UNIVERSIDAD SAN FRANCISCO DE QUITO

Colegio de Ciencias e Ingeniería

Técnicas de detección de ataques en un sistema SIEM (Security Information and Event Management)

Patricio Javier Moreano Jurado

Mauricio Iturralde Ph.D., Director de Tesis

Tesis de grado presentada como requisito para la obtención del título de Ingeniero de Sistemas

Quito, 30 de julio del 2015

Universidad San Francisco de Quito
Colegio de Ciencias e Ingeniería

HOJA DE APROBACION DE TESIS

Técnicas de detección de ataques en un sistema SIEM (Security Information and Event Management)

Patricio Javier Moreano Jurado

Mauricio Iturralde, Ph.D.,
Director de Tesis _____

Fausto Pasmay, MSc.,
Director de la carrera de Ing. En Sistemas _____

Fernando Sánchez, Ph.D.,
Miembro del Comité de Tesis _____

Quito, 30 de julio del 2015

Firma:

Nombre: Patricio Javier Moreano Jurado

C. I.: 1715315204

Fecha: Quito, 30 de julio del 2015

DEDICATORIA

Este trabajo es dedicado a mis padres que durante toda mi vida estudiantil me han dado su apoyo incondicional. Además a mi hermano Juan Fernando y a Nicole.

Patricio J. Moreano J.

AGRADECIMIENTO

A la Universidad San Francisco de Quito, por permitir una educación liberal en todos los ámbitos y ayudar a formar profesionales comprometidos con la democracia y la libertad en el país.

Una mención especial a mi profesor y tutor de tesis, Mauricio Iturralde por su apoyo a lo largo de este trabajo y a mi director de carrera el Sr. Fausto Pasmay por su conocimientos en estos más de 5 años que estuve en las aulas de la Universidad.

PATRICIO J. MOREANO J.

RESUMEN

El avance de la tecnología ha logrado un mundo casi enteramente globalizado. La velocidad con la que se consigue nuevos inventos ya sean digitales o no, ha revolucionado el ritmo de vida en la mayoría de las personas. La información se ha vuelto un recurso muy utilizado y de mucho valor, por lo que proteger dicha información se ha vuelto un trabajo muy demandado. La globalización y la interconectividad de redes (el Internet) han logrado mantener en contacto a seres humanos muy alejados unos de otros.

Debido a estos avances, los ataques informáticos a las redes se han vuelto objetivos por parte de atacantes que intentan conseguir información confidencial o no permitir la disponibilidad de recursos en la red. Los sistemas de información y manejo de eventos (SIEM por sus siglas en inglés) se han vuelto la defensa a estos ataques. Como detectar ataques y preparar procedimientos y algoritmos para proteger información es el objetivo de este trabajo que desarrolla soluciones a base de entender los sistemas y la teoría detrás de cada ataque informático.

ABSTRACT

Technology advance has achieved an almost entirely globalized world. New inventions are achieved at a speed that has revolutionized people's pace of life. Information has become a very helpful and of great value resource. This has made the protection of information a demanded work. Globalization and the Internet have managed to maintain in contact to people all around the world.

Due to this progress cyber-attacks to networks have become a main objective for hackers that attempt to gain people credentials or not allowing the availability of network resources. System Information and Event Management (SIEM) have become the main defense against those attacks. How to detect attacks and prepare procedures and algorithms to protect information is the objective of this work that develops solutions when understanding theory and systems behind every cyber-attack.

TABLA DE CONTENIDOS

LISTA DE FIGURAS

CAPÍTULO 1.- INTRODUCCIÓN

Justificación e Importancia del Proyecto

La tecnología y la digitalización de la información convierten a los datos en un activo muy importante de las empresas y de los individuos en general. Es fundamental saber cómo proteger los datos para evitar ser víctima de fraudes y delitos informáticos. A pesar de que hoy en día las técnicas de seguridad hacia los datos y hacia la infraestructura de las comunicaciones están en auge, tales herramientas de seguridad como firewalls o aplicaciones que permiten prevenir ataques informáticos no son absolutamente confiables. Es importante señalar que existen algunos problemas con las maneras tradicionales de combatir problemas de seguridad en redes, algunos son: 1) Muchos registros (logs) de eventos generalmente se auto programan para ser borrados en un periodo de tiempo limitados, y muchas veces, algunos logs particulares pueden ser muy importantes para evitar ataques. 2) Los sistemas de seguridad SIEM (Security Information and Event Management) suelen ser muy rígidos y no soportan eventos que sean diferentes a los estandarizados. 3) Tener 'warehouse' de datos (para almacenamiento) suelen ser muy costosos y muchas empresas no están dispuestos a incurrir en estos costos, sin verlos como una inversión.

Por estas razones y algunas otras, es imperioso que se sigan realizando análisis de cómo mejorar el rendimiento de sistemas de seguridad SIEM para reducir la

probabilidad de ataques a redes de seguridad de cualquier tipo. El mejoramiento puede darse en distintos campos de los sistemas hechos para prevenir ataques: mejor análisis visual (componentes GUI), procesamiento más rápido de correlación de eventos en ‘logs’ o diferentes algoritmos que permitan estandarizan mayor cantidad de eventos para prevenir futuros ataques en base a comportamientos parecidos.

Antecedentes

Dentro del campo de la seguridad para redes informáticas, los sistemas de seguridad SIEM tienen varios modelos que han servido para el mejoramiento hacia sistemas muchos más seguros y eficaces en su funcionamiento. A continuación se muestra una recopilación de lo hecho en este campo para tener una mejor visión de hacia dónde apuntar con este trabajo.

Los IDS (Intrusion Detection Systems) son parecidos y trabajan bajo los mismos parámetros que los SIEM. Muchos problemas para tratar amenazas a los sistemas recaen en que no existe normalización para encapsular eventos similares de manera coherente. Diferentes sistemas utilizan diferentes formatos lo que hace difícil de correlacionar si no se utiliza el mismo sistema. “Se provee una plataforma para normalizar eventos en un súper evento unificado basado en un estándar del CEE (Common Event Expression) desarrollado por la corporación Mitre.” [4].

Otro caso de análisis es el de la técnica LSA (Latent Semantic Analysis) propuesto en el ensayo de Pavarit Dairinram y otros autores. Se analiza esta técnica debido a que los sistemas SIEM no pueden procesar tantas amenazas distintas. “LSA es propuesto para alivianar los problemas de volumen. Mejora el rendimiento al reducir el ‘ruido’ de grandes cantidades de datos generados en dispositivos. También se usa para detectar patrones en las amenazas basados en eventos y logs. Los experimentos muestran que el enfoque LSA ayuda eliminando datos insignificantes

sin eliminar a su vez datos válidos y relevantes” [10]. En el artículo se habla de 3 distintas técnicas para identificar amenazas a la red. Es otro enfoque muy válido al momento de analizar y optimizar los sistemas SIEM.

“Una nueva arquitectura AMSEC (Attack Modeling and Security Evaluation Component) es propuesta” [18]. Los componentes propuestos para sustentar un SIEM permiten: “encontrar y corregir errores en la configuración de red, revela posibles actos de amenazas a la seguridad, determina recursos críticos de la red y elige una política segura y efectiva para manejar las amenazas” [18].

Objetivos

Objetivo Principal

Proponer algoritmos de detección de ataques para cuatro ataques comunes en un sistema SIEM, mediante la revisión del funcionamiento de los ataques, así como también de revisión de trabajos empíricos. Esto se propone alcanzar con la ayuda de métodos matemáticos y estadísticos para fortalecer las redes de seguridad y evitar ataques posteriores.

Objetivos Específicos, Metas y Actividades

Objetivos Específicos	Metas	Actividades
Diseño y explicación de un sistema SIEM en base a otros ya creados, donde se explicará cada componente en particular.	Entregar al final del estudio, al menos un nuevo método de técnica para la detección de ataques a un sistema SIEM.	Crear gráficos que permitan entender el funcionamiento y composición de un sistema SIEM.
		Realizar conclusiones sustentables en el análisis que ratifiquen la viabilidad del nuevo algoritmo propuesto.
	Evaluar todas las ventajas y desventajas de algoritmos previamente propuestos para la detección de ataques a redes de seguridad.	De una manera gráfica inducir a una comprensión visual de algoritmos de detección de ataques previamente hechos y puestos a prueba.
Explorar toda la terminología de un sistema SIEM, así como también tendencias de ataques que se han registrado a dichos sistemas.	Entender holísticamente el funcionamiento de un sistema SIEM y como se producen ataques a redes de seguridad.	Realizar análisis estadístico e informativo de alguna red de seguridad en particular.
		Realizar un glosario de términos como ‘keywords’ para facilitar el entendimiento de la tesis para los lectores.
Analizar distintos ataques a distintas redes de seguridad para encontrar tendencias de cómo se producen los ataques.	Aprender a determinar distintas tendencias de actores que buscan atacar redes de seguridad.	Entender y analizar los registros (logs) de redes de seguridad, para poder prevenir ataques.
		Enumerar las tendencias más conocidas a ataques de redes y proporcionar descripción para cada una de ellas.
	Realizar un análisis estadístico de las tendencias más utilizadas para atacar redes.	Mostrar algunos análisis de manera visual, tomando conclusiones acerca de ataques a redes de seguridades.

Metodología

Para la elaboración de este trabajo de tesis se va a seguir la siguiente metodología propuesta:

Encontrar las fuentes suficientes para entender de manera holística los conceptos más importantes de trabajo como: sistema SIEM, concepto de red, ataques predeterminados y cómo funcionan, software, hardware, etc.

Elaborar el marco teórico, la justificación y los antecedentes del proyecto. Se debe incluir ya la lista de los ataques a analizar y describir a un sistema SIEM.

Proponer del listado de ataques descritos y estudiados los que se va a encontrar algoritmos para la detección en un sistema SIEM.

Se comienza con el trabajo empírico. Utilizando software como Splunk o Logstash se analiza registros (logs) de firewalls de redes. Se propone analizar logs del firewall de la USFQ.

Luego de tener el trabajo teórico y empírico se realiza el análisis sobre lo que se encontró y sobre lo que se propone (objetivo principal de la tesis).

Se concluye y se recomienda mejoras al proyecto. Se hace un análisis general de todos los componentes del proyecto para ayudar a personas que quieran continuar con dicho análisis en un futuro.

Se revisa el contenido del proyecto y se mejora hasta tener la versión final que será presentada para la graduación del estudiante.

Alcance del Proyecto

En los siguientes capítulos del trabajo se va a dar una descripción de la mayoría de ataques a redes informáticas. Además, se va a proporcionar toda la explicación teórica de cómo funciona un sistema SIEM, para posteriormente detectar técnicas y procedimientos que serán transformados en algoritmos de detección de ataques en un sistema SIEM.

Los ataques que serán analizados con mayor profundidad y presentados en forma de algoritmo para su detección son cuatro: ataque de DoS, ataque de Defacement, ataque de fuerza bruta y ataque de DDoS. De los demás ataques, se mencionará las posibles técnicas de ataque y de detección, no obstante, sin una explicación práctica. Este análisis será cubierto principalmente en el tercer capítulo de trabajo. De los otros cuatro ataques, los procedimientos y técnicas de detección serán descritos en el capítulo cuatro.

Los sistemas SIEM serán descritos con sus componentes más generales y comunes, para después pasar a explicar y argumentar el porqué de la elección de un sistema SIEM en particular que será el utilizado durante el desarrollo del trabajo.

Para concluir se analiza los resultados de los procedimientos, así como también se argumenta ciertas conclusiones de puntos clave y recomendaciones para trabajos futuros que tomen como fuente principal este trabajo, o sean una continuación a técnicas y procedimientos distintas a los expuestos.

CAPÍTULO 2.- REVISIÓN DE LITERATURA

Revisión Literaria

La tecnología y ahora la digitalización de la información hace que los datos sean un activo muy importante de las empresas. Se debe saber cómo proteger los datos para no ser víctima de estafa. "A pesar de que hoy en día las técnicas de seguridad de datos están disponibles para protegerlos y también proteger las infraestructuras informáticas, muchas de estas técnicas tales como firewalls y herramientas de seguridad de red no son capaces de proteger los datos de los ataques planteados por las personas que trabajan dentro de las organizaciones" [7]. No solo es importante proteger los datos del exterior de las entidades sino también elaborar procedimientos para protegerlos de los mismos empleados. De ataques 'desde adentro' como se los conoce.

Hay distintas técnicas para abordar el problema. Por ejemplo: "técnicas de detección de anomalías en los accesos a datos por parte de los empleados. Estas anomalías son a menudo indicativos de potenciales ataques a información privilegiada" [7]. Herramientas tales como SIEM (Security Information and Event Management) "tienen como objetivo recopilar, analizar y correlacionar, en tiempo real, información relevante para la seguridad de una organización. Pueden ser un elemento clave en la búsqueda de una solución para las amenazas internas" [7].

El concepto de ‘Big Data’ cada vez es más popular debido a las cantidades inmensas de información que manejas las compañías. “Grandes compañías generan alrededor de 10 a 100 billones de eventos por día, dependiendo de su tamaño. Estos números tienden a crecer cuando las empresas permitan registros de eventos en más fuentes, contraten más empleados, usen más máquinas y corran más aplicaciones de software. Desafortunadamente, este volumen y variedad de datos se vuelve cada vez más abrumadora” [8]. Según estos autores el análisis de ‘big data’ y su poder de procesamiento en algunos campos han llamado la atención de la comunidad de seguridad por la cantidad de datos que se pueden analizar y correlacionar de manera eficiente. Algo nunca antes visto en la historia. (2013). Aunque el análisis masivo de datos no es la solución a problemas de seguridad, claro que ayuda; y además incentiva a los programadores a continuar innovando para prevenir ataques en la seguridad de redes.

Según Igor Kotenko y Andrey Chechulin (2012)[18] no solo se debe crear sistemas en donde se asume que el ataque común a los sistemas informáticos se basa en una actitud de malhechor. Creen que ese modelo estándar no es compatible para analizar todos los posibles ataques que no tengan un comportamiento similar. “Una nueva arquitectura AMSEC (Attack Modeling and Security Evaluation Component) es propuesta” [18]. Los componentes propuestos para sustentar un SIEM permiten: “encontrar y corregir errores en la configuración de red, revela posibles actos de amenazas a la seguridad, determina recursos críticos de la red y elige una política segura y efectiva para manejar las amenazas” [18].

