módulo IDS del mismo, se obtuvo los logs generados por este módulo en un determinado periodo de tiempo para ser analizados de manera exhaustiva y así determinar los patrones de ataque que afectan a la infraestructura universitaria y clasificarlos en una escala cuantitativa.

Como segunda fase y basados en el análisis de la infraestructura existente en la institución, se implementó el piloto experimental en un entorno emulado, dividido en dos etapas. En la primera etapa se utilizó Snort, el cual es un IDS open source que funciona en modo de NIDS, el mismo que obtuvo un conjunto de logs con patrones de ataque evaluados en un periodo de tiempo determinado. La segunda etapa es la ejecución del mismo conjunto de pruebas pero utilizando Suricata como NIDS, obteniendo un nuevo conjunto de logs en su análisis. Estos resultados serán utilizados para ejecutar un estudio de la efectividad en respuesta de detección de amenazas de las tres plataformas puestas en prueba. Para esta fase se realizaron ataques controlados para efectuar un análisis de vulnerabilidades en el entorno antes mencionado, en la ejecución de los ataques programados se utilizaron herramientas que forman parte de la distribución Kali Linux.

En la tercera fase se realizó el análisis comparativo del comportamiento de los IDS antes descritos, destacando el nivel de detección de ataques en cada uno de ellos. Con estos resultados se emiten buenas prácticas de configuración para la implementación de NIDS en instituciones universitarias las cuales constan en este artículo.

El diseño de la investigación es experimental, transversal, cuantitativa, de laboratorio y correlacional.

Este proyecto es experimental debido a que el investigador incluye en su estudio una comparativa de los IDS en Entornos Universitarios, en el cual la variable independiente de estudio fue manipulada, ya que, se emularon varias pruebas de concepto para validar la eficiencia en la detección de amenazas por parte de IDS basados en software libre.

Experimento: Situación de control en la cual se manipulan, de manera intencional, una o más variables independientes (causas) para analizar las consecuencias de tal manipulación sobre una o más variables dependientes (efectos) (Hernandez et al., 2010).

Es transversal debido a que la recolección de datos se da en tiempo establecido, en este caso, el proyecto se empezó a ejecutar con datos obtenidos de enero a junio de 2020, como lo señala (Hernández et al., 2010) “Diseños transeccionales (transversales): Investigaciones que recopilan datos en un momento único”.

Como lo menciona (Hernández et al., 2010) “Enfoque cuantitativo: Usa la recolección de datos para probar hipótesis, con base en la medición numérica y el análisis estadístico, para establecer patrones de comportamiento y probar teorías”. Este proyecto tiene un enfoque cuantitativo ya que se obtuvo datos estadísticos de las principales y más frecuentes amenazas a las que está expuesta la infraestructura de red de la UCE[1].

Fue en un contexto de laboratorio al controlar con mayor rigurosidad el ambiente reducido. Inicialmente estaba prevista que sea de campo al ser aplicada en la UCE; pero, a

[1] UCE: Universidad Central del Ecuador

causa de la pandemia de Covid-19, se debió emular el entorno, como lo menciona (Hernández et al., 2010) “Contexto de laboratorio: Experimento en que el efecto de todas o casi todas las variables independientes influyentes no concernientes al problema de investigación se mantiene reducido lo más posible”.

Es correlacional de acuerdo a lo señalado por (Hernández et al., 2010) “Investigación correlacional: Asocia variables mediante un patrón predecible para un grupo o población”, debido a que las variables de investigación mejoran los lineamientos de seguridad en los entornos universitarios, para conocer su relación en un contexto específico, midiendo la variable independiente cuantificándolas y analizando su relación, las mismas que fueron sometidas a pruebas.

Resultados y Discusión

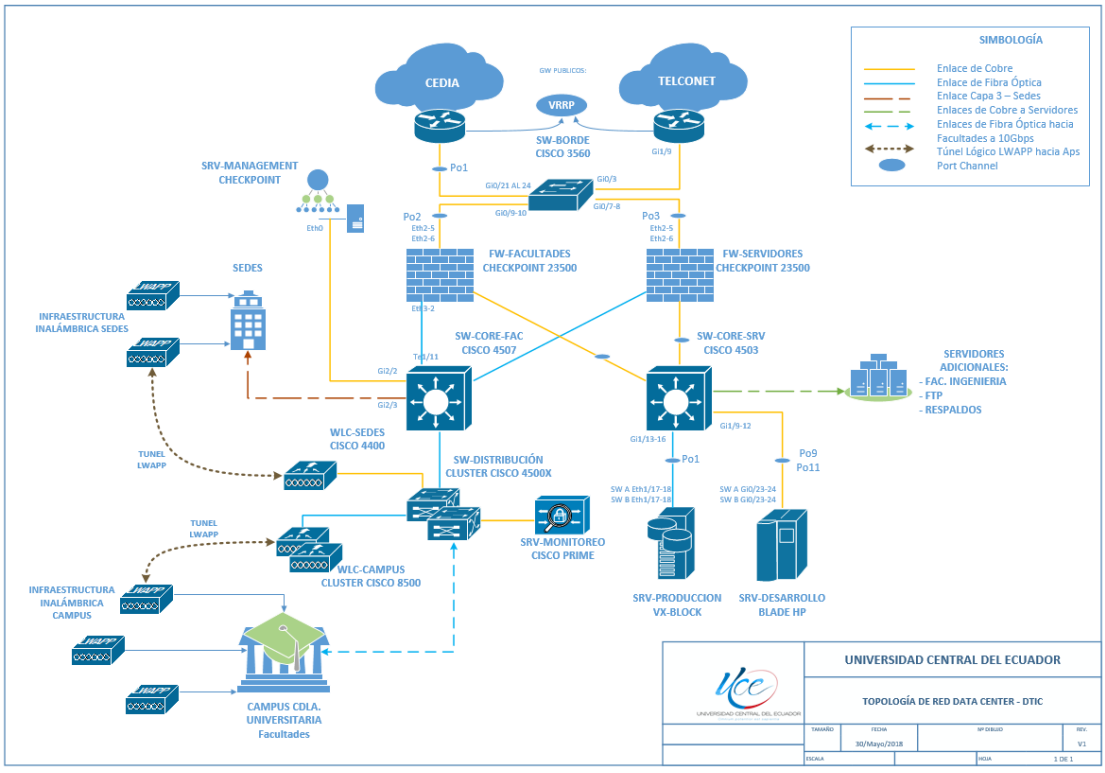
Análisis de la Topología de Red Existente

En la red de la UCE conviven aproximadamente 45000 personas entre estudiantes, docentes y personal administrativo, quienes utilizan recursos de red, servicios y ancho de banda, por lo que, también resulta un escenario atractivo para un atacante debido a la variedad de conexiones entrantes y salientes desde esta infraestructura.

Como se puede observar en la Figura 1 la topología de red de la institución posee un esquema de conexión redundante hacia dos proveedores de servicio. En el borde de la red se encuentran dos equipos de seguridad perimetral de la marca Check Point. Vale recalcar que en esta sección se ha hecho una división de la red en dos partes ya que uno de los equipos de seguridad perimetral protege la conectividad hacia las instalaciones de facultades, mientras que, el otro dispositivo se encarga de la protección del segmento de servidores.