Con la popularización de sistemas SIEM, hay aspectos que se vuelven más importantes a la hora de medir la efectividad del sistema. “El sistema SIEM ayuda a comprender grandes cantidades de datos acerca de la seguridad de los sistemas. La visualización es una parte esencial de los sistemas SIEM. La arquitectura de los componentes visuales permiten incorporar diferentes tecnologías de visualización que extienden rápidamente la funcionalidad de la aplicación” [24]. Según Novikova la visualización del sistema en general debe proveer un efectivo GUI (interface) para poder interactuar con los diferentes componentes del sistema SIEM. La arquitectura debe permitir una funcionalidad ‘user friendly’ que permita interacción entre componentes y con los usuarios que manejen los sistemas.

El controlar ataques de seguridad en las redes introduciendo sistemas SIEM todavía tiene mucho campo para desarrollarse. Los diferentes sistemas “en la fase de explotación, deben tomar en cuenta los eventos y alertas de seguridad, la configuración de redes en los ordenadores puede cambiar, nuevas vulnerabilidades pueden ser descubiertas, nuevos contra ataques pueden ser desarrollados, nuevos servicios añadidos, y es necesario continuar monitoreando la red, analizando vulnerabilidades visibles y evaluando el nivel de seguridad” [20]. En este artículo se habla acerca de los CVE (Common Vulnerabilities and Exposures) un diccionario que contiene la lista de las vulnerabilidades y exposiciones de los sistemas de seguridad. Es un registro muy grande que se ha logrado obtener gracias a datos recopilados usando SIEM en detección de ataques a redes. (2013). Asimismo “el uso de NVD (National Vulnerability Database) basado en el diccionario CVE es la base para construir gráficos de ataques vía las vulnerabilidades” [20].

Entender cómo funciona un SIEM es esencial para poder desarrollar una investigación más amplia de los ensayos ya expuestos. “Sistemas SIEM utilizan colecciones de datos que se obtienen de sensores ubicados en lugares críticos de la red, que reenvían los eventos más importantes a una base donde se procesan y analizan. Por ejemplo: reporta ataques en tiempo real, hace inventario del manejo y mide el riesgo de recibir ataques” [15]. Se habla en este ensayo acerca de cómo proteger la base en donde se analizan los datos del SIEM. Se presenta un nuevo diseño de firewall que no falla en algunos supuestos escenarios y además resisten ataques externos e internos. (2013). Es claro que el proteger a una red de ataques no solo consisten en hacer más robusto al sistema SIEM sino también fijarse en diferentes componentes de toda la red y buscar combatir la mayor cantidad posible de vulnerabilidades.

Otra manera de combatir el problema de seguridad en las redes es el de enfocarse en cómo proteger los datos y ‘logs’ (registros de datos en archivos .txt) que transmite un SIEM. “Los actuales sistemas SIEM carecen de un lugar donde almacenar los datos de manera segura” [1]. Además se analiza los limitantes de los sistemas forenses actuales donde se almacena la información de los sistemas SIEM. “Proponemos una arquitectura que implementa una variante al algoritmo RSA que supera a los sistemas actuales en cuanto a intrusos y tolerancia de fallas” [1]. Se comenta bastante acerca de OSSIM, el modelo ‘open source’ de los sistemas SIEM.

El motivo de tanto estudio de los sistemas de seguridad SIEM es que muchas veces ningún sistema se encuentra libre de vulnerabilidades que puedan ser atacados. Un

ejemplo concreto es el de los Juegos Olímpicos cuando el departamento de IT sufrió un ataque de múltiples pasos antes de alcanzar el objetivo. “Las capacidades de correlación actuales de los sistemas SIEM basados en un nodo central como servidor, ha sido puesto en evidencia de lo insuficiente e ineficiente que se vuelve al procesar eventos muy grandes de ‘stream’ ” [28]. Las singularidades de los distintos sistemas SIEM creados para distintos propósitos hace posible el avance en la seguridad por distintos frentes siempre buscando mayor seguridad y estabilidad de los sistemas.

Otro caso de análisis es el de la técnica LSA (Latent Semantic Analysis) propuesto en el ensayo de Pavarit Dairinram y otros autores. Se analiza esta técnica debido a que los sistemas SIEM no pueden procesar tantas amenazas distintas. “LSA es propuesto para alivianar los problemas de volumen. Mejora el rendimiento al reducir el ‘ruido’ de grandes cantidades de datos generados en dispositivos. También se usa para detectar patrones en las amenazas basados en eventos y logs. Los experimentos muestran que el enfoque LSA ayuda eliminando datos insignificantes sin eliminar a su vez datos válidos y relevantes” [10]. En el artículo se habla de 3 distintas técnicas para identificar amenazas a la red. Es otro enfoque muy válido al momento de analizar y optimizar los sistemas SIEM.

Otro enfoque más hacia la optimización y eficiencia de los sistemas SIEM es el de crear un sistema no supervisado basado en algoritmos para detectar anomalías. “Nos enfocamos en los requerimientos para un sistema de este tipo y dar una idea de cuan diversos deben ser los eventos de seguridad analizados y unificados para

poder pre procesarlos y obtener algoritmos no supervisados que detecten anomalías en los sistemas" [2]. Detectar conductas extrañas, así como patrones son discutidas en el artículo. El propósito es encontrar correlación entre los eventos de amenazas para entender cómo tratarlos e implementar algoritmos no supervisados.

"The Ontological Approach for SIEM Data Repository Implementation" habla acerca de un enfoque mucho más global (ontológico) para resolver los problemas de seguridad en las redes. "El enfoque ontológico está en auge en muchas áreas técnicas incluyendo la de seguridad de la información. Asumimos que este enfoque es también prometedor para el desarrollo de nuevas generaciones de sistemas SIEM" [19].

Soluciones un poco más específicas para evitar vulnerabilidades de seguridad en los sistemas es el de analizar datos conseguidos en los 'logs' (registros de datos) que son creados por virus (malware software). En algunos casos se ha llegado a comprobar que 'data mining' o minería de datos es un enfoque muy óptimo para detectar patrones escondidos de malware y dar más soporte a los sistemas SIEM[14].

Los IDS (Intrusion Detection Systems) son parecidos y trabajan bajo los mismos parámetros que los SIEM. Muchos problemas para tratar amenazas a los sistemas recaen en que no existe normalización para encapsular eventos similares de manera coherente. Diferentes sistemas utilizan diferentes formatos lo que hace difícil de correlacionar si no se utiliza el mismo sistema. "Se provee una plataforma para normalizar eventos en un súper evento unificado basado en un estándar del CEE (Common Event Expression) desarrollado por la corporación Mitre" [4].

Lo que depara el futuro en cuanto al desarrollo de sistemas SIEM o IDS es el de sistemas conscientes en tiempo real que puedan tomar decisiones al instante. En esto se basa la discusión para la infraestructura del futuro. La defensa cibernética de sistemas que cada vez deben ser más inteligentes para ser más confiables [23].

En conclusión, se necesita un nuevo enfoque que permita leer logs en tiempo real y que sean compatibles con sistemas SIEM o IDS. Proveer a sistemas de ambos tipos a un acceso directo y sin tanta demora hacia los datos en tiempo real y normalizar cada evento que se registre [5].

Marco Teórico

Conceptos generales.

El concepto de red informática, tan utilizado ahora por un gran porcentaje de la población mundial, pero tan poco comprendido a cabalidad, se refiere a un grupo (2 o más computadoras) conectadas entre ellas de alguna manera. Existen muchos tipos de redes, y se las puede describir mediante distintos atributos. Por ejemplo: por su topología, por su arquitectura, por el protocolo utilizado, etc. Con el pasar de los años, la evolución del Internet, que no es más que una red de redes a nivel mundial, ha hecho que la preocupación por la seguridad de las redes incremente. Esto ha llevado a crear software específico para la detección de ataques a redes con la finalidad de evitar el robo de información en formato digital.

Software es alguna forma de organizar información ya sea como sistema operativo, programa o aplicaciones que permiten a las computadoras realizar trabajos. Consiste en instrucciones de código escritas por programadores en lenguajes de programación que luego son ejecutados por los sistemas operativos de las máquinas para hacer alguna función en específico.

Definición de SIEM.

El término SIEM (Security Information and Event Management) se refiere a productos de software que combinan dos servicios distintos: SIM (Security Information Management) y SEM (Security Event Management). Los sistemas SIEM proveen análisis en tiempo real de alertas de seguridad generadas por hardware de redes y de aplicaciones (software). Aunque los acrónimos SIEM, SIM y SEM se han usado muchas veces para referirse a los mismo términos, vale recalcar que SIEM es una fusión de SIM y SEM como una aplicación de software que permite lidiar con monitoreo de red en tiempo real, correlación de eventos, notificaciones de consola, componentes GUI (graphical user interface), reportes de registros de eventos (logs) y áreas de almacenamiento de información.

El término SIEM se lo usa por primera vez en 2005 cuando Mark Nicolett y Amrit Williams describen a un producto por sus características de recolectar, analizar y presentar información de redes y dispositivos de seguridad. Además identificar accesos de administración de aplicaciones, así como sus vulnerabilidades; amenazas de datos externos a un sistema y logs de monitoreo de redes. Este tipo de sistemas de seguridad es una buena alternativa para proveer un acercamiento holístico a la seguridad IT (Information Technology) de cualquier organización.

Muchos artículos científicos se han escrito acerca de cómo optimizar el trabajo de sistema SIEM. El aspecto visual es muy importante. Se han desarrollado varios prototipos de los componentes visuales de dicho sistema, en donde se proponen distintos procesos a seguir para lograr un sistema SIEM robusto y eficaz al momento

de detectar ataques. En los cuales se propone algunos de los sub-sistemas que una SIEM debería proporcionar como interface para el usuario: 1) monitoreo de datos en tiempo real. 2) Trabajar con un repositorio de eventos (análisis histórico, formación de reportes). 3) Creación y edición de reglas operacionales para los módulos de análisis de riesgo, correlación de eventos, modelamiento de ataques. 4) Representación de resultados en modelos de ataques simulados, y contramedidas de selección (de ataques). 5) Administración de incidentes de seguridad y 6) Administración y manejos de recursos. Este es solo uno de los varios enfoques que primero se describirán, antes de pasar al desarrollo de la tesis. Entender que se ha hecho anteriormente ayuda a dar un enfoque mucho más específico a este trabajo, que propone aportar con nuevas soluciones a la detección de ataques en sistemas SIEM.

CAPÍTULO 3.- LOS SISTEMAS SIEM Y LOS ATAQUES DE RED

Introducción a los Ataques de Red

Introducción.

La seguridad es un componente principal para el diseño de una red informática. En la planeación, construcción, diseño y operación de una red, se debe entender la importancia de las fuertes medidas y políticas de seguridad que se deben adoptar.

En el pasado los ‘atacantes’ eran programadores habilidosos que podían explotar vulnerabilidades de las comunicaciones entre computadoras. Ahora, un atacante puede ser cualquier persona que encuentre en Internet herramientas para explotar dichas vulnerabilidades. Estas nuevas herramientas sin costo para una persona común y redes sin políticas de seguridad han incrementado la necesidad para seguridad en redes y tener políticas de seguridad dinámicas. La manera más sencilla de proteger una red de ataques externos es la de cerrarla completamente del mundo exterior. Una red con restricciones permite conectividad solo a conocidos y sitios de confianza. Una red cerrada (closed network) no permite conexiones con redes públicas. Al no existir conexiones a Internet, ataques de afuera se descartan. Sin embargo, amenazas internas también ponen en riesgo la seguridad de las redes. Se estima que el mal uso de la red proviene de dentro de la misma. En la actualidad con el desarrollo de grandes redes, las amenazas a la seguridad han crecido

exponencialmente. Una persona interesada en lanzar un ataque a una red puede tener distintas motivaciones, desde robar información relevante, crear un DoS (Denial of Service) o solo por el desafío que implica quebrar la seguridad de la red.

Desarrollo.

Ataque de phishing.

Ataque por ‘phishing’ es cuando el atacante crea un sitio web falso que luce muy similar al sitio web auténtico. El ataque consiste en que se envía un correo electrónico tratando de engañar al usuario para que ingrese mediante un enlace (link) al sitio web falso. Cuando el usuario intenta hacer un ‘log in’ con su información personal y privada de la cuenta, esa información (nombre de usuario y contraseña) son almacenados por parte del atacante para poder ingresar al sitio web auténtico que el usuario buscaba acceder.

El problema de ‘phishing’ es básicamente que los correos electrónicos enviados por los atacantes son casi idénticos a los de grandes corporaciones, que incluyen logos y formato muy similares a los mails auténticos. El nombre ‘phishing’ significa una manera de confundir al usuario. Como generalmente estos correos son de confirmación de cuentas o de actualización de información el usuario se halla en la encrucijada de abrir el correo y sufrir el robo de sus credenciales o el de ignorar el correo y en cambio verse envueltos en líos administrativos por perder el acceso a cuentas personales o corporativas.

“ ‘Phish’ es un fenómeno en aumento, comprometiendo más de 50% de los casos reportados en Internet por temas de seguridad” [25]. Phishing puede afectar a todo tipo de usuarios de Internet y puede llegar a tener consecuencias desastrosas. Los elementos que se repiten en un ataque de ‘phishing’ son: un mensaje de mail con la apariencia a la de una organización de confianza, una lista de destinatarios extensa, un sitio web similar al original, un nombre y logo similar, una dirección IP del servidor, ‘malware’ que recoge información de los visitantes que acuden al sitio web falso. “El Internet no fue construido con una noción fuerte de lugar y apariencia” [25]. Aunque en la capa 7 del modelo OSI, la capa de aplicación es en donde se codifica y decodifica la presentación de un sitio web junto con HTML (hypertext markup lenguaje). Esto no es una gran garantía para confiar en cualquier sitio web solo por su apariencia. El gran problema está en que en que HTML separa por completo al formato (presentación) del fondo (parte funcional) de un sitio web, y es justo en la parte funcional donde toma fuerza el ‘phishing’.

Por ejemplo, “un mensaje ‘phishing’ de una red social puede ser una simple notificación de que algún amigo tuvo actividad reciente en el sitio; un aparente enlace o enlaces al sitio web pueden estar en el mensaje. Uno de estos enlaces puede conducir al sitio web de phishing, mientras los demás pueden ser auténticos” [25]. En cambio, un mensaje de phishing de un banco o una cuenta de correo indicará al usuario que su cuenta puede cerrarse o bloquearse por distintos problemas. Sim embargo, “mensajes con demandas inexplicables de modificaciones en cuentas pueden ser ignorados, de vez en cuando, quizás, una o dos veces al año el mensaje

puede ser real y requiere la atención del usuario" [25]. La clave está en tener la intuición para reconocer los mensajes fraudulentos y los que no lo son.