Figura 1

Topología de red - Universidad Central del Ecuador



Fuente: Dirección de Tecnologías de la Información y Comunicaciones (DTIC)

Los mencionados equipos de seguridad perimetral son UTM[2] que ejecutan funcionalidades como: firewall, IDS/IPS, filtrado de contenido, NAT[3], VPN[4], entre otros, los cuales son un punto de enlace entre la capa de core y la salida a internet institucional. A estos equipos de seguridad perimetral se conecta toda la infraestructura LAN[5] redundante de esta red universitaria, la misma que tiene publicada aproximadamente treinta servicios en Internet destinados a sus usuarios.

Análisis de Tráfico y Conexiones Concurrentes en la Institución

En la actualidad, la UCE tiene un contrato de servicio de internet con una capacidad de 3 Gbps de Ancho de Banda, el cual es distribuido entre los segmentos de red de facultades, servidores y sedes a nivel nacional. Se debe destacar que la mayoría de consumo de tráfico y conexiones concurrentes se presentan en el campus principal de la Ciudadela Universitaria; sin embargo, por cuestiones relacionadas con la pandemia de la Covid-19 -la cual llevó a un confinamiento a nivel mundial- en aquellos momentos gran parte de esa carga fue trasladada al segmento de servidores por lo que los usuarios accedían a los servicios en línea de la institución.

A continuación, se listan los datos obtenidos de la infraestructura universitaria desde el 19 de enero hasta el 19 de junio de 2020:

- Pico más alto de tráfico: 2,2 Gbps el 24 de enero de 2020.
- Aproximadamente 11300 conexiones concurrentes diarias.
- Pico más alto de tráfico en la red (Periodo confinamiento): 868 Mbps el 20 de mayo de 2020.
- En todo el periodo del análisis realizado se obtuvieron 27200 logs del módulo de IPS.
- Aproximadamente un 80% de las conexiones a esta red se las hace desde dispositivos móviles personales conectados de manera inalámbrica.

Análisis Estadístico de los Eventos de Seguridad Logs Proporcionados por los Equipos de Seguridad Perimetral

En esta sección se realiza un análisis del comportamiento de las amenazas de seguridad en entornos universitarios mediante el estudio de los logs tomados del módulo de IDS/IPS de los equipos de seguridad perimetral pertenecientes a la UCE, la DTIC[6] ha permitido la toma de muestras de enero a junio de 2020, las mismas que fueron clasificadas por su severidad, su recurrencia y los servicios que pueden afectar.

Figura 2

Referencia CPAI de la Vulnerabilidad Inyección de Comandos sobre Payload HTTP

Fuente: Check Point Software Technologies Ltd

[2] Unified Threat Management o Gestión Unificada de Amenazas
[3] Network Address Translation
[4] Virtual Private Network
[5] LAN (Local Area Network): Red de Área Local
[6] DTIC: Dirección de Tecnologías de la Información y Comunicaciones

En la fase previa del análisis se logró observar que el equipo de la marca Check Point de la institución clasifica los eventos detectados por el IDS/IPS mediante unas referencias llamadas CPAI (Check Point Advisories), las cuales manejan la codificación mostrada en la Figura 2.

Las vulnerabilidades descritas en este trabajo hacen referencia al catálogo administrado por el MITRE[7] llamado CVE[8] lo cual facilita la clasificación de vulnerabilidades, ya que son de dominio público. De manera adicional estas referencias permiten también buscar estos ataques en otras plataformas como por ejemplo la de OWASP[9].

En el siguiente apartado se muestra uno de los análisis realizados mediante los gráficos estadísticos obtenidos en febrero de 2020: En los cuales se establece los tipos de ataques más frecuentes de forma mensual, teniendo como criterio que los mismos superen un 10% del volumen de datos capturados.

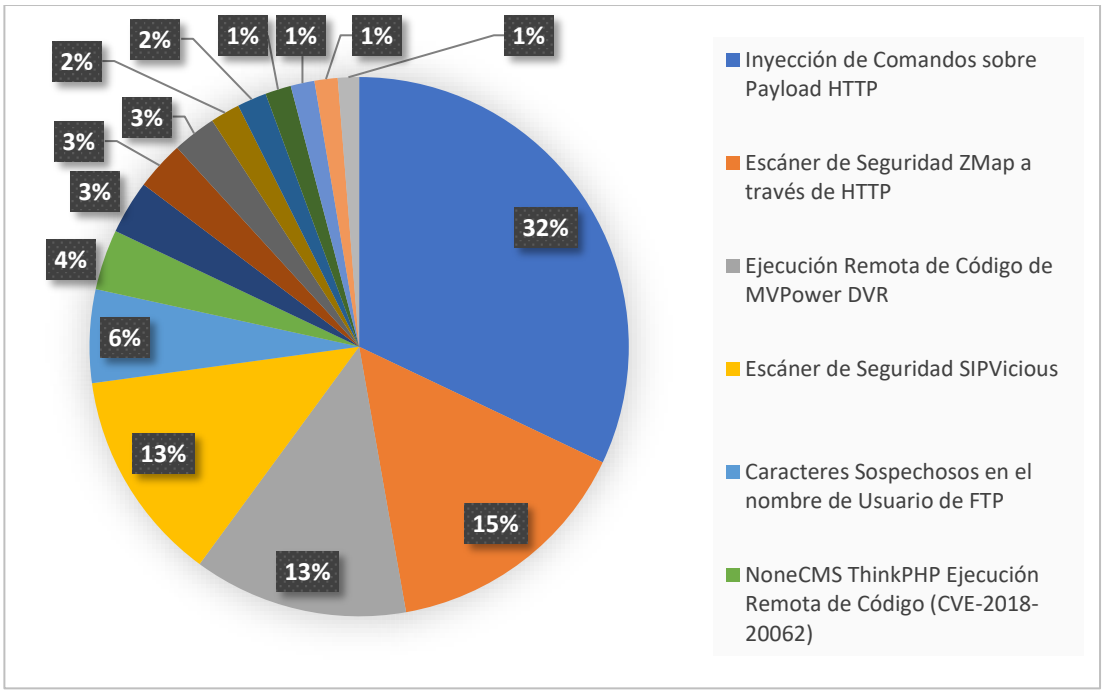
Análisis de Logs Correspondientes al Mes de febrero de 2020

Se recolectaron un total de 16214 eventos, siendo el mes en el que se han registrado la mayor cantidad de ellos.

En la Figura 3, constan los ataques más frecuentes detectados por el módulo de IDS del equipo de seguridad perimetral. Se reportan los siguientes resultados: en primer lugar, se ubica la Inyección de Comandos sobre Payload HTTP con el 32%, en segundo lugar, Escáner de Seguridad ZMap a través de HTTP con el 15%, seguido de Ejecución Remota de Código de MVPower DVR y Escáner de Seguridad SIPVicious con el 13%.