Un método bastante obvio es llamar a la institución y preguntar acerca del correo recibido. Otro, es el de utilizar la función 'hover' del mouse, pasar el cursor por el enlace sin darle click y revisar la URL a la cual va a re direccionar el sitio web. Otro buen hábito para evitar estos problemas es el de nunca utilizar los enlaces en estos correos sospechosos, sino que en una nueva ventana de navegador buscar el sitio web del cual proviene el correo y así tener confianza de que la dirección URL es la oficial.

Otra manera de ataques 'phishing' consiste en que el mensaje entregado al usuario contiene información personal correcta, como nombre del empleador, o nombre completo del usuario. Estos ataques dirigidos tienen el nombre común de 'spear phishing'. Grandes empresas como Google por ejemplo, tiene la capacidad de reconocer automáticamente los sitios web de 'phishing'. Proveen a los usuarios una lista de direcciones IP que pueden llegar a ser peligrosas porque se han registrado anteriormente problemas de phishing y mostrarle al usuario una advertencia cuando vaya a navegar en dicho sitio web. Aunque para sitios web completamente dedicados a hacer 'phishing' es fácil reconocerlos, hay veces en que atacantes utilizan parte de sitios web auténticos y en vez de dañarlos los utilizan con propósitos de hacer daño. Los dueños de dichos sitios web no tienen ni idea que están pagando el soporte de hosting para phishing de algún atacante. "Ya que casi la mitad de todos los sitios de phishing son construidos con una herramienta de malware, Avalanche, los

defensores pueden escanear las páginas web en busca de evidencia de código phishing. Los sitios web que den positivo al escaneo van a la lista de sitios web peligrosos por posible phishing" [25]. A pesar de este gran esfuerzo, Google reporta casi 10 millones de advertencias de phishing por día a usuarios y obtienen posibles 10,000 posibles sitios nuevos de phishing diarios. Los sitios web que son utilizados solo para phishing son los que más tratan de ocultarse de las listas de phishing de grandes corporaciones como Google. Estos sitios se registran con muchos nombres distintos en un día, pero como están constantemente cambiando las 'blacklists' se vuelven inútiles.

Una manera distinta y más compleja que utilizan los atacantes para evadir ser detectados de phishing es la de instalar código que chequee conexiones entrantes al sitio web y compararla con la de un 'antiphisher'. Si coincide, envía respuestas benignas a las peticiones de estos sitios que buscan desenmascarar a los sitios web de phishing, esta técnica se conoce como 'IP cloacking'. Los sitios ágiles, en inglés 'agile sites' se refiere a cuando sitios web cambian muchas veces de nombres DNS (Domain Name Server) para evadir 'blacklisting'. Según estudios se asocian a sitios de phishing, entidades que registren varios nombres en el mismo día. Estos nombres son próximos a ser usados pronto. Generalmente en promedio para considerar a un sitio como de phishing se asocian 10 distintos nombres a una misma dirección IP. Aunque la mayoría de phishers intentan engañar a la mayor cantidad de usuarios, hay cierto tipo de phishers más expertos que intentan adentrarse en empresas y conseguir usuarios de altos mandos para sacar más provecho de los engaños. Por ejemplo CEO's o gerentes de IT.

Con el tiempo el número de ataques por phishing se ha incrementado del mismo modo que se ha incrementado la manera de combatir estos ataques. La mayoría de la información estadística que se obtiene sobre estos ataques viene de la fuente oficial de ‘Anti-Phishing Working Group’ APWG que es una coalición de distintas personas que luchan contra estos ataques cibernéticos.

Aunque se han sumado varios esfuerzos por frenar los ataques de ‘phishing’, en los últimos años se observa un crecimiento en el número de estos ataques. Se puede decir que hay dos razones causantes para esto: 1) los usuarios tienen demasiada confianza en herramientas ‘anti-phishing’ y 2) la mayoría de usuarios se cree lo suficientemente inteligente para no dejarse engañar por este tipo de ataques cibernéticos cuando las herramientas adecuadas están funcionando para proteger al usuario. Es por esto, que es relevante el analizar y describir las diferentes técnicas utilizadas en la actualidad para prevenir ataques de ‘phishing’ y en base a eso concluir con las características en común y ver en campo se puede mejorar así como también resaltar lo que se ha hecho de forma correcta.

Phishers utilizan muchas técnicas que incluyen HTML y DOM (Document Object Model).

a) Spoofed anchor (ancla falsa): Lo que el usuario lee en el enlace a la siguiente página a visitar no es lo que ese enlace tiene en el tag como atributo de href, que es el atributo que indica a que página navegar con un click. b) Domain name inconsistency (nombre de dominio inconsistente): una página de phishing muestra inconsistencia entre el dominio real del sitio web y lo que se muestra visualmente en

dicha página. Por ejemplo, la página puede decir: ‘Bienvenido a E-bay’ pero el dominio de esa página no es ebay.com. c) Fake SSl Certificate (Certificados SSL falsos): como muchas organizaciones utilizan conexiones seguras de HTTP, es decir HTTP(s) para transferencia de información, phishers desarrollan sitios web que soportan conexiones HTTP(s) pero con certificados propios no emitidas por los certificadores de confianza. d) Sub-domain usage (uso de subdominios): URL’s de phishing contienen parte de URL’s legítimas. Por ejemplo: la URL del banco NatWest es: www.natwest.com, la URL falsa podría ser www.natwest.com.mjhdr.com. e) Image (Imagen): con imágenes se trata de sustituir porciones legítimas de los sitios web auténticos, por ejemplo una imagen reemplaza al logo o la barra de menú, pero es solo una imagen. f) Customization of status bar (personalización de la barra de status): algunos ataques se enfocan en utilizan funciones del mouse como hover o el atributo del título de un enlace para engañar a los usuarios de que el enlace no es maligno. g) XSS-based form: vulnerabilidades en Cross site scripting (XSS) permite realizar ataques de phishing que recogen datos de los usuarios que introducen en ‘inputs’ falsos que se colocan. Estos son algunas técnicas para realizar ataques de phishing, como contrarrestarlos es ahora el tema a analizar.

Se han identificado 5 técnicas para detectar páginas sospechosas de phishing, estas son: Lista blanca, lista negra, híbrida, independiente y aleatoria [27].

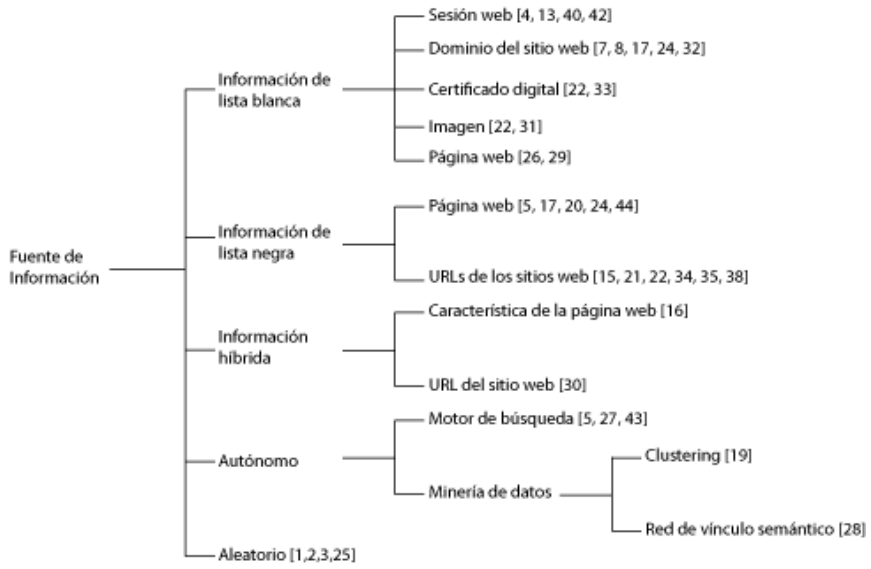

Figura 1. Clasificación de las técnicas de detección de phishing.

1) Información de lista blanca: el análisis de detección se basa en distintas maneras de recoger datos como se muestra en la Figura 1. a) Sesión Web: los usuarios al iniciar sesión en algún sitio web dan sus credenciales confidenciales. Las herramientas anti-phishing comparan las credenciales del dominio con las guardadas anteriormente en el computador del usuario, sino llegan a coincidir, se genera una advertencia. Sin embargo, puede haber casos donde se utiliza el mismo nombre de usuario y contraseña para distintos sitios web que generarán advertencias de 'falso positivos'. Es decir, que la herramienta de software lo ve como una amenaza pero que en realidad no lo son. b) Dominio del sitio web: Los dominios de los sitios web que visita el usuario se almacenan en una 'lista blanca'. Cuando un sitio web pide credenciales, se verifica que el dominio de ese sitio web conste en la lista o que haya resultados parecidos. "Dado que la mayoría de usuarios interactúa con pocos sitios web de confianza. Guardar registros de estos es menos complicado que hacer de los

maliciosos en la ‘lista negra’ de URL’s. Wang desarrolló un plug-in de navegadores que guarda los registros de dominios relacionados a información financiera u otras transacciones sensibles y privadas” [27]. c) certificado digital: la técnica es parecida a la de almacenar los dominios de sitios web de confianzas, solo que esta técnica no necesita inputs (nombre de usuario o contraseñas) que ingrese el usuario pues los certificados digitales los almacena algún administrador y los verifica y compara con otro nuevo certificado que requiere autenticación para crear el vínculo entre el sitio web y el navegador del usuario. d) Imagen: se almacenan logos o imágenes representativas de sitios web auténticos y se las compara con sitios web sospechosos de phishing. e) Página web: se evalúa la similitud de páginas web de la lista blanca con las páginas sospechosas. Se compara el diseño, la plantilla (layout) y la distribución de texto e imágenes. Si coinciden en un alto porcentaje es muy probable que sea una página de phishing.

2) Información de lista negra: algunas técnicas de detección de ataques utilizan información de repositorios en línea con URL’s y páginas web que han sido calificadas como phishing. Existen 2 enfoques de esta técnica: a) Páginas web: una manera de analizar si la página web es sospechosa de phishing es el de hacer un análisis de ‘token’, es decir de las palabras que se utilizan en dicha página web. Previamente en las páginas web que aparecen en la lista negra ya se sabe cuáles son las palabras más utilizadas y se comparan con la nueva página web a analizar. Se califica probabilísticamente y se le da un ‘phishing score’; si excede el valor para ser considerada como una página web de phishing entonces es considerado una amenaza. Una desventaja de este método es que puede consumir muchos recursos

el analizar las palabras de la página web. Una alternativa propuesta es la de utilizar los checksums de MD5 [27]. Se computa estas sumas y se comparan estadísticamente con páginas web anteriormente reportadas como phishing. b) URL del sitio web: otro enfoque consiste en extraer información de repositorios de lista negra de las cadenas de caracteres en las URL de sitios web. Con este análisis se provee pistas de los administradores de dichos sitios web y se puede inferir si hay sitios web en servidores de phishing. “La edad del dominio también es una característica importante para clasificar un sitio web como phishing” [27]. Se utiliza la herramienta WHOIS en donde se puede encontrar información relevante sobre el sitio web y a quien pertenece.

3) Información híbrida: esta técnica recae en información de lista blanca y negra donde el objetivo es utilizar clasificadores para que nueva información generada recaiga en listas blancas o negra dependiendo de lo que se encuentre. Se utilizan ‘learning algorithms’. a) Características de la página web: dependiendo las características de cada página web y utilizando algoritmos de SVM (support vector machine) se clasifica los sitios web en legítimos y de phishing. Los parámetros para esta clasificación son: enlaces, imágenes, archivos de descarga, identidad. b) URL del sitio web: se pueden clasificar distintos sitios web en base al léxico y de donde provienen (es decir, el host de cada sitio web).

4) Método Independiente: esta técnica se caracteriza por analizar páginas web descargadas y utiliza técnicas de minería de datos para determinar si es legítima o no. Existen dos técnicas principales. a) Máquina de búsqueda: se utilizan máquinas

de búsqueda confiables como Google para encontrar información acerca de una página web. Información detallada del dominio, se compara las URL, y si no coinciden ciertos datos entonces puede ser una página web sospechosa. Se analizan los términos más frecuentes y el inverso los términos menos frecuentes en una página web (es decir, los términos más importantes, pues se mencionan en menor cantidad) para el análisis respectivo. Hay una técnica nueva todavía poco popular que consiste en identificar textos al escanear una 'imagen de pantalla' de la página web. Los textos obtenidos se los pasan a Google y si no existen coincidencias con los dominios legítimos entonces es detectada como phishing. b) Minería de datos: existen dos técnicas para la detección de phishing: clustering (agrupamiento) y enlace de red semántico (SLN). "En la técnica de clustering, una página web es analizada para encontrar clusters con respecto a ciertas características. Por ejemplo: similitud de texto, enlaces, plantillas de diseño… La presencia de un cluster indica que la página web es sospechosa de phishing y el centro del cluster representa el objetivo legítimo de una página web que se quiere clonar por phishers" [27].

5) Aleatorio: este enfoque asume que algunos usuarios eventualmente van a caer en la trampa de phishing por lo que se provee credenciales aleatorias no auténticas antes de dar las legítimas. Esto hace que el identificar las credenciales verdaderas sea extremadamente difícil. Un sitio web es considerado como phishing si la respuesta a dichas credencias falsas y verdaderas son siempre de éxito o fracaso sin nada más de texto. Este método puede no ser muy conveniente cuando sitios web de phishing permiten limitados intentos de autenticación.

Básicamente estos son los métodos más importantes de cómo identificar y contrarrestar ataques de phishing. Saber utilizar las herramientas anti-phishing así como interpretar la información es crucial para un análisis correcto que sirva a futuro a miles de usuarios para que no caigan en estas estafas cibernéticas.

Ataques de DDoS y DoS.

Las anomalías en el tráfico de red son muy comunes hoy en día. Inundar una red con paquetes 'basura' busca que paquetes de información relevante no lleguen a los destinatarios pertinentes. Esta forma de ataque a redes informáticas es la más común. Los dos tipos de ataques más comunes por inundación son: DoS (Denial of Service) y DDoS (Distributed Denial of Service).

DoS se ha convertido en una de las más grandes amenazas a servicios de Internet y transacciones comerciales. Estos ataques son causados al explotar vulnerabilidades o inundar una red. En cualquiera de los dos casos el objetivo del atacante es el de hacer un recurso no disponible para los usuarios. Como la inundación se debe a información 'basura', los auténticos usuarios deben soportar un DoS. En general, existen dos formas de detectar este tipo de ataques: por firma o por anomalía. Por firma el sistema tiene un conocimiento previo del ataque venidero y lo compara con características de ataques pasados. Sin embargo, este método resulta inútil cuando se compara ataques que no han sido registrados anteriormente. En detección por anomalía el sistema logra la detección al encontrar desviación significante entre el tráfico normal y el tráfico que ocasiona el DoS. "Los sistemas de detección de

intrusos tradicionales (IDS) utilizan análisis al nivel de paquetes para detectar ataques maliciosos por cualquiera de los dos métodos mencionados" [12]. Información de protocolo que almacenan los paquetes en los encabezados se utilizan para identificar el emisor del paquete y si tiene algún comportamiento malicioso. Los métodos convencionales para detectar este tipo de ataques son muy pobres al momento de diferenciar entre un ataque DoS o DDoS. Mientras el primero es iniciado por un solo atacante, el segundo en cambio, es lanzado por varios atacantes. Si ambos atacantes son tratados de manera similar se corre el riesgo de causar falsos positivos al sistema ya que muchos agentes en un ataque DDoS no son los que inician el ataque, pero son tratados como iniciadores. Es necesario encontrar un nuevo enfoque que separe para el respectivo análisis a estos dos tipos de ataques.

El tráfico de TCP (Trasfer Control Protocol) exhibe una periodicidad en su PSD (Power Spectrum Density) cuando el paquete llega. Una señal de que un ataque está en marcha puede ser la falta de dicha periodicidad. "El PSD de varias fuentes en ataques DDoS son distribuidos a una frecuencia baja comparados a ataques DoS de un solo atacante" [12]. Utilizando el análisis de la frecuencia del tráfico de TCP se puede obtener los siguientes resultados:

En este ejemplo se utilizó una red local de 23 computadoras y un router. De las 23 computadores 22 actúan como 'zombies' y solo una de ellas es la víctima que además actúa como servidor de algún servicio en particular. Para simular un ataque DoS solo una computadora zombie contribuye al ataque, mientras que en un ataque DDoS las 22 computadoras zombies participan. Para simular los ataques se utiliza el

protocolo UDP (User Datagram Protocol), un protocolo de red no orientado a la conexión. Un ataque por inundación de UDP puede iniciarse enviando muchos paquetes a distintos puertos de un mismo host de manera aleatoria. La tasa de paquetes enviados para este ejemplo es de 1 milisegundo.

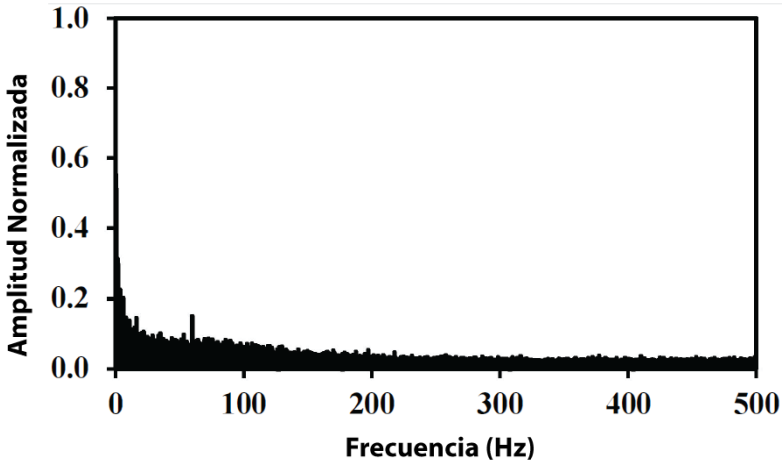

Figura 2. Amplitud del espectro del flujo de TCP normalizada.

La Figura 2 indica que la energía de TCP está distribuida en diferentes frecuencias. No hay una frecuencia dominante para el tráfico normal de paquetes.

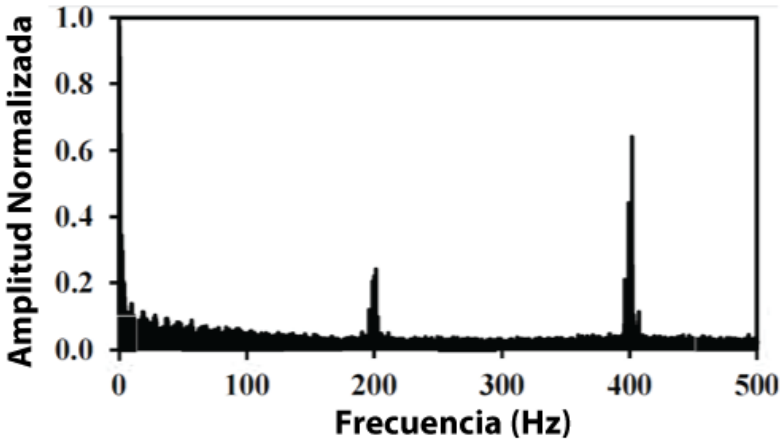

Figura 3. Amplitud del espectro de un ataque DoS.

En la Figura 3 en cambio que muestra el espectro durante un ataque DoS se puede apreciar diferencias con el tráfico normal a altas frecuencias.

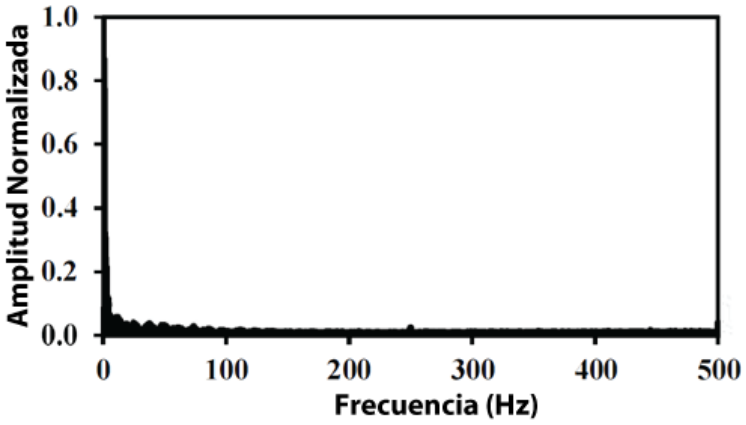

Figura 4. Amplitud normalizada del espectro durante un ataque DDoS.

La distribución del espectro para un ataque DDoS es diferente al normal y al de DoS. Como se aprecia en la Figura 4. A altas frecuencias casi no hay movimientos pero en cambio a bajas frecuencias haciendo un zoom a la escala se encuentra distintos picos en las frecuencias más bajas de 0Hz a 50Hz como se aprecia en la Figura 5.

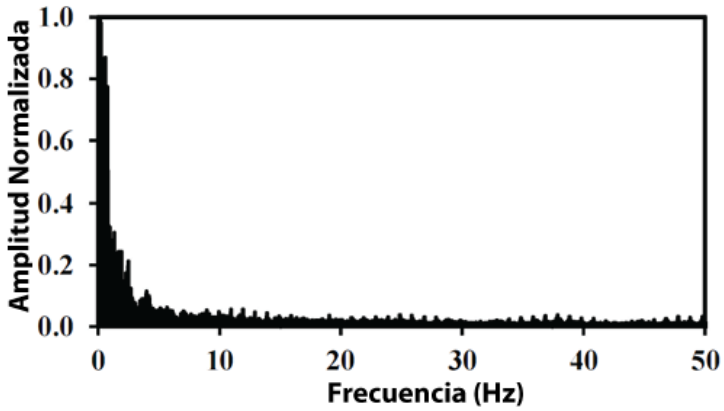

Figura 5. Amplitud del espectro de un ataque DDoS a bajas frecuencias.

Como ya se ha mencionado un ataque de DDoS puede ser tan potente que puede acabar con el ancho de banda del objetivo. Basado en el tipo de objetivo estos tipos de ataques pueden ser de dos niveles: de aplicación o de red.

Las aplicaciones de red generalmente tienden a dejar abiertos ciertos puertos para comunicación lo que les hace débiles para afrontar ataques de DoS. “Considerando adversarios que pueden espiar y lanzar ataques directos de DoS a puertos abiertos de aplicaciones, se ha sugerido la solución de ‘pseudo-random port-hopping” [13]. Donde las aplicaciones se defienden de estos ataques cambiando sus puertos de comunicación periódicamente. Esta solución es bastante compleja debido a la sincronización que debe haber entre servidor-cliente para que la información de los puertos habilitados llegue a tiempo. Se han propuesto dos algoritmos para evitar estos fallos de sincronización. El primero el HoPerrAA que permite a cada cliente interactuar con el servidor de manera independiente. El segundo algoritmo es BigWhell que en cambio permite comunicación del servidor con múltiples clientes, evitando la sincronización individual.

Sin embargo, estas soluciones se vuelven obsoletas cuando el ataque DoS es dirigido hacia congestionar la red de la víctima. Para este otro tipo de problemas la solución propuesta es la de tener mecanismos de ‘network-capability’ (capacidad de red) que puede ser chequeada por routers para saber si el tráfico de paquetes es legítimo o se trata de algún ataque. Aun tomando esta medida, muchos malhechores pueden solicitar establecerse en la capacidad de la red causando posibles ataques de DoC (Denial of Capability). Se propone un algoritmo para mitigar ataques de DoC.

El algoritmo divide la capacidad de un servidor para manejar capacidades en quotas. Estas ‘quotas’ se sitúan dependiendo de la arquitectura de árbol a la que correspondan. Este algoritmo no es solo utilizado para enfrentar problemas de DoC sino también problemas directos de DoS.

Ataque basados en contraseñas.

Las contraseñas (passwords en inglés) son el método más común de autenticación debido a la conveniencia y la facilidad para que los usuarios lo utilicen , además de la fácil implementación por parte de los programadores. Si existen otras alternativas de autenticación como tokens de hardware que generan en tiempo real códigos para garantizar acceso o certificados SSL/TLS. Ninguno de estos métodos se ha esparcido en las aplicaciones comerciales debido a los altos costos en infraestructura y mantenimiento. A pesar de que los sistemas basados en contraseñas son convenientes, tienen sus puntos débiles. Por ejemplo, las personas tienden a elegir contraseñas muy fáciles y cortas. Por el contrario, contraseñas que las crea los servidores son muy difíciles de recordar y poco amigables (user-friendly).

Este tipo de sistemas basado en contraseñas para la autenticación sufre dos tipos de ataques: en línea y no en línea. En los ataques no en línea (offline), el atacante escucha el canal de comunicación y registra datos. Posteriormente, sin entablar comunicación con el servidor prueba las contraseñas que registró. Este tipo de ataques pueden ser prevenidos usando una llave pública de criptografía.

De otra forma, en los ataques ‘online’ el adversario intenta las posibles contraseñas para ingresar al sistema. Existen dos modalidades en cómo se presentan los ataques, la primera es la de fuerza bruta, ya que un programa intenta todas las palabras posibles para lograr acceso a la cuenta. La segunda, en cambio, es la de un ataque de diccionario, se intentan todas las palabras de diccionario como contraseña. Hasta el momento, no han surgido métodos para defenderse de este tipo de ataques en línea.

ATT (Automated Turing Test) es un tipo de pregunta/desafío que es usado para diferenciar entre un humano y un robot. Estas pruebas son fáciles de generar por la computadora, pero muy complicadas de que una computadora la resuelva. Por lo tanto, si la respuesta a la pregunta es correcta, se presume que si se está lidiando con una persona. Existen varios tipos de ATT, el más conocido el CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humas Apart). Este tipo de prueba consiste en pedir al usuario que escriba letras o dígitos de una imagen distorsionada que aparece en la pantalla del computador. Existe también un protocolo propuesto llamado PGRP (Password Guessing Resistant Protocol) que limita el número de intentos para iniciar sesión de un usuario a uno solo. Posteriormente lo desafía con una prueba de ATT. Este protocolo previene los ataques en línea de diccionarios. A continuación se dan a conocer sistemas para prevenir ataques en línea.

Los sistemas basados en contraseñas son vulnerables ataques en línea de diccionarios debido a que las contramedidas utilizadas son generalmente muy

costosas y no muy efectivas. Algunas de las medidas son: a) Bloqueo de cuenta: consiste en permitir al usuario un número limitado de intentos para iniciar sesión. Si todos los intentos son erróneos la cuenta se bloquea. Esto previene ataques de tratar adivinar la contraseña auténtica pero en cambio abre las puertas para que ataques de DoS o DDoS se den con frecuencia, ya que si el sistema se bloquea se logra el objetivo de que el usuario legítimo no ingrese al sistema. Además se corre el riesgo que realmente el usuario no recuerde su clave y por bloque de sistema deba esperar un tiempo (generalmente mayor a 1 día) para que su cuenta vuelva a estar habilitada. b) Respuesta demorada: consiste en que el servidor demora su respuesta a la petición. Con esto se previene al atacante de chequear muchas contraseñas en poco tiempo, y demorarse mucho más tiempo que el planeado. c) Protocolos Pinkas y Sander: el protocolo introducido por Pinkas y Sander (PS) consiste en requerir la respuesta de un desafío ATT antes de continuar a iniciar sesión ingresando nombre de usuario y contraseña. Si se falla el desafío no se puede continuar con el acceso a la sesión. Aunque es un buen método para evitar ataques de diccionario, en cambio también cada auténtico usuario debe responder cada vez una pregunta de ATT, creando molestias y requiere trabajo del servidor en generar un código ATT diferente para cada ingreso al sistema. d) Protocolo Van Oorschot y Stubblebine (VS): es considerado un protocolo mejorado que PS. Consiste en que solo si los datos de nombre de usuario y contraseña son incorrectos, entonces se pide responder correctamente una pregunta de ATT. A veces el problema consiste en que a usuarios legítimos se les pide ingresar muchos desafíos ATT lo que causa molestias. Muchas

técnicas actuales incluyen en sus métodos ATT's debido a la presunción que estos desafíos son muy difíciles para robots pero en cambio muy sencillos para humanos.