Figura 3

Ataques detectados en el mes de febrero de 2020



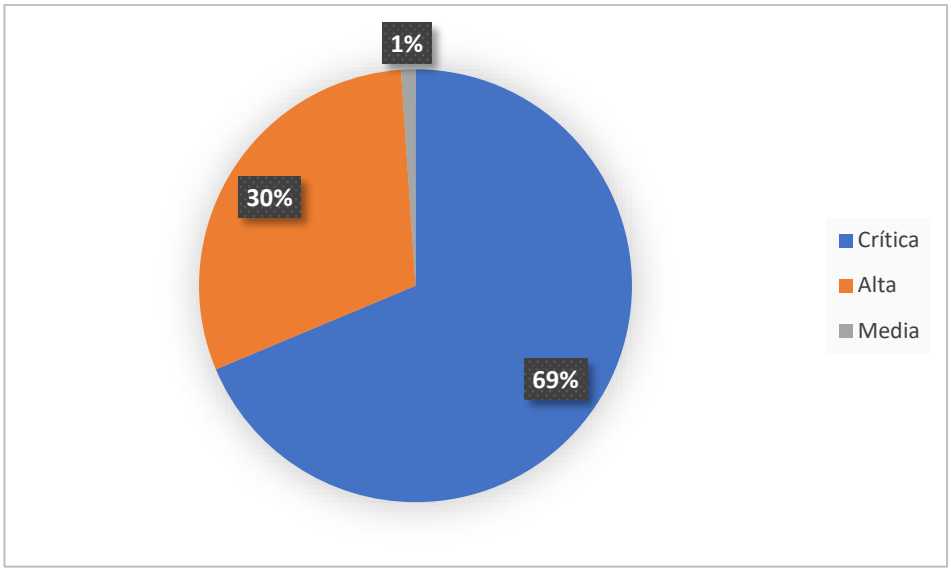
Fuente: DTIC – UCE

[7] MITRE: Organización sin ánimo de lucro perteneciente a los Estados Unidos de América.
[8] CVE: Common Vulnerabilities and Exposures, es un catálogo de vulnerabilidades.
[9] OWASP: Open Web Application Security Project que es un proyecto enfocado a la seguridad de aplicaciones Web.

En la Figura 4, se evidencia un análisis estadístico tomando en cuenta la severidad de los ataques detectados por el dispositivo, de los cuales se ha obtenido que el 69% de los ataques son críticos, mientras que el 30% son de severidad alta y el 1% media.

Figura 4

Severidad de ataques detectados en el mes de febrero de 2020

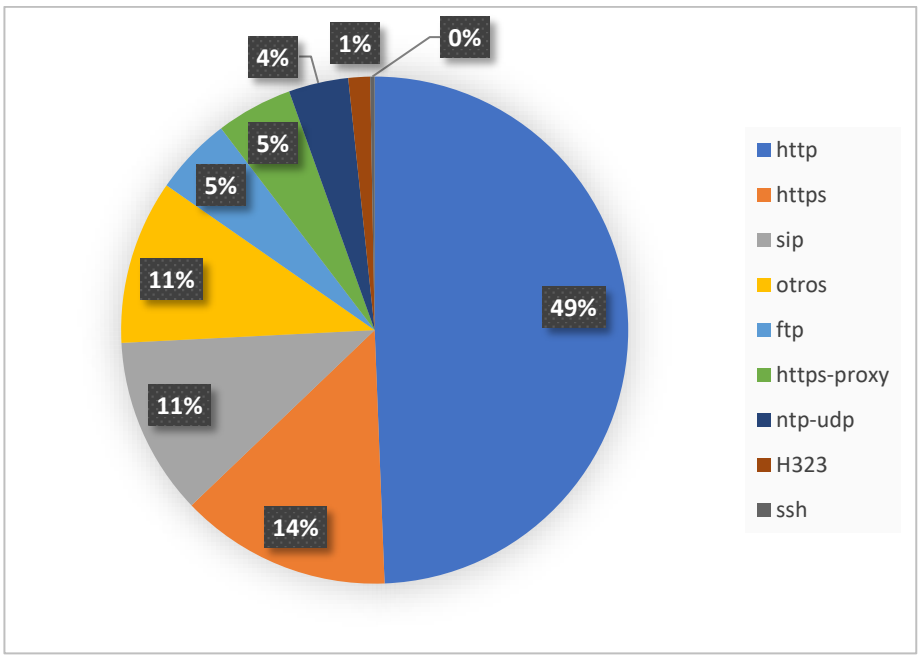


Fuente: DTIC – UCE

En la Figura 5, se verifica que los ataques detectados en febrero de 2020 están dirigidos a la afectación de los siguientes servicios: en primer lugar, se tiene http con el 49%; en segundo lugar, https con el 14%; seguido por SIP con el 11%, y para finalizar se tiene la categoría de otros con el 11%.

Figura 5

Servicios afectados por ataques en el mes de febrero de 2020



Fuente: DTIC – UCE

Interpretación de Resultados

En esta sección es importante señalar que el análisis previo a la ejecución del piloto experimental contiene en total trece gráficos estadísticos debido a que se ha realizado un promedio de tres tipos de mediciones entre los meses de enero y junio de 2020.

De los cinco gráficos que corresponden a los principales tipos de ataques captados por el IDS del equipo de seguridad perimetral se evidencia que los eventos más frecuentes son:

- En primer lugar, se encuentra Inyección de Comandos sobre Payload HTTP (Vulnerabilidad 1) el cual posee un número de referencia de Check Point CPAI-2018-0567, categorizado como "Violación de Servidor Aplicaciones Web", considerado como una vulnerabilidad con severidad crítica. La misma que afecta a la carga útil de http, en el cual el atacante envía solicitudes diseñadas para las víctimas. Si dicha explotación es exitosa el atacante puede ejecutar código arbitrario en el ordenador de la víctima (Check Point Software Technologies LTD, 2020).
- En segundo lugar (Vulnerabilidad 2), se tiene Escáner de Seguridad ZMap a través de HTTP, catalogado por Check Point como un ataque de tipo "Violación Mediante Escáner de Aplicaciones", código CPAI-2016-0215 de severidad alta. Éste es considerado un potencial atacante que puede utilizar herramientas de escaneo de vulnerabilidades como ZMap[10] para detectar brechas de seguridad en servidores de la o las víctimas (Check Point Software Technologies LTD, 2020).
- El Escáner de Seguridad SIPVicious (Vulnerabilidad 3), considerado de severidad alta, clasificado por Check Point como "Violación Mediante Escáner de Aplicaciones", código CPAI-2016-0255 (Check Point Software Technologies LTD, 2020). Por lo que respecta a SIPVicious es un conjunto de herramientas pertenecientes a la distribución de Kali Linux, que permite auditar sistemas de VoIP con protocolo SIP (KALI ORG, 2020).
- Ejecución Remota de Código de MVPower DVR (Vulnerabilidad 4) se encuentra en cuarto lugar, como una severidad crítica, identificada por Check Point mediante el código CPAI-2017-0863 y clasificado como "Violación de Protección de Servidores de Aplicaciones" (Check Point Software Technologies LTD, 2020). MVPower es un grabador de video digital que se ve afectado por una vulnerabilidad de ejecución remota de comandos, la cual permite al atacante escalar privilegios, la misma ha sido utilizada por la botnet de nombre IoT Reaper (VULNERS.COM, 2020).
- En quinto lugar, con el código CPAI-2019-0083 de Check Point se encuentra NoneCMS ThinkPHP Ejecución Remota de Código (Vulnerabilidad 5), de severidad crítica (Check Point Software Technologies LTD, 2020). Ésta es una brecha que permite ejecución remota de código, la misma también se encuentra referenciada en el CVE-2018-20062 se la considera como crítica con una calificación de 9.8/10 (NIST, 2020).
- En sexto lugar, se tiene a Inyección de Comandos a Draytek Vigor (Vulnerabilidad 6), identificado por Check Point como "Violación de Protección de Servidores de Aplicaciones", código CPAI-2020-0320 de severidad crítica, está enfocada en equipos de la marca Draytek Vigor y consiste en la ejecución de código malicioso de manera remota por parte de un atacante (Check Point Software Technologies LTD, 2020), se encuentra referenciado en el CVE-2020-8515 (NIST, 2020).
- Por último, en este análisis se tiene a Escáner de Puertos Masscan (Vulnerabilidad 7), código CPAI-2015-0259, de severidad alta, catalogado como "Violación

[10] ZMap: Proyecto de código abierto que permite realizar estudios de host y servicios publicados en Internet.

Mediante Escáner de Aplicaciones" (Check Point Software Technologies LTD, 2020). Masscan es un escáner de puertos de red semejante a NMap que permite ejecutar escaneo de puertos de una manera rápida en grandes extensiones de Internet.

Los servicios que reportaron mayor afectación son http, https y SIP, los mismos que están directamente relacionados con las vulnerabilidades descritas con anterioridad y que se encuentran debidamente referenciados.

En abril y mayo se evidencia la categoría de "otros" en la cual están incluidos varios puertos y servicios, tanto TCP como UDP, la cual ocupa el primer lugar en las estadísticas y de los cuales se ha identificado las siguientes vulnerabilidades:

- En primer lugar, se encuentra Ejecución Remota de Comandos e Inyección de Objetos en Joomla (Vulnerabilidad 8), clasificada como "Violación de Servidor Aplicaciones Web", código CPAI-2015-1401 (Check Point Software Technologies LTD, 2020), de severidad crítica y catalogada también mediante el CVE-2015-8562. Esta vulnerabilidad afecta a varias versiones de Joomla y permite a los piratas informáticos ejecutar ataques de inyección de objetos PHP (NIST, 2020).
- En segundo lugar, Técnicas de Evasión en Inyección SQL (Vulnerabilidad 9), identificada mediante código CPAI-2014-1565, de severidad crítica y clasificada como "Inteligencia de Aplicaciones", permite al atacante ejecutar comandos SQL en servidores remotos obteniendo información confidencial, ejecutar código malicioso, modificar bases de datos, entre otros (Check Point Software Technologies LTD, 2020).
- DoS[11] por Inundación de Red en Servidores Web Memcached (Vulnerabilidad 10), se encuentra en tercer lugar, identificada con el código CPAI-2018-0154, el cual también hace referencia al CVE-2018-1000115, de severidad alta y clasificada como "Violación de Servidor Aplicaciones Web" (Check Point Software Technologies LTD, 2020). Principalmente afecta al puerto UDP 11211 de Memcached, el cual es utilizado por varios sitios web para el almacenamiento de datos en caché (NIST, 2020).
- En cuarto lugar, Ejecución Remota de Comandos en Apache Struts (Vulnerabilidad 11), identificada como "Violación de Protección del Servidor Apache", código CPAI-2015-0737 de severidad alta, hace referencia al CVE-2013-2251 (Check Point Software Technologies LTD, 2020). Permite la ejecución de código malicioso en el lenguaje OGNL[12] utilizado por Apache Struts (NIST, 2020).
- El siguiente en la lista es DoS para Equipos Cisco (Vulnerabilidad 12), el cual tiene una severidad alta, con código CPAI-2018-1038, catalogado como "Violación de Protección de Servidores de Aplicaciones" (Check Point Software Technologies LTD, 2020). Tiene relación con el CVE-2018-15454 y afecta a dispositivos de la marca Cisco, en este caso el atacante puede efectuar una denegación de servicio haciendo que el CPU de los dispositivos eleve su procesamiento (NIST, 2020).
- En sexto lugar, Divulgación de Información de OpenSSL/TLS/DTLS Heartbeat (Vulnerabilidad 13), con código CPAI-2014-1336, clasificado como "Violación de la Aplicación SSL", de severidad crítica (Check Point Software Technologies LTD, 2020). El mismo tiene relación con dos vulnerabilidades reconocidas como CVE-2014-0160 y CVE-2014-0346, conocido como el error Heartbleed. El cual permite

[11] DoS: Denial-of-service attack, ataque de Denegación de Servicios.
[12] OGNL: Object-Graph Navigation Language, Lenguaje de Expresiones de código abierto utilizados en Java.

a un atacante obtener información confidencial aprovechándose de brechas en el manejo de paquetes de tipo Heartbeat Extension de TLS y DTLS en OpenSSL(NIST, 2020).

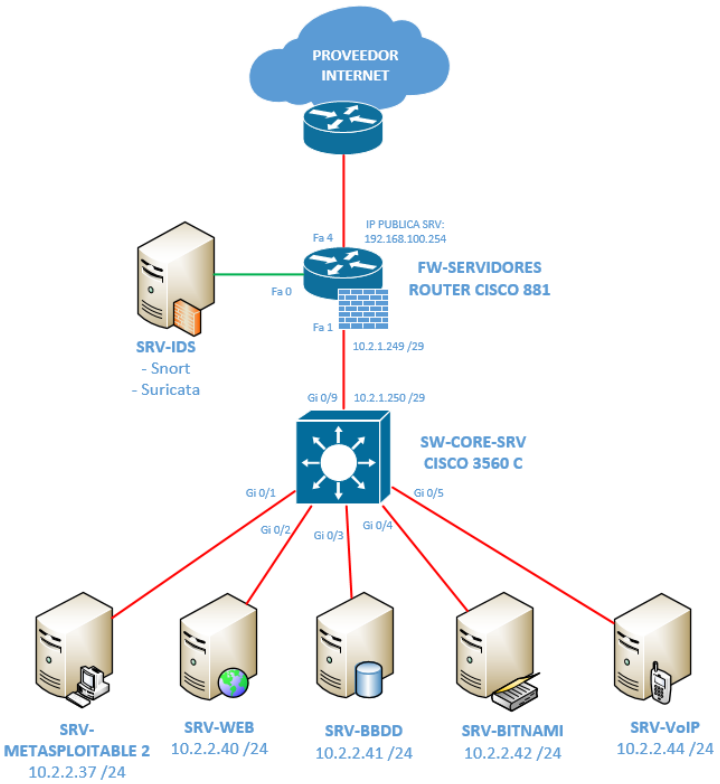
Pruebas Comparativas de Sistemas de Detección de Intrusiones de Red (NIDS) utilizando Snort y Suricata en un Entorno Emulado

En este apartado se describe la topología de red y las pruebas experimentales ejecutadas con los NIDS de código abierto Snort y Suricata. Basándose en el análisis realizado en las secciones anteriores se ha considerado importante emular el segmento red correspondiente a los servidores de la universidad, ya que al momento de realizar la investigación dicho segmento soportaba todas las conexiones que los usuarios establecían para consumir los servicios proporcionados por la institución.