El protocolo PGRP antes mencionado tiene la tarea de cumplir algunos objetivos: 1) debe lograr que ataques de diccionario y de fuerza bruta sean ineficientes. 2) El protocolo no debe causar daño en la usabilidad de los usuarios. 3) El protocolo debe ser fácil de desplegar y escalable, requiriendo mínimos recursos computacionales en términos de memoria, procesamiento y espacio en disco [17]. PGRP fue diseñado en base a los protocolos PS y VS previamente explicados. Consiste en utilizar una prueba de ATT solo cuando el usuario haya ingresado mal el nombre de usuario o la contraseña. Todo el procedimiento se especifica en [17] pero básicamente la metodología consiste en 3 pasos primordiales: 1) el registro de nuevos usuarios y el inicio de sesión. Se recolecta datos personales además de los auténticos para nombre de usuario y contraseña utilizados para compararlos más adelante. 2) Identificando al usuario. Se procede a dejar que el usuario ingrese sus datos para autenticar la sesión. Si falla se le exige que apruebe un desafío ATT si falla más de 3 veces queda inhabilitado. 3) Se clasifica al usuario. Puede tener status de válido, inválido, correcto (si falla en la autenticación pero si logra aprobar la prueba de ATT) o incorrecto si falla en la autenticación y además las pruebas de ATT.

Aunque los ATT son buenos para limitar los intentos que los atacantes pueden tratar de adivinar para ingresar en la cuenta de algún usuario, tiene sus restricciones en cuanto a la eficiencia y la conveniencia desde el punto de vista del usuario. Utilizar una llave encriptada de intercambio es otro enfoque para solucionar problemas de

sistemas de contraseñas. El método consiste en una combinación de asimetría con una llave pública y simetría con una llave privada, donde la llave privada se utiliza para generar la llave pública encriptada donde solo el emisor y el receptor tienen el 'pass-code' para descifrar la llave. Este método conocido como EKE (Encripted Key Exchange) protege las contraseñas de ataques de diccionario offline.

Para entender cómo funciona el método EKE supongamos que A y B comparten un secreto P. Para establecer una conexión segura, A genera una llave aleatoria R basada en P y la envía, dando como resultado P(R) a B. Este mecanismo en esencia es el mismo que se utiliza en la autenticación Kerberos [6]. En (número de referencia de este artículo) se especifica más a detalle este proceso explicado brevemente aquí.

El objetivo principal de las soluciones planteadas para problemas de sistemas basados en contraseñas es el de proteger a usuarios con contraseñas muy débiles. Lastimosamente, en el mundo que se vive, esto es real y se ha demostrado empíricamente. Se ha fallado en los varios intentos por presionar a la gente a que fortalezca sus contraseñas y más bien en experimentos hay máquinas que de una base de datos de 15,000 passwords, se logró hackear el 25% aproximadamente.

Ataques de diccionarios.

Aunque cuando se trató los ataques a sistemas de autenticación con contraseñas se mencionó los ataques de diccionarios, en esta sección se enfoca un

poco más este tipo de amenaza hacia los servicios de SSh (Secure Shell). Los ataques de diccionario a SSH se han detectado de dos maneras: los que se soportan en archivos de registros (logs) y los que se soportan en tráfico de red. Para contrarrestar la primera manera de ataque se analiza los logs disponibles para reconocer entre intentos de autenticación de éxito o de fracaso. En grandes redes, esto implica grandes costos en administradores que se dediquen al monitoreo que crece igual al número de hosts en la red. Para neutralizar los ataques de tráfico a la red en cambio es monitorear a todos los usuarios (hosts) de la red y bloquear a los que muestren conexiones masivas en ciertos momentos. Aunque los costos son mucho menores, en este enfoque no se pueden distinguir entre ataques exitosos y no exitosos. Desafortunadamente, ambos enfoques son inefectivos en ataques sigilosos. “Un método ideal detectaría un ataque individual y distinguiría entre un éxito o un fracaso usando solo información derivada del tráfico de red” [26].

Para entender mejor esta amenaza se define más claramente un ataque de diccionario SSH. Estos son los intentos de adivinar el par nombre de usuario/ contraseña para ganar acceso de forma fraudulenta a una sesión en específico. El atacante confía en el hecho de que muchos usuarios eligen contraseñas fáciles de adivinar. Cuando comienza a poner en práctica los intentos hasta encontrar el correcto, los ataques ya han contaminado logs e inundado el tráfico de red. SANS (SysAdmin, Audit, Network, Security Institute) ha reportado el surgimiento de un nuevo tipo de ataque: ataque de diccionario distribuido de SSH. Estos ataques son mucho más sigilosos y realizados por varios atacantes en coordinación, generalmente lo hacen los bots. Intentan los ataques desde distintas máquinas por lo

que tienen un impacto casi nulo en logs y tráfico de red, por lo que es más complicado detectarlos.

El método que se propone para detectar ataques SSH de diccionario se basa en dos elementos: el primero en la existencia de una conexión mediante un protocolo. Es decir, que se pueda estimar si el par nombre de usuario / contraseña son aceptados en la autenticación, y poder distinguir entre un ataque exitoso o no. El segundo elemento es 'la diferencia del tiempo de arribo de una paquete de autenticación'. Un criterio para estimar si el par ingresado fue realmente ingresado por un usuario o no. En [26] se explica mejor el procedimiento que se adoptó para llegar al método incluso con los resultados empíricos. La Figura 6 es un resumen bastante simple del método detección de ataques de diccionario SSH basado en análisis de flujo.

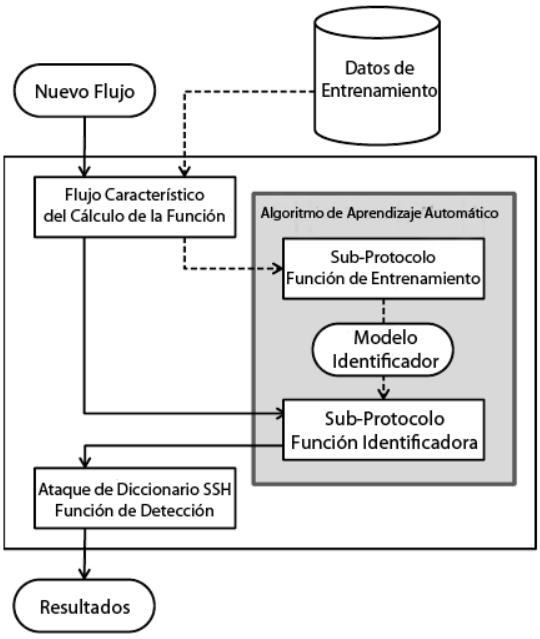

Figura 6. Método de detección de ataques de diccionario SSH.

En la Figura 6 se pueden apreciar 4 funciones que son la estructura para el método de detección propuesto. 1) Flow Feature Calculation. 2) Sub-protocol training. 3) Sub-protocol identification. 4) SSH dictionary attack detection [26].

Ataque de Man-in-the-Middle.

Como el número de usuarios de VoIP (Voice IP) incrementa, y el despliegue de servicios de los SIP (Session Initiation Protocol) gana terreno, la seguridad se ha incrementado. VoIP es un tipo de aplicación corriendo en la red que ha heredado los problemas más comunes de IP, como la amenaza a la seguridad. Los ataques de 'hombre en el medio' (Man in the Middle Attack) como su nombre lo indica ocurre cuando hay alguien en el medio de una comunicación cliente-servidor monitoreando, capturando y controlando los datos que se transmiten. Un ataque de este tipo puede hacer que los datos sean re-dirigidos a otro computador sin que ninguno de los dos participantes de la comunicación se dé cuenta.

"Los ataques de MitM pueden reemplazar a uno o los dos involucrados en la comunicación o convertirse en un intermediario, donde puede reemplazar a cualquiera a ambos involucrados en la comunicación y retrasmitir datos" [9]. Muchas veces ataques de DoS son lanzados por MitM escuchando comunicaciones en SIP de VoIP. Estos ataques tienen el nombre común de ataques MitM-DoS. En lo que se basan los ataques MitM-DoS es en insertar mensajes 'especiales' elaborados por el atacante de MitM, interrumpir la comunicación entre el servidor de VoIP y un usuario

y luego lanzar un ataque DoS ya sea hacia el servidor o hacia el agente usuario. También hay la posibilidad de que se intervenga en una comunicación entre dos usuarios, esto incluso es más apetecible para MitM ya que generalmente si no es un servidor con el que se comunica un usuario no pide autenticación. Una vez que el MitM este en medio de la comunicación, puede recoger la información que se transmite, o cambiarla. Este tipo de ataque cuando una conversación puede ser manipulada por otro actor que se ha logrado establecer en el medio de dos usuarios es muy peligroso por el alcance que tiene el atacante.

Ataque de inyecciones SQL.

Un ataque mediante inyecciones SQL (Structured Query Language) es una técnica de inyección de código generalmente a aplicaciones que se manejan con soporte de base de datos. El código malicioso de sentencias SQL se insertan para ejecución. Este tipo de ataques explota las vulnerabilidades de seguridad y es conocido como un ataque de ‘vector’ en sitios web o como un ataque común en bases de datos que manejen el lenguaje de SQL. Este tipo de ataques se encuentra entre los 10 inconvenientes de seguridad más preocupantes. El SQLIA (SQL Injection Attack) es más común a aplicaciones web que a cualquier otro tipo de aplicación. “Acorde al reporte de White Hat de vulnerabilidades en seguridad web del 2011, muestra que 14-15% de los ataques hacia aplicaciones web son del tipo SQLIA” [3].

SQLIA se da principalmente por validación errónea de inputs del usuario. Para prevenir este tipo de ataques hay ya soluciones que consisten en prácticas de códigos defensivos con encriptación de algoritmos basados en la aleatorización (randomization). Muchas veces la programación defensiva es de ardua labor y no muy efectiva previniendo ataques de SQLIA. Por qué los atacantes eligen este SQLIA es debido a que pueden ganar acceso ilegal a bases de datos con lo que pueden extraer información, ganar privilegios, modificar registros de la base existente o enfocarse en lograr el mal funcionamiento de una aplicación en particular.

Para visualizar como se puede dar un ataque por inyección de sentencias SQL este ejemplo es muy claro: cuando se pida en un formulario que el usuario ingrese nombre de usuario y contraseña, la sentencia SQL a comprobar es: ‘SELECT * FROM tablename WHERE user = ‘’ and password = ‘’ ’. El atacante al ingresar en los campos lo siguiente: “OR ‘1’=’1’ --” puede tranquilamente ganar acceso sin autorización al sistema pues, la sentencia a comprobar como verdadera si lo es. Esto sucede porque 1 = 1 (verdadero) y el – representa que todo después de aquello es un comentario. De esta manera, si los inputs del usuario no han sido saneados de una manera correcta, se vuelven una amenaza a la seguridad de la aplicación.

Existen distintos tipos de ataques de inyección de SQL. El primero utilizando tautología como el del ejemplo presentado en el párrafo anterior. También se puede utilizar ‘queries’ (pedidos) ilegales o incorrectos para que la base de datos y el sistema como retroalimentación retorne información importante sobre cómo está estructurada la base. También los atacantes pueden utilizar queries conocidas como

‘piggy-backed’, donde se incluyen en los inputs caracteres especiales como ‘,’ ‘;’ ‘--’ para modificar o borrar registros de la base de datos. El último tipo de ataque, un poco más elaborado son conocidos como inyecciones ciegas, ya que el atacante al no recibir información de cómo está estructurada la base introduciendo queries incorrectas, comienza a realizar ataques progresivos y puntuales para analizar el comportamiento de la seguridad de la aplicación.

Existen ya varios enfoques para evitar este tipo de ataques. Por ejemplo, se utiliza validación del lado del cliente de la siguiente manera como indica la figura:

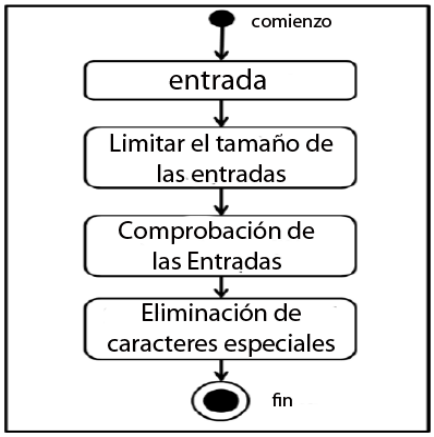

Figura 7. Diagrama de actividad para la validación del lado del cliente.

Aunque muchos de los pasos que se muestran en la Figura 7 son válidos e implementados con JavaScript, no son efectivos para todos los tipos de ataques. Por ejemplo evitan mucho trabajo al servidor, e impiden ataques del tipo de introducir queries incorrectas y el de tautología pero en cambio, son nulas contra las inyecciones ciegas.

Otra alternativa para prevenir ataques del tipo SQLIA es el algoritmo ‘Random4’ basado en la aleatorización, que se usa para convertir los textos de input del usuario en texto cifrado. Un input introducido por un usuario puede contener minúsculas, mayúsculas, caracteres especiales, etc. por lo tanto el cifrado se hace de manera aleatoria. En [3] se dan detalles más específicos de cómo realizar la aleatorización de dicho cifrado.

Esta alternativa de prevenir ataques no solo se la utiliza en sistemas para textos introducidos por usuarios sino para todo tipo de texto. Por ejemplo, la tabla ‘user’ de la base de datos según una randomización del algoritmo viene a ser user = userA2;h. El tiempo requerido para hackear estos textos cifrados son años. Para que el ataque del tipo ‘piggy-backed query’ funcione se necesita saber los nombres de los campos y tablas de la base de datos, sin eso es muy difícil de des encriptar y desplegar el ataque, por lo que este tipo de ataques se logran prevenir de manera exitosa. Del mismo modo de cómo se encripta nombres y campos de la base de datos, para ataques ciegos de SQL se puede encriptar el input introducido en los formularios. De esta manera las sentencias SQL no tendrá sentido para el programa y no se puede realizar este tipo de ataques tampoco. La Figura 8 muestra los pasos y los procesos que son parte del sistema con el método de ‘Random4’:

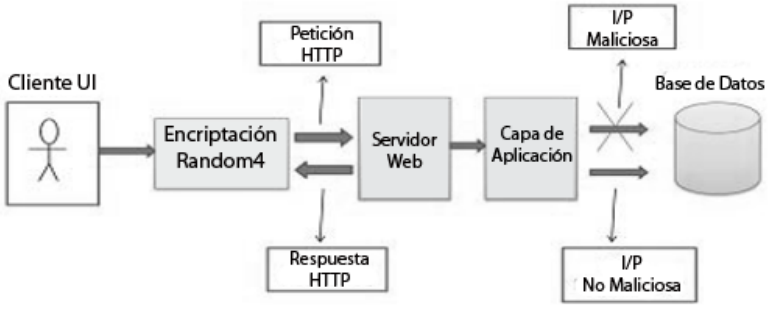

Figura 8. Diagrama general del sistema Random4.

Síntesis.

Con este análisis se concluye una visión corta y concisa de los ataques con los que amenazan la seguridad de aplicaciones de red. Se han mostrado varios tipos de ataques y se ha propuesto maneras ya establecidas de cómo neutralizarlos, así como también de cómo reconocerlos y saber identificar sus debilidades y fortalezas. El análisis continúa ahora con los frameworks ya propuestos de sistemas SIEM. El objetivo de este trabajo es el de proponer algoritmos de detección de ataques en este tipo de sistemas, por lo que esta introducción a los ataques es válida para definir e identificar las amenazas existentes en este tipo de sistemas.

Introducción a los Sistemas SIEM

Existen distintos tipos de frameworks que permiten a desarrolladores tener información en tiempo real sobre todo tipo de eventos concernientes a la seguridad. A muchos de esos frameworks se los conoce como sistemas SIEM por sus funcionalidades que permiten análisis de información y manejo de eventos en simultáneo y bajo módulos ya estructurados que muchas veces trabajan juntos para lograr informes detallados sobre posibles ataques a la seguridad y monitoreo de red.

Introducción a ELK.

Para este trabajo y la detección de los ataques de red que se presentarán de manera más detallada más adelante, se utilizó el software 'open source' Logstash. Este software es una herramienta para la administración de logs (registros). Logstash facilita el trabajo en la recolección y tratamiento de los datos que pueden venir de cualquier fuente. Estos archivos llamados comúnmente ´logs' pueden ser simples mensajes del sistema o datos sobre eventos que se registran como una bitácora de todos los acontecimientos que van ocurriendo en la vida útil de una máquina. El crecimiento de Logstash lo hace muy flexible a utilizar, además de que cuenta con 165 plugins que ayudan a refinar el análisis de datos [22]. Cómo funciona Logstash es bastante fácil de entender. Como todo software, recibe entradas (inputs), luego procesa y al final crea salidas (outputs). De manera ilustrada se asemeja a la siguiente figura:

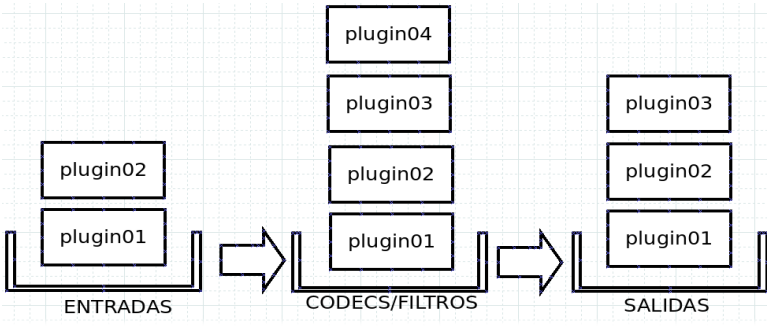

Figura 9. Arquitectura simple del funcionamiento de Logstash.

A los procesos en Logstash se los conoce como codecs o filtros. Una entrada es un plugin que utiliza Logstash para extraer datos de algún sitio. Estos datos son tratados con los codecs/filtros para al final obtener resultados y dárselos a los plugins de salida.

Aunque Logstash se puede definir como un repositorio de logs, el trabajo se complementa con lo que hace 'Elasticsearch'. Este software es un servidor de búsqueda RESTful también de open source desarrollado en Java y distribuido bajo la licencia de Apache. Elasticsearch puede ser usado para buscar cualquier tipo de documento. Provee escalabilidad de búsqueda, búsquedas en tiempo real y además permite a los usuarios el organizar los datos en distintos clusters que pueden ser consultados como grupos o independientemente. Es un analizador de todos los logs que se encontraron con Logstash, y lo hace al instante.

Logstash y Elasticsearch se complementan bastante bien tomando en cuenta las tareas que realizan. El último software de ELK es Kibana. Este software fue diseñado para trabajar conjuntamente con Elasticsearch y darle una forma a los datos analizados. Hace un gran trabajo para entender análisis de grandes volúmenes de

datos con gráficos de todo tipo que se construyen en tiempo real. Cada set de gráficos o análisis que se crean son fáciles de guardar y compartir para una comunicación más efectiva con otros usuarios.

ELK tiene un tremendo poder al momento de hacer monitoreo de eventos y análisis de información para obtener resultados útiles para la toma de decisiones. Se puede resumir el cómo funcionan en la siguiente figura:

Figura 10. Ilustración gráfica de las funciones de ‘ELK Stack’.

Con este breve análisis sobre el software a utilizar se profundiza en el siguiente capítulo del trabajo sobre como con la utilización de ELK se puede llegar no solo a descubrir ataques informáticos a redes, sino además como prevenirlos de manera rápida y eficiente. ELK y en general los sistemas SIEM cada vez son más utilizados para el análisis de tráfico de datos y el monitoreo constante de la red. Logstash puede ser instalado en una máquina y servir solo a monitoreo a un servidor local, como también puede ser instalado en un servidor centralizado y re direccionar todo el tráfico de datos hacia allá y filtrar esos datos y obtener información valiosa que requieren los usuarios.

¿Por qué utilizar ELK?

ELK permite una consolidación de las distintas herramientas en softwares compatibles entre ellos. El hecho de que Logstash entable una comunicación tan rápida y sencilla con Elasticsearch para la indexación de datos y que luego eso pueda ser visualizado mediante la interface web de Logstash: Kibana es un beneficio que tal vez otros sistemas SIEM no lo tienen.

Muchos de los sistemas SIEM que se analizaron anteriormente en este trabajo no tienen la facilidad ni la volatilidad para adaptar el uso de la recolección de datos como ELK lo permite. Además se tiene en el sitio oficial una documentación bastante amplia y además una comunidad de desarrolladores que hablan de errores actuales que presentan las herramientas. Como anécdota personal, mientras desarrollaba este trabajo, al descargar la herramienta del sitio oficial y registrar la causa del trabajo (tesis de pregrado); el equipo de desarrolladores de Logstash envió un correo electrónico para averiguar personalmente el uso que se le estaba dando a la herramienta. Esto hace aún más notorio el trato personalizado por parte del equipo de trabajo de ELK.

CAPÍTULO 4.- DESARROLLO

Introducción

La parte práctica de esta investigación se realizó en una máquina con sistema operativo de distribución Linux (Ubuntu). Como se explicó en la sección anterior, ELK es un sistema 'open source' y como bien pudo haber funcionado en Windows u OSX, en una distribución Linux hay mayor facilidad para manipular la escritura y lectura de archivos que deben ser creados de manera dinámica por scripts.

Logstash es un paquete de software que ya contiene ciertos plugins pre-instalados aunque tiene una documentación muy bien descrita para la personalización de filtros nuevos si es que el usuario los cree necesarios. Ya con Logstash la interface web Kibana viene como complemento. Elasticseach, que se encarga del indexamiento de los archivos que se recopilen en Logstash es otro paquete de software independiente que complementa a Logstash. Una vez instalados ambos paquetes de software es necesario dar los permisos respectivos para que puedan correr en el servidor local de la máquina.

Para iniciar a recopilar archivos de registros (logs) con Logstash, se puede hacer de distintas maneras. En este trabajo se realizó principalmente desde archivos de configuración externos que se cargaban a la ejecución de Logstash. También a modo de prueba se referenció a los logs desde la consola. Para efectos prácticos de eficiencia los archivos de configuración externos ya tienen cargados los parámetros

necesarios para que Logstash filtre los archivos de la manera que cada usuario quiera.

En la siguiente figura se aprecia un archivo de configuración cualquiera para Logstash:

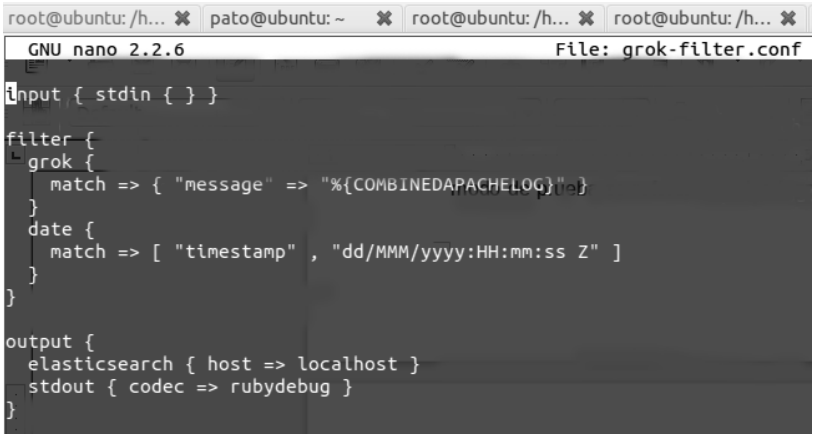

Figura 11. Archivo de configuración genérico de Logstash.

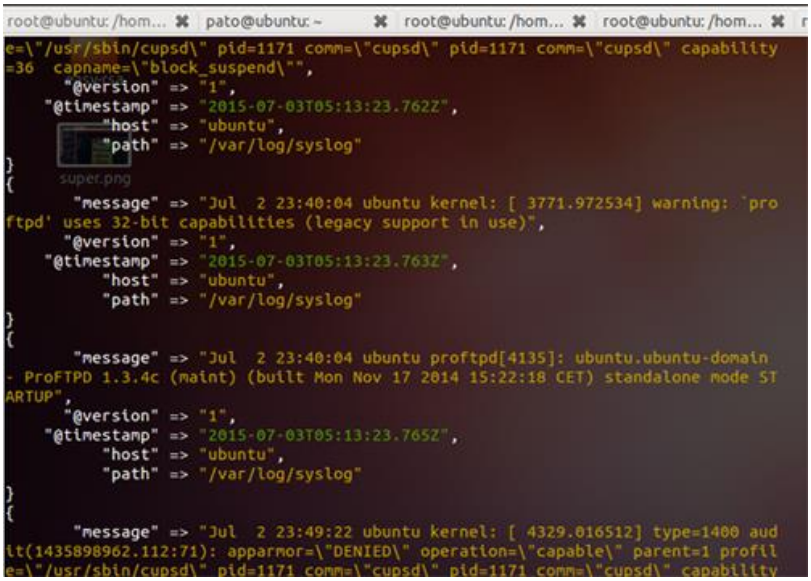

Figura 12. Recolección de los eventos del sistema del archivo syslog.log.

En esta figura se muestra como los archivos ya son formateados de acuerdo a la programación respectiva. Por ejemplo en la figura se aprecia los campos timestamp, message, path y host, entre otros.

Para ejecutar una instancia de Elasticsearch también es necesario escribir los comandos correspondientes. Mientras se inicia, y durante todo el tiempo que la aplicación esté corriendo se puede observar en una ventana de navegador la interface gráfica de Elasticseach.

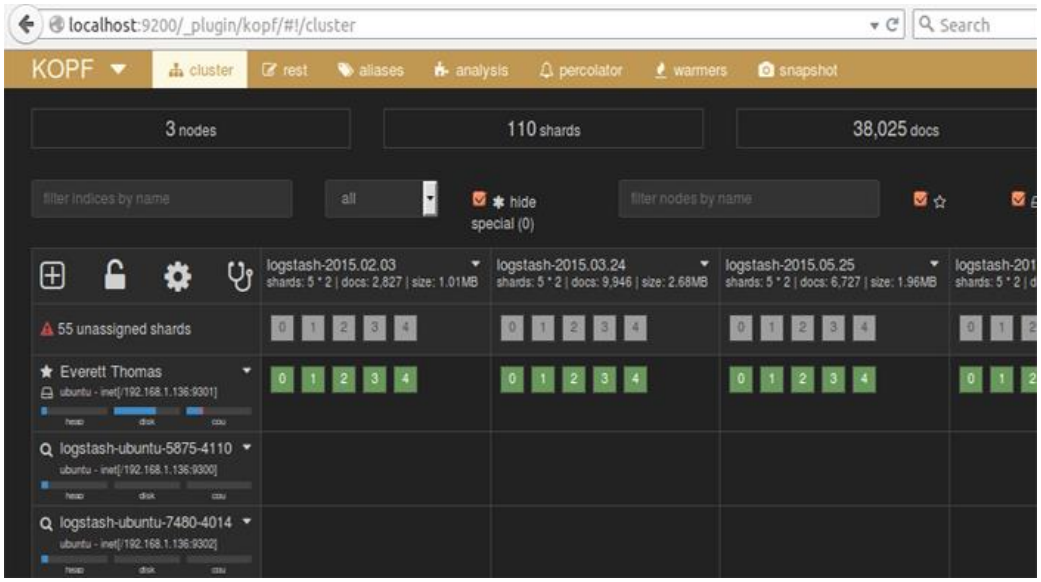

Figura 13. Ambiente desarrollador de Elasticsearch en la ventana de un navegador.

Como se observa en la figura se tiene varias opciones para el indexamiento de los archivos pre-definidos por Logstash. Cada instancia de Logstash es cargada de manera independiente con sus distintos filtros y formato de visualización de texto. Además se tiene diferentes índices que quedan grabados aunque la aplicación deje de correr. Hay que recordar en este punto, que como se está corriendo en una

máquina con servidor local, las aplicaciones no van a estar en funcionamiento todo el tiempo.

Por último, queda por explicar en esta breve introducción la interface de Kibana (el complemento web de Logstash). En una nueva instancia de Logstash hay que dejar ejecutando el comando ./logstash web. Luego en una ventana de cualquier navegador se escribe como URL http://localhost:9292/index.html. A continuación se muestra capturas de pantallas de todo lo que se puede lograr con esta interface web.