En la topología mostrada en la Figura 6, se pueden diferenciar la capa de borde representada por un router Cisco 881, el cual cumple con las funciones de firewall, como capa de core se ha utilizado un switch Cisco 3560. Al mismo se conectan todos los servidores de pruebas que tienen publicados diferentes servicios. De manera adicional se ha ubicado un ordenador que hace las funciones de NIDS y que tiene virtualizados tanto Snort como Suricata, los mismos que están destinados a obtener los datos producidos por los ataques que van a ser ejecutados desde el exterior de la red.

Figura 6

Diagrama Topológico del Prototipo Experimental Emulado



Implementación de Snort como NIDS

Snort es una de las aplicaciones open source más conocidas por su efectividad en el análisis de tráfico, debido a la gran cantidad de firmas que posee. Además, en la actualidad, está avalada por la empresa Cisco destacada en el ámbito de las redes a nivel mundial.

Para la implementación de Snort como NIDS se ha tomado como base la distribución CentOs 7 Minimal, en la cual se han instalado todos los paquetes requeridos y se han seguido los procedimientos de configuración proporcionados en la sección "Documents" de la página web de Snort www.snort.org/documents.

Implementación de Suricata como NIDS

Suricata es otra aplicación open source la cual puede ser utilizada como IDS, la misma que ha sido escogida para este piloto experimental debido a sus prestaciones y el óptimo manejo de reglas el cual es parecido a las utilizadas por Snort. Sin embargo, y como se pudo determinar en el estado del arte realizado al inicio de este proyecto Suricata tiene como ventaja la utilización de varios procesadores del servidor, en el cual se encuentre instalado lo que lo hace más efectivo al momento de realizar el análisis de tráfico, en especial cuando se habla de grandes infraestructuras de red.

De manera análoga a lo explicado en la sección anterior, para la instalación y puesta en marcha de Suricata como NIDS se han seguido las guías de configuración presentes en la página web de este aplicativo https://suricata.io/documentation/.

Análisis Comparativo Basado en los NIDS Snort y Suricata

A continuación, como muestra se presenta una de las PoC[13] realizadas a base de la topología presentada al inicio de esta sección. Cada uno de estos escenarios de ataque emulados se enfocan en las trece vulnerabilidades previamente analizadas en la infraestructura de red perteneciente a la institución universitaria y listadas previamente en este trabajo de investigación. Para cada vulnerabilidad se efectúa un procedimiento experimental con el objeto de evaluar si la misma es detectada tanto por Snort como por Suricata que están funcionando como NIDS. Con los resultados obtenidos se realizó el estudio comparativo acerca de la efectividad de los mismos al ser enfrentados a diferentes tipos de amenazas que existen en este tipo de entornos. El modelo de PoC ejecutado se presenta en el siguiente apartado.

Vulnerabilidad 11: Ejecución Remota de Comandos en Apache Struts

Como se explica en el análisis previo a esta prueba, la vulnerabilidad permite inyección de código en el lenguaje OGNL que utiliza Apache. Para comprobar si Snort detecta este tipo de ataques se plantea un escenario donde una máquina virtual con Metasploitable 2 es la víctima, la cual será el objetivo de un atacante que utiliza Metasploit framework de Kali Linux para explotar esta vulnerabilidad.

En la Figura 7 se indica que dentro de la herramienta ya mencionada se encuentra el exploit "Apache Struts 2 Namespace Redirect OGNL Injection" con el cual se procede a ejecutar el ataque hacia el ordenador vulnerable.

Figura 7

Ataque Apache Struts 2 Namespace Redirect OGNL Injection

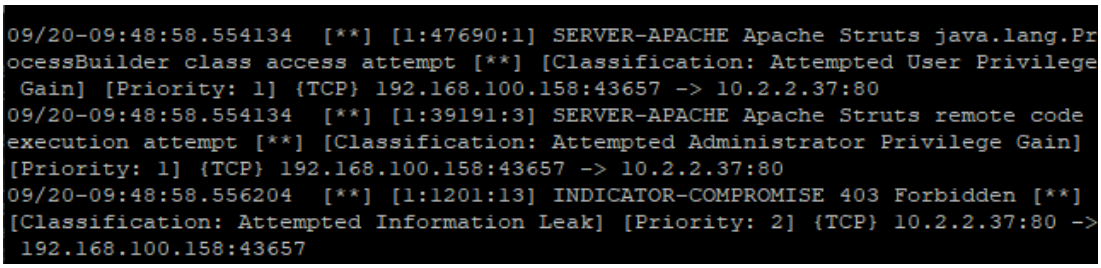
```
msf5 exploit(multi/http/struts2_namespace_ognl) > set RHOSTS 192.168.100.24
8
RHOSTS ⇒ 192.168.100.248
msf5 exploit(multi/http/struts2_namespace_ognl) > set RPORT 80
RPORT ⇒ 80
msf5 exploit(multi/http/struts2_namespace_ognl) > exploit

[*] Started reverse TCP handler on 192.168.100.158:4444
[*] Generated 207 byte binary payload
```

[13] PoC (Proof of Concept): Prueba de Concepto

La Figura 8 muestra que al momento de realizar la comprobación mediante la consola del NIDS Snort se puede verificar que se han generado tres tipos de logs relacionados con esta vulnerabilidad.

Figura 8

Alertas en la consola de Snort para el ataque Apache Struts 2



De las alertas obtenidas se pueden diferenciar dos que pertenecen a la categoría "SERVER-APACHE RULES" la primera con Sid[14] 1-47690 mediante la cual se notifica al momento que un atacante intenta acceder a la clase java.lang.ProcessBuilder con el objeto de obtener privilegios de administrador. La segunda tiene el identificador Sid 1-39191, el cual notifica intentos de ejecución de código remoto enfocado en versiones de Apache Struts 2.3.20.X. El tercer log mostrado con Sid 1-1201, se genera al momento que un servidor responde con un estado "403 Forbidden" y pertenece a la categoría "INDICATOR-COMPROMISE RULES" (Snort, 2020).

De manera adicional se ha verificado en la página web de Snort que existen otras reglas relacionadas a este tipo de vulnerabilidad la mismas que pertenecen a la categoría "SERVER-APACHE RULES" (Sid 1-27243, Sid 1-27244, Sid 1-27245, Sid 1-29747, Sid 1-29748 y Sid 1-49885) (Snort, 2020). Con lo que se puede determinar que este tipo de vulnerabilidad está ampliamente soportada por la plataforma.

Por otro lado, se procedió a revisar los eventos registrados por el IDS Suricata. Se comprobó de la existencia de cuatro reglas relacionadas con esta amenaza. La primera es "ET EXPLOIT Apache Struts Posible OGNL Java ProcessBuilder URI" la misma que se describe como un intento de escalada de privilegios el identificador de la misma es el 2017172. La segunda alerta detectada es "ET WEB_SERVER Posible Apache Struts OGNL in Dynamic Action", el identificador 2017277 de la misma manera que la anterior detecta esta actividad como un intento de escalada de privilegios. El siguiente evento registrado es "ET WEB_SPECIFIC_APPS Apache Struts java.lang inbound OGNL injection remote code execution attempt" que también alerta lo descrito en los otros dos logs, su identificador es el 2026033. Por último, se registró "GLP WEB_SERVER 403 Forbidden" el mismo que alerta sobre un intento de fuga de información, con esta prueba se determina que este IDS detecta de manera efectiva este ataque.