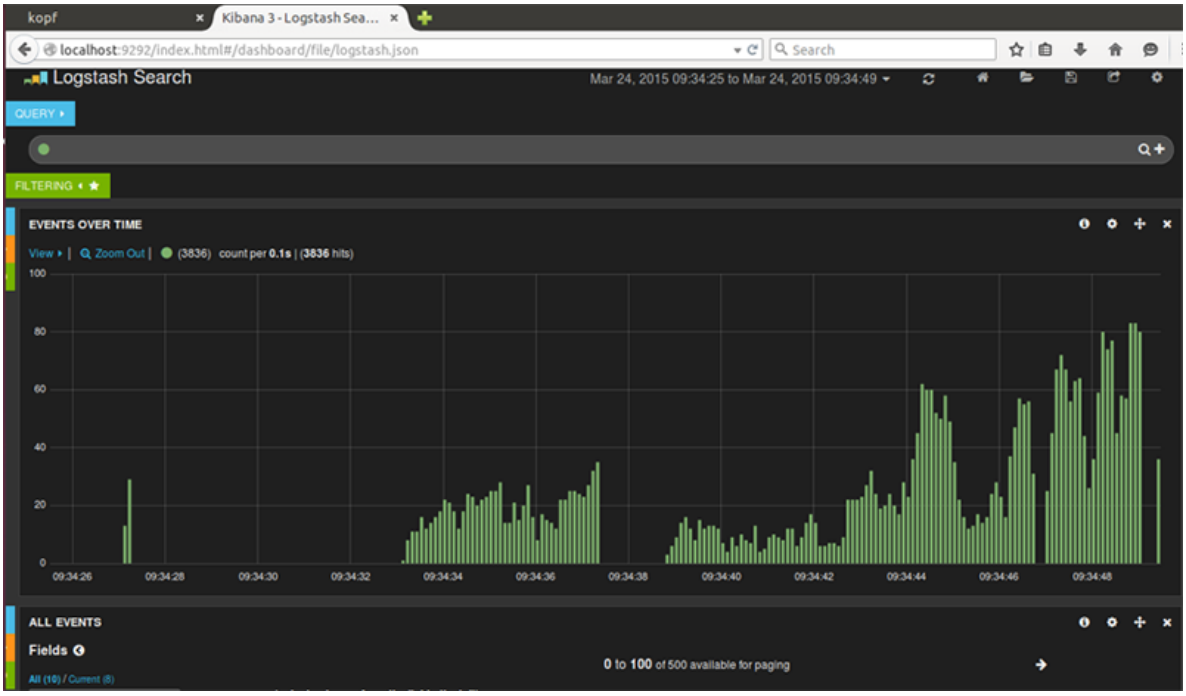

Figura 14. Panel principal de Kibana con registros ya indexados y mostrados gráficamente.

La figura muestra el panel principal de Kibana. Donde se puede ver todos los eventos de los archivos cargados anteriormente. Este dashboard tiene algunas opciones para filtrar eventos de manera gráfica o leer la descripción de algún mensaje en particular.

Además, se puede cambiar las configuraciones de cuantos archivos cargar y mostrar.

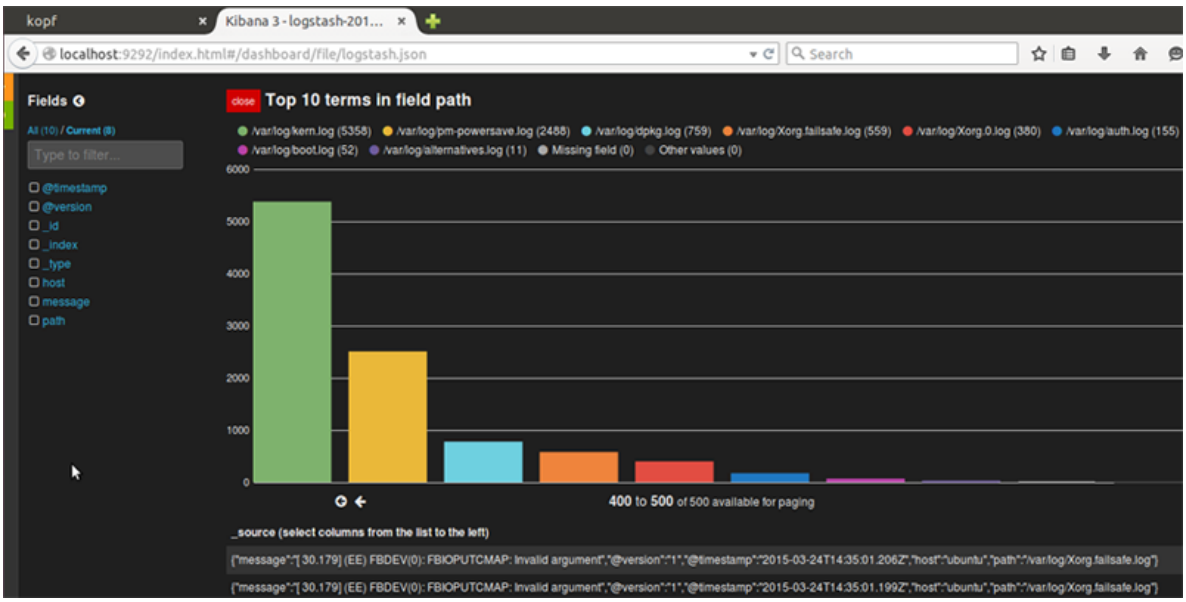

Figura 15. Filtración visual de los registros indexados en Kibana.

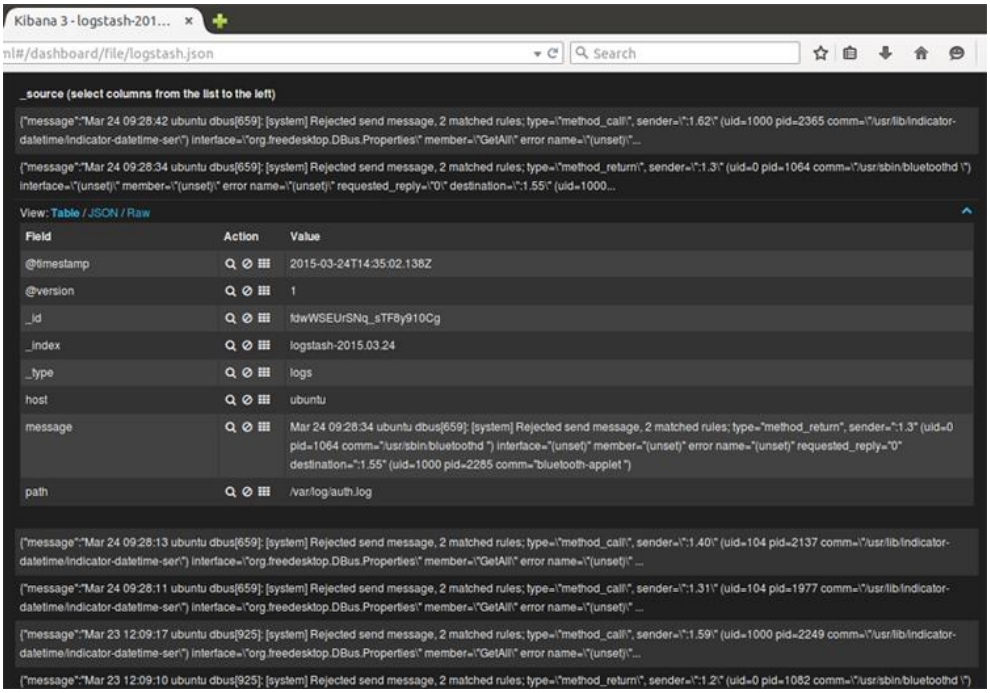

Figura 16. Descripción de mensajes de registros en Kibana.

Las dos imágenes que se muestran representan algunas de las opciones que maneja la interface gráfica de Logstash. Se puede filtrar los eventos de que archivo en específico vienen con gráficos visuales; o también se puede leer la descripción del mensaje y entender como se ha organizado.

Una vez ya mostradas todas las herramientas de software a utilizar, se va a dar la explicación del procedimiento para la detección de ataques en base a lo descrito en las anteriores secciones.

Desarrollo

Introducción.

Los distintos tipos de ataques y como neutralizarlos tienen un seguimiento único que incluye distintos archivos de logs, así como también distintos procedimientos y conclusiones que serán expuestos a continuación. Por diferentes razones que luego serán explicadas en el análisis y las conclusiones del trabajo en esta sección de desarrollo se analizarán y darán los procedimientos de cuatro distintos tipos de ataques: ataque DoS, DDoS, de Defacement y de Fuerza Bruta para romper contraseñas.

Cada detección de ataque incluye figuras y además explicaciones bien detalladas del procedimiento que se siguió. Existen varias ocasiones en donde software adicional de apoyo es utilizado para lograr el objetivo de forzar el ataque y que luego los rastros del ataque sean registrados en Logstash y su posterior visualización con Kibana utilizando el indexamiento de Elasticsearch.

Detección Ataque DoS.

El ataque DoS (Denial-of-Service) se lo realizó teniendo como parámetro de referencia una página web habilitada para mostrar en el servidor local de la máquina. Este ataque como se explicó anteriormente, consiste en que el atacante inunda con

paquetes de datos algún puerto en específico del servidor para no permitir que se muestre dicha página web.

Hay varias maneras de lograr realizar un ataque DoS. Para este caso práctico se utiliza el programa hping3 como se muestra en la figura.

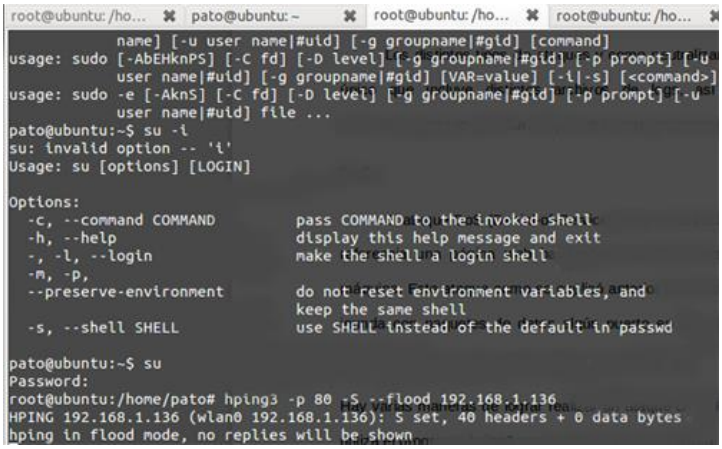

Figura 17. Ejecución software Hping3 para realizar el ataque de DoS.

El 80 representa al puerto que se va a atacar. El puerto 80 generalmente es el puerto por defecto para mostrar todos los archivos que se encuentren ubicados en las carpetas del servidor local. La dirección IP debe ser la de la víctima.

Para comprobar que el ataque está ocurriendo se tiene imagen de la actividad de red de la máquina.

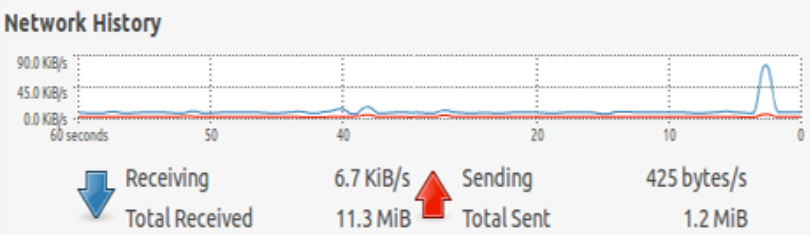

Figura 18. Actividad de red de la máquina víctima del ataque DoS.

La figura muestra claramente el incremento en la cantidad de paquetes que recibe la red local.

Una vez que el ataque interno ha dado resultado, entonces la página cargada en el servidor local debe dar un error de 404 (página no encontrada) cuando se intente actualizar en el navegador.

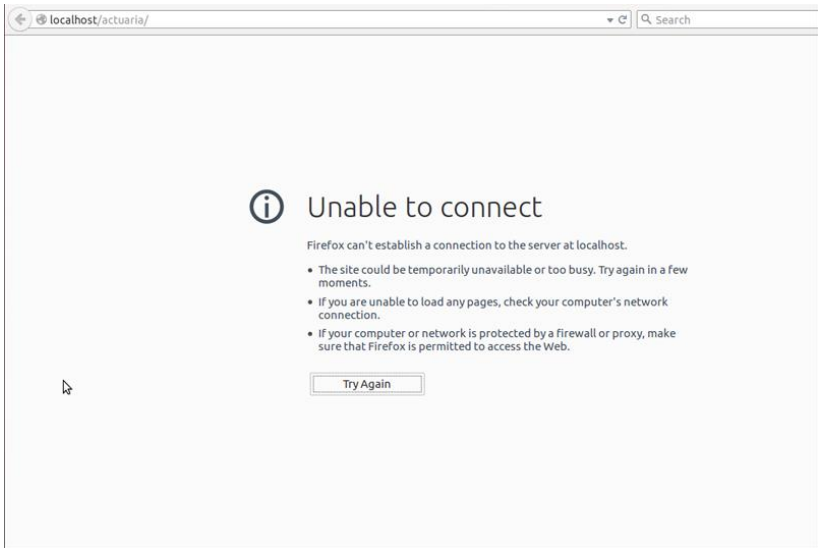

Figura 19. Error 404 por el ataque DoS al servidor local.

En realidad la página debería haberse mostrado como en la siguiente figura:

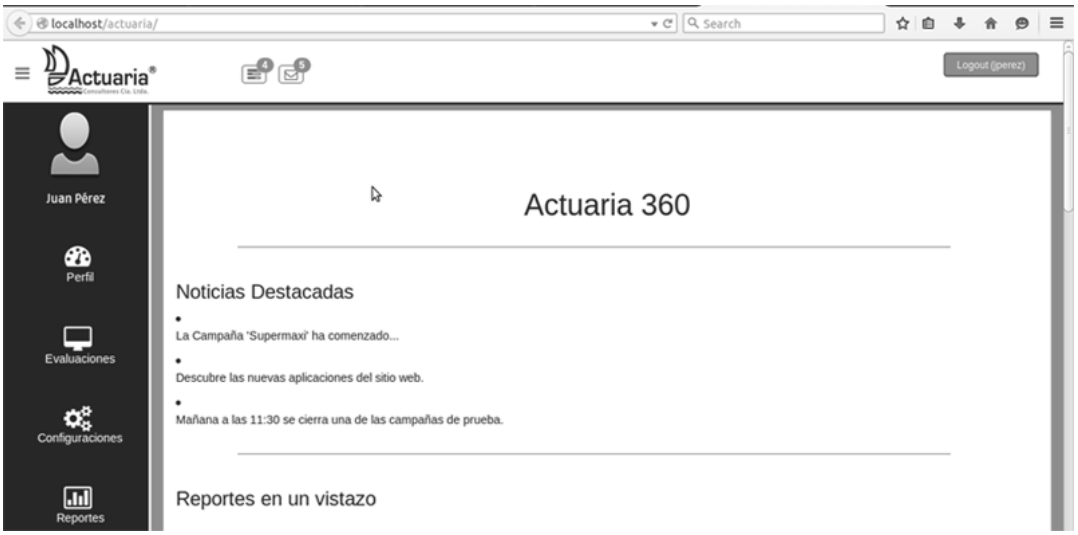

Figura 20. Página actual mostrada sin realizar el ataque de DoS.