Las Figuras Figura 9 y Figura 10 muestran el contenido de los archivos fast.log y eve.json respectivamente.

[14] Sid: Snort identifier (Identificador de la categoría de una regla).

Figura 9

Contenido del archivo fast.log para la vulnerabilidad de Apache Struts 2

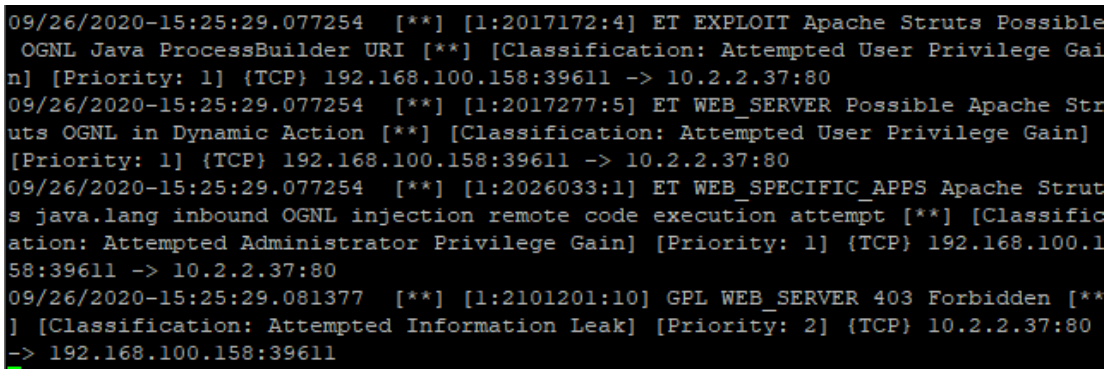
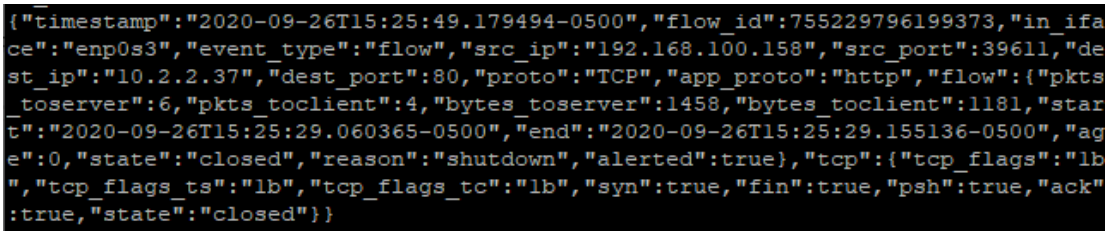


Figura 10

Tráfico almacenado en el archivo eve.json de Suricata para la vulnerabilidad de Apache Struts 2



Análisis Comparativo de las Soluciones de IDS Check Point, Snort y Suricata

En el Tabla 1 se evidencia el análisis comparativo de las plataformas en software libre versus la de software licenciado evaluadas en este trabajo de investigación. Esta tabla cuanticualitativa está compuesta por el conjunto de vulnerabilidades previamente analizados, una sección de evaluación de cumplimiento de la prueba experimental, una explicación técnica basada en las pruebas experimentales y al final el porcentaje de vulnerabilidades detectas por cada una de las soluciones.

Tabla 1

Análisis Comparativo de IDS Evaluados

VULNERABILIDAD	IDS CHECKPOINT (UCE)		IDS SNORT		IDS SURICATA	
	DETECCIÓN	ANÁLISIS	DETECCIÓN	ANÁLISIS	DETECCIÓN	ANÁLISIS
1	X	Identificado como una vulnerabilidad que afecta a aplicaciones en entornos Web.		No detecta el agente de ataque, solo detectar intentos de emitir solicitudes HTTP utilizando una dirección IP en particular.	X	Identifica que es un ataque de inyección de comandos sobre una aplicación web, de manera adicional identifica el agente de ataque.

VULNERABILIDAD	IDS CHECKPOINT (UCE)		IDS SNORT		IDS SURICATA	
	DETECCIÓN	ANÁLISIS	DETECCIÓN	ANÁLISIS	DETECCIÓN	ANÁLISIS
2	X	Detecta al software Zmap como el origen del ataque de escaneo de puertos en la infraestructura de red.	X	A pesar de no detectar el agente malicioso de manera específica si detecta los intentos de escaneo en los puertos donde se efectuó los ataques.		No captó los ataques efectuados por este tipo de software.
3	X	Señala al software Sipvicius como el origen del ataque.	X	Detecta de manera efectiva al software Sipvicius como origen del ataque.	X	Detecta de manera efectiva al software Sipvicius como origen del ataque .
4	X	Señala la afectación de equipos de la marca MVPower debido a ejecución remota de comandos.	X	Detecta efectivamente la vulnerabilidad en dispositivos MVPower.	X	Detecta efectivamente la vulnerabilidad en dispositivos MVPower.
5	X	Reconocido por una vulnerabilidad que permite ejecución remota de código.	X	Se ha detectado de manera efectiva esta vulnerabilidad.		No soporta este tipo de vulnerabilidad, no se encontró referencias en el conjunto de reglas.
6	X	Señala la afectación de equipos de la marca Draytek debido a ejecución remota de comandos.	X	Existen varias reglas que soportan este tipo de vulnerabilidad.		No soporta este tipo de vulnerabilidad, no se encontró referencias en el conjunto de reglas instaladas.
7	X	Identifica a MASSCAN como agente de ataque mediante la ejecución de escaneo de puertos.		No reconoce al software MASSCAN como origen del ataque, solo muestra logs genéricos.		No reconoce al software MASSCAN como origen del ataque, solo muestra logs genéricos.

VULNERABILIDAD	IDS CHECKPOINT (UCE)		IDS SNORT		IDS SURICATA	
	DETECCIÓN	ANÁLISIS	DETECCIÓN	ANÁLISIS	DETECCIÓN	ANÁLISIS
8	X	Ataque de inyección de objetos PHP a varias versiones de Joomla.	X	Si reconoce el tipo de vulnerabilidad referido a Joomla.		No soporta este tipo de vulnerabilidad, no se encontró referencias en el conjunto de reglas instaladas.
9	X	Detecta técnicas relacionadas con la inyección de comandos en SQL Server.	X	Existe una amplia variedad de reglas que detectan este tipo de ataques.	X	Existe una amplia variedad de reglas que detectan este tipo de ataques.
10	X	Detecta ataques de DDoS a Memcached en el puerto UDP 11211.	X	Existe una amplia variedad de reglas que detectan este tipo de ataques.	X	Existe una sola regla enfocada en este tipo de ataque.
11	X	Detecta ejecución de código malicioso en el lenguaje ONGL que utiliza Apache Struts.	X	Existe una amplia variedad de reglas que detectan este tipo de ataque.	X	Existe una amplia variedad de reglas que detectan este tipo de ataque.
12	X	Detecta DoS a equipos Cisco.	X	Existe un conjunto de reglas genéricas que detectan este tipo de ataque.		Detecta de manera parcial este tipo de ataque.
13	X	Detecta brechas de seguridad en Open SSL TLS DTLS con paquetes de tipo Heartbeat.	X	Existe una amplia variedad de reglas que detectan este tipo de ataque.	X	Existe una amplia variedad de reglas que detectan este tipo de ataque.
Total de Vulnerabilidades detectadas	13		10		7	
Porcentaje de Vulnerabilidades detectadas	100%		77%		54%	

Mientras que la Tabla 2 resume un conjunto de ventajas y desventajas proporcionadas por estas plataformas que fueron evaluadas en distintos ámbitos de acorde a los resultados obtenidos en las pruebas experimentales.

Tabla 2

Ventajas y Desventajas de las Plataformas Evaluadas

COMPARATIVO DE VENTAJAS Y DESVENTAJAS			
CRITERIOS	**CHECKPOINT (UCE)**	**SNORT**	**SURICATA**
Actualización de reglas	Las actualizaciones se pueden configurar para que se ejecuten de manera automática. La marca posee una nube propia donde se encuentran todo tipo de actualizaciones y bases de datos de ataques de día cero; sin embargo, esto depende si tiene el licenciamiento activado. Como desventaja dicho licenciamiento tiende a ser costoso. La marca ha creado una comunidad de usuarios para reportar amenazas o falsos positivos llamada CheckMates. No permite la creación de reglas propias.	Posee una gran base de datos de firmas para actualizar el IDS. Se puede programar actualizaciones manuales o automáticas con diversas herramientas. Por ejemplo, oinkmaster, que posee soporte de Talos que es una gran nube con bases de datos de reglas, se puede tener nivel de suscripción, por ejemplo, una cuenta gratuita actualiza sus bases de datos cada 30 días, mientras que una suscripción pagada puede recibirlas diariamente. Los costos con accesibles en comparación con Checkpoint. De manera adicional puede soportar las community rules proporcionadas por EmergingThreasts. Se pueden configurar reglas de manera manual.	La actualización de las firmas del IDS se las ejecuta desde el sitio Web de EmergingThreasts, lo cual potencializa a esta plataforma ya que las reglas nativas de la misma son limitadas, se puede programar actualizaciones automáticas mediante herramientas como oinkmaster. Una desventaja es que no se pueden cargar otro tipo de firmas por lo que lo hace limitado. Se pueden configurar reglas de manera manual. Ampliamente apoyado por comunidades de software libre.
Referencias CVE	Cada una de las reglas de la marca poseen su respectiva referencia con la lista de vulnerabilidades de CVE lo que hace más fácil la identificación de una amenaza.	La mayoría de las reglas buscadas en la página Web de Snort se referencia a la lista de vulnerabilidades de CVE.	Se debe buscar en la página de EmergingThreasts, el tipo de vulnerabilidad para verificar la existencia de una regla, no siempre se puede encontrar la referencia CVE.
Información proporcionada por los logs	Las reglas presentan alertas en la interfaz gráfica de la plataforma las cuales tienen información completa de la vulnerabilidad detectada.	Las alertas visualizadas en la consola del equipo entregan información básica de la vulnerabilidad encontrada, para complementar esa información se debe buscar en la página Web de Snort.	Suricata registra las alertas de vulnerabilidad en dos archivos, eve.json que entrega información bastante detallada sobre el flujo de tráfico detectado, protocolo, clasificación, entre otros, mientras que en el archivo fast.log se registra el evento con un formato similar al presentado por Snort.

COMPARATIVO DE VENTAJAS Y DESVENTAJAS			
CRITERIOS	**CHECKPOINT (UCE)**	**SNORT**	**SURICATA**
Capacidad de detección	Alta capacidad de detección, gracias a la actualización de vulnerabilidades de día cero que maneja la marca.	En el estudio realizado se determinó que tiene una buena capacidad de detección de amenazas, tomando en cuenta que las pruebas se las ejecutó con una cuenta de suscripción gratuita.	La capacidad de detección es buena, pero depende de la cantidad de firmas que tenga el IDS y de la actualización de estas, resulta a veces laborioso la búsqueda de este tipo de reglas.
Facilidad de uso	Posee interfaz gráfica lo que proporciona un manejo más intuito del módulo de IDS. Sin embargo, si se exige de un manejo de datos avanzado se requiere un conocimiento a nivel de experto en la marca ya que la consola de comandos maneja comandos propietarios.	Los comandos de ejecución no son complejos, por lo que resulta fácil el acceso a la herramienta y la lectura de logs, los mismos que se presentan en tiempo real en la consola del equipo. Se puede integrar una interfaz gráfica como complemento de Snort.	Los comandos de ejecución no son complejos, por lo que resulta fácil el acceso a la herramienta, las alertas se almacenan en tiempo real en dos archivos sin necesidad de configuración previa lo que resulta una ventaja al momento de extraer esos datos para un posterior estudio. Se puede integrar una interfaz gráfica como complemento de Suricata.
Integración con otras funcionalidades	Al ser un equipo propietario la mayor parte de funcionalidades deben ser de la marca y que contengan el licenciamiento respectivo, lo que incrementa el costo de la solución.	Fácil integración con otras funcionalidades, como manejo de entorno gráfico, bases de datos, geolocalización, entre otras, lo que no conlleva costos extras en la implementación.	Fácil integración con otras funcionalidades, como manejo de entorno gráfico, bases de datos, geolocalización, entre otras, lo que no conlleva costos extras en la implementación.

Mejores Prácticas de Configuración de IDS

A base del análisis comparativo realizado en la sección anterior se ha podido establecer un conjunto de buenas prácticas de configuración de reglas de IDS las cuales están dirigidas a la optimización del funcionamiento de estas plataformas en entornos universitarios:

- Al momento de la implementación de un sistema de detección de intrusiones es fundamental establecer una fase inicial destinada a la recopilación y análisis de vulnerabilidades. En esta fase en la que el dispositivo se encuentra en modo de escucha se pueden detectar los tipos de ataques más recurrentes, clasificarlos en una escala dependiendo de su riesgo y ejecutar un plan de acción para la configuración de las reglas de IDS basándose en los patrones previamente captados. Con eso se logra una implementación acorde a las necesidades de la institución.
- Es importante que dentro de los análisis realizados se puedan buscar patrones de comportamiento de tráfico legítimo como, por ejemplo, servicios publicados por la institución que no requieran estar siendo monitorizados y se puedan crear excepciones a las reglas sobre dicho tráfico. Con esto se logra ahorrar en recursos de procesamiento del servidor donde se tenga instalada la solución.
- Es importante tener conocimiento de cómo realizar un contacto con la empresa u organismo propietario de la solución con el objeto de poder reportar tráfico anómalo, falsos positivos, brechas de seguridad en las reglas, entre otros. Como,

por ejemplo, en el caso de Check Point se debe notificar al centro de soporte de usuarios, Snort puede reportarlo en su página web y Suricata puede remitir esta información en la página web de EmergingThreasts.

- Nunca se debe dejar de lado el procedimiento de actualización no solo de las bases de datos de firmas que tienen este tipo de plataformas, sino también se debe actualizar y parchar de manera adecuada el sistema operativo donde se encuentre funcionando el IDS, como un ejemplo de esto. Snort y Suricata se actualizan por medio de la herramienta oinkmaster, así también, se debe actualizar debidamente el sistema operativo CentOS 7 que es la base de este prototipo experimental.
- A pesar de tener un conjunto de configuraciones funcional y firmas debidamente actualizadas es indispensable que en cualquier infraestructura de red se realicen análisis de tráfico de manera periódica, con el objeto de revisar los eventos generados y con esto poder prevenir o descartar potenciales nuevas amenazas.
- A base de los trabajos ejecutados en este piloto experimental se recomienda como una buena práctica en la implementación de este tipo de entornos realizar copias de seguridad del IDS que tenga la configuración más estable, antes de ejecutar cualquier cambio en el sistema operativo, su aplicativo o sus reglas, este contingente se vuelve fundamental en el caso de recuperación de fallos.
- Es indispensable en especial en un entorno de tipo universitario no solo monitorear el tráfico externo a la red, sino también el tráfico de la LAN ya que en este tipo de ambientes existe una gran cantidad de equipos que no pueden ser controlados ya que los mismos son de uso personal como es el caso de Smartphone, laptops, entre otros.
- Es importante realizar una revisión y clasificar las reglas que se encuentren descargadas en el dispositivo, con el objeto de deshabilitar cualquiera de ellas que no tenga nada que ver con el tráfico monitorizado. Por ejemplo, en un entorno universitario no se tiene tráfico de tipo industrial, por lo que, se puede deshabilitar reglas referidas a Scada, Modbus, etc., este procedimiento permite optimizar los recursos de procesamiento del servidor.
- El rendimiento del equipo donde se encuentre instalada una de estas plataformas depende de la complejidad de las reglas configuradas y la cantidad de tráfico que atraviese la misma, ya que en muchos casos cuando el dispositivo inspecciona una gran cantidad de paquetes dicho rendimiento baja o empieza a descartar algunos de ellos.
- Es importante verificar si cada una de estas plataformas tienen la funcionalidad de IPS, ya que luego de verificar la red y registrar todo tipo de comportamiento sea este anómalo o no, se deben configurar las reglas necesarias para poner a la plataforma de seguridad en modo de Prevención.

Tomando como referencia las buenas prácticas aquí descritas se ha procedido a la aplicación de éstas en el módulo de IPS del equipo Check Point perteneciente a la UCE. Los procedimientos realizados se los ejecutaron en conjunto con el personal de Infraestructura que es parte de la DTIC.

Conclusiones

En este trabajo se concluyó que mediante un análisis de los logs generados por los IDS propuestos en el piloto experimental se puede obtener información relevante acerca del comportamiento de determinados tipos de ataques a los que se enfrentan los entornos universitarios. En particular a identificar diferentes tipos de tráfico, sus orígenes, el tipo de contenido de los paquetes, y como estos son evidenciados por plataformas basadas en software

libre. De esta manera determinar el grado de efectividad de este tipo de IDS al enfrentarse a entornos reales.

Se comprobó que implementando un piloto experimental se puede establecer un entorno de pruebas de concepto completo en el cual se configuran un conjunto de dispositivos los mismos que puedan emular un entorno real. En este caso en particular, se estableció una infraestructura tecnológica que permitía tener servicios publicados al exterior de la red, atacantes externos y los NIDS los cuales permiten analizar toda esa interacción de la red experimental con el exterior. Esto proporciona un ambiente de pruebas controlado el cual no pone en riesgo información crítica de usuarios, equipos y servicios pertenecientes a los entornos universitarios.

Con el conjunto de ataques ejecutado en este trabajo de investigación se logró determinar la efectividad de los sistemas de detección de intrusiones basados en software libre, previamente implementados, al momento de identificar diferentes tipos de patrones de ataques, los cuales se obtuvieron de un análisis previo de amenazas captadas en un entorno universitario real.

Con el análisis comparativo efectuado se verificó las fortalezas y debilidades de cada uno de los IDS puestos a prueba. Los cuales fueron evaluados de acuerdo con la capacidad de detección del conjunto de vulnerabilidades emuladas en laboratorio, con esta comparativa se obtuvo el porcentaje de efectividad de los mismos. Por lo que, a base de estos parámetros valorados también se logró determinar un conjunto de ventajas y desventajas de cada una de estas soluciones.

Luego del estudio realizado se logró establecer un conjunto de buenas prácticas de configuración las cuales son una base para la implementación de cualquier tipo de plataformas de sistemas de detección de intrusiones en general. Estas prácticas permiten optimizar el funcionamiento de estos dispositivos, ahorrar recursos en procesamiento y clasificar de manera adecuada los eventos reportados por el IDS.

Se determinó mediante un análisis exhaustivo los tipos de patrones de ataque más recurrentes en un entorno universitario evaluados en un determinado periodo de tiempo basado en las pruebas experimentales ejecutadas y una efectiva valoración de estas al realizar el análisis comparativo.

Finalmente es importante acotar que dentro de este estudio existió una limitante circunstancial, ya que, al encontrarse en pleno periodo de pandemia se hizo complicado la implementación de los IDS experimentales directamente en la infraestructura de la institución ya que no se contaban con los permisos de acceso requeridos. Por ello, se considera una opción viable a futuro implementar estos dispositivos de manera presencial, así también, este proyecto sienta un precedente para la ejecución de trabajos futuros en los que se pueda analizar nuevos patrones de ataque especialmente utilizando técnicas de análisis de anomalías de tráfico enfocadas en este tipo de instituciones. De esta manera potenciar la detección de malware de día cero.

Reconocimientos

Este trabajo forma parte de la Tesis “Análisis y Pruebas Comparativas de Sistemas de Detección de Intrusos (IDS) en Entornos Universitarios” en la Universidad Internacional de La Rioja. El autor desea expresar su agradecimiento a la Dirección de Tecnologías de la Información y Comunicaciones (DTIC) de la Universidad Central del Ecuador y en particular

a su Director el Ing. Cesar Morales Mejía, MEd. por la colaboración en este trabajo de investigación.

Referencias

Check Point Software Technologies LTD. (2020, Julio 16). *Check Point Advisories*. https://www.checkpoint.com/advisories/

Hernández, R., Fernández, C., & Baptista, P. (2010). *Metodología de la Investigación.* Mexico: The Mc Graw-Hill.

KALI ORG. (2020, Julio 16). *KALI TOOLS*. SIPVicious Package Description: https://tools.kali.org/sniffingspoofing/sipvicious

NIST. (2020, Abril 14). *National Vulnerability Database*. https://nvd.nist.gov/vuln

OWASP. (2020, Julio 16). *OWASP COMMUNITY*. Obtenido de https://owasp.org/www-community/attacks/

Snort. (2020, Septiembre 21). *snort.org*. SNORT: https://snort.org/

VULNERS.COM. (2020, Julio 2). *VULNERS.COM*. https://vulners.com/nessus/AOST_NVR_SHELL.NASL