Hasta el momento el ataque DoS no ha sido todavía monitoreado en ELK, pero cuando se cargue el archivo de registros correspondiente la interface gráfica nos mostrará los registros.

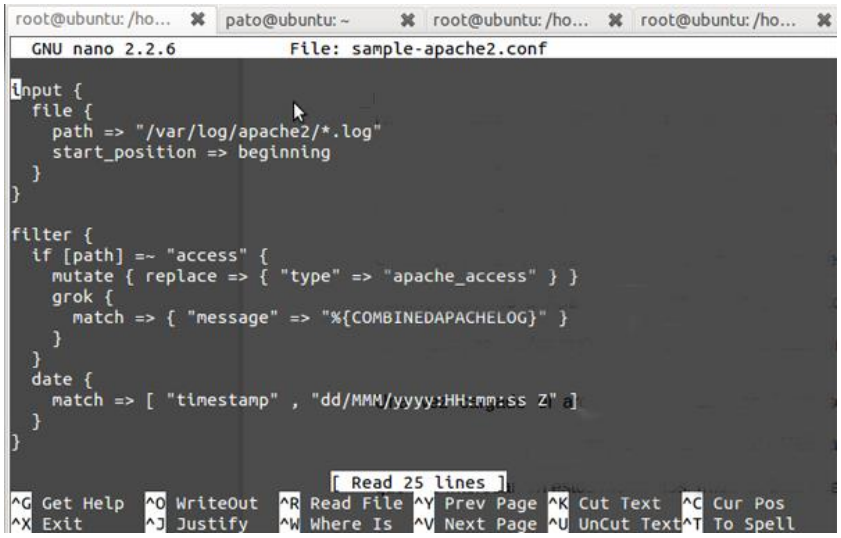

Figura 21. Archivo de configuración de Logstash para registrar eventos de Apache.

Como se puede mostrar en la figura, se debe recolectar los archivos de logs correspondientes a Apache (el servidor local) instalado en la máquina donde deberían aparecer los registros del ataque antes mencionado. Además se filtra por fecha y se busca en los directorios especificados.

Una vez cargado el archivo e indexado en Elasticsearch, se puede realizar distintas queries (peticiones) en la interface de Kibana para filtrar los mensajes que nos interesan en estos momentos. Generalmente los registros de monitoreo de la red se encuentran en el archivo syslog.log. De ahí podemos encontrar los siguientes eventos:

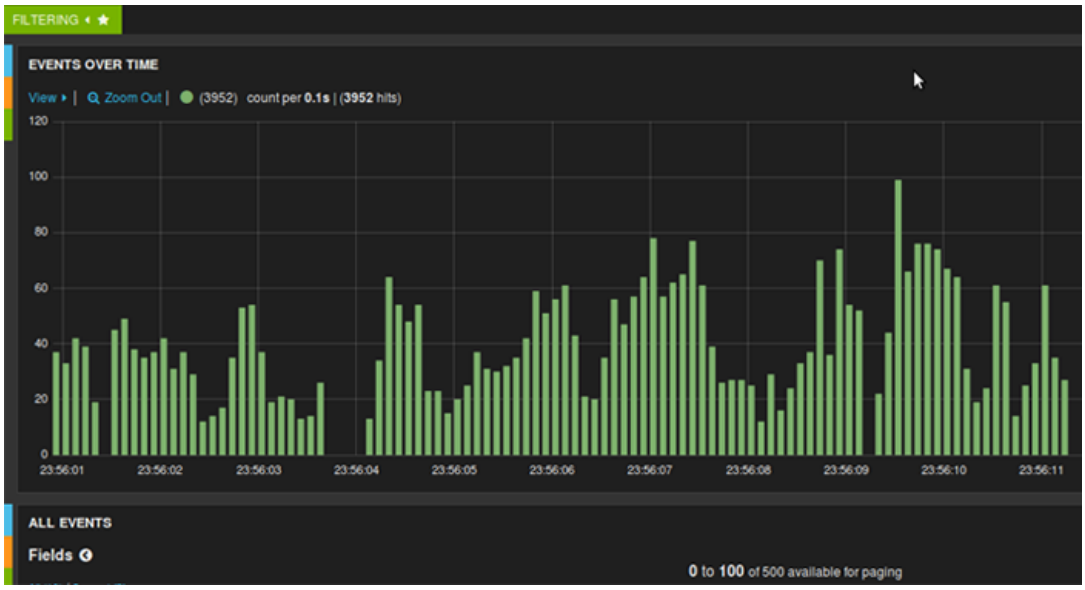

Figura 22. Archivos cargados en Kibana luego de la recolección y el indexamiento.

Filtrando por puerto o por dirección IP encontramos los siguientes eventos relevantes para el caso:

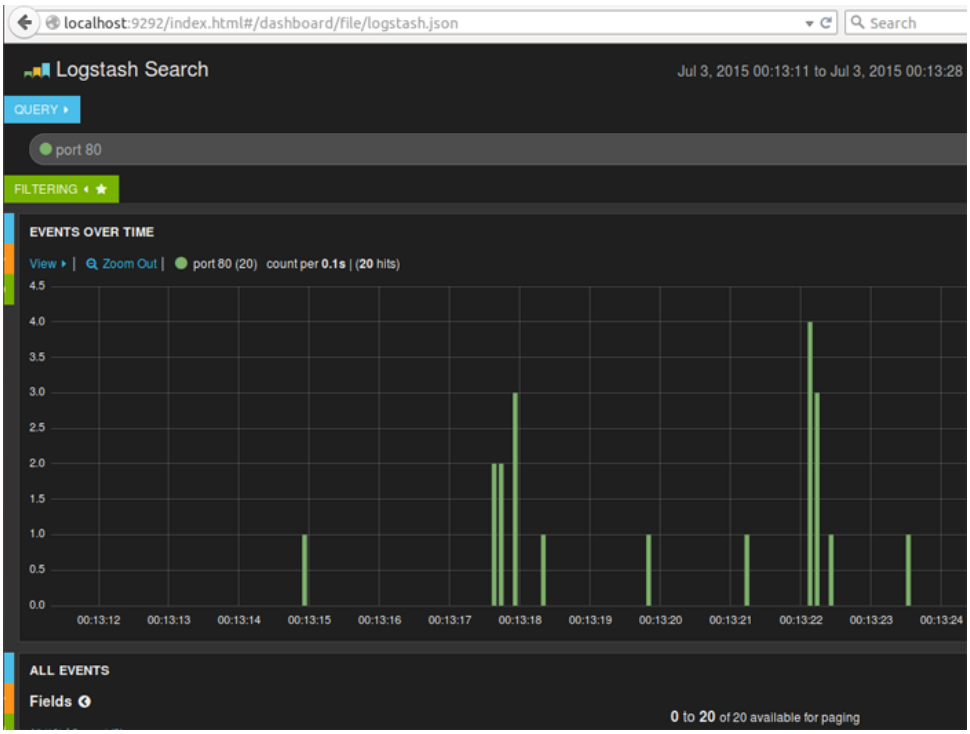

Figura 23. Filtro de eventos en Kibana por la frase: 'Puerto 80'.

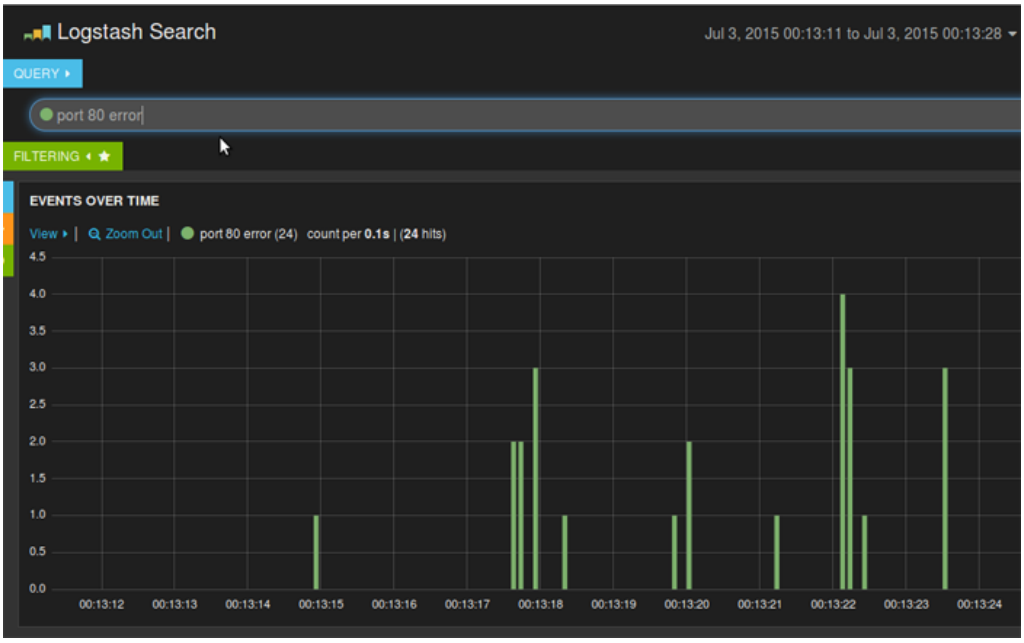

Figura 24. Filtro de eventos en Kibana por la frase 'error puerto 80'.

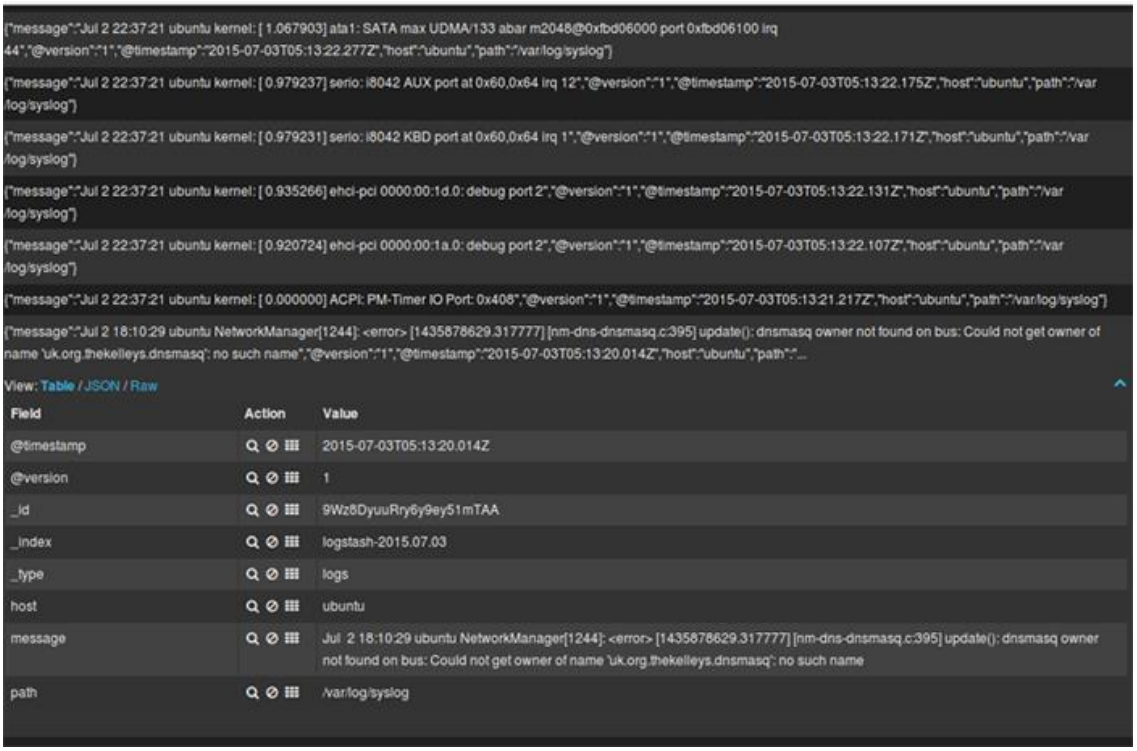

Figura 25. Detalles de los mensajes de error por el ataque de DoS.

Para cualquier ataque del tipo DoS se puede tomar como referencia este proceso y aplicarlo. Como se puede observar en las figuras descriptivas, lo más importante para poder prevenir o contrarrestar este tipo de ataques es saber que están ocurriendo. El registro, a menos de que se cambien las opciones pre-establecidas, siempre guardará toda la actividad de red o posibles ataques a páginas que están alojadas en el servidor local. En realidad, el saber que archivos contienen esos registros y cargarlos a Logstash es lo que no se intuye de manera rápida.

Suponiendo que esto ocurra en el servidor, debería existir un monitoreo constante para evitar este tipo de ataques. Una vez que se ha detectado el ataque DoS entonces se puede dar paso a contrarrestarlo. En este caso se puede bloquear el

puerto por el cual llega el ataque o bloquear la dirección IP de la máquina de dónde se está produciendo el ataque.

Algoritmo de detección.

Tras haber descrito claramente el procedimiento para la detección del ataque de Denial of Service (DoS), se presenta un algoritmo formal de manera escrita y en forma de diagrama.

Algoritmo:

1) Monitorear la actividad de red en busca de anomalías.

2) Si no se encuentran comportamientos extraños
Se continúa con el monitoreo (paso 1)
.
3) Si se llegan a encontrar anomalías en los picos de paquetes recibidos en la red, se prosigue con el paso 4.
Cualquiera que fuera el caso, alimentar Logstash con logs correspondientes de tiempos y eventos.

4) Indexar los logs correspondientes a través de Elasticsearch.

5) Filtrar los eventos en Kibana para encontrar el origen de las anomalías y tener los reportes de manera gráfica
.
6) Filtrar los eventos por:
Puerto 80 (http)
Puerto 8080 (https)

7) Encontrar eventos repetitivos y notorios de peticiones http hacia cualquiera de los dos puertos descritos anteriormente.

8) Encontrar la dirección IP de la máquina que envía las peticiones.

9) Bloquear el IP de la máquina para que las peticiones no sean acogidas por la máquina víctima del ataque.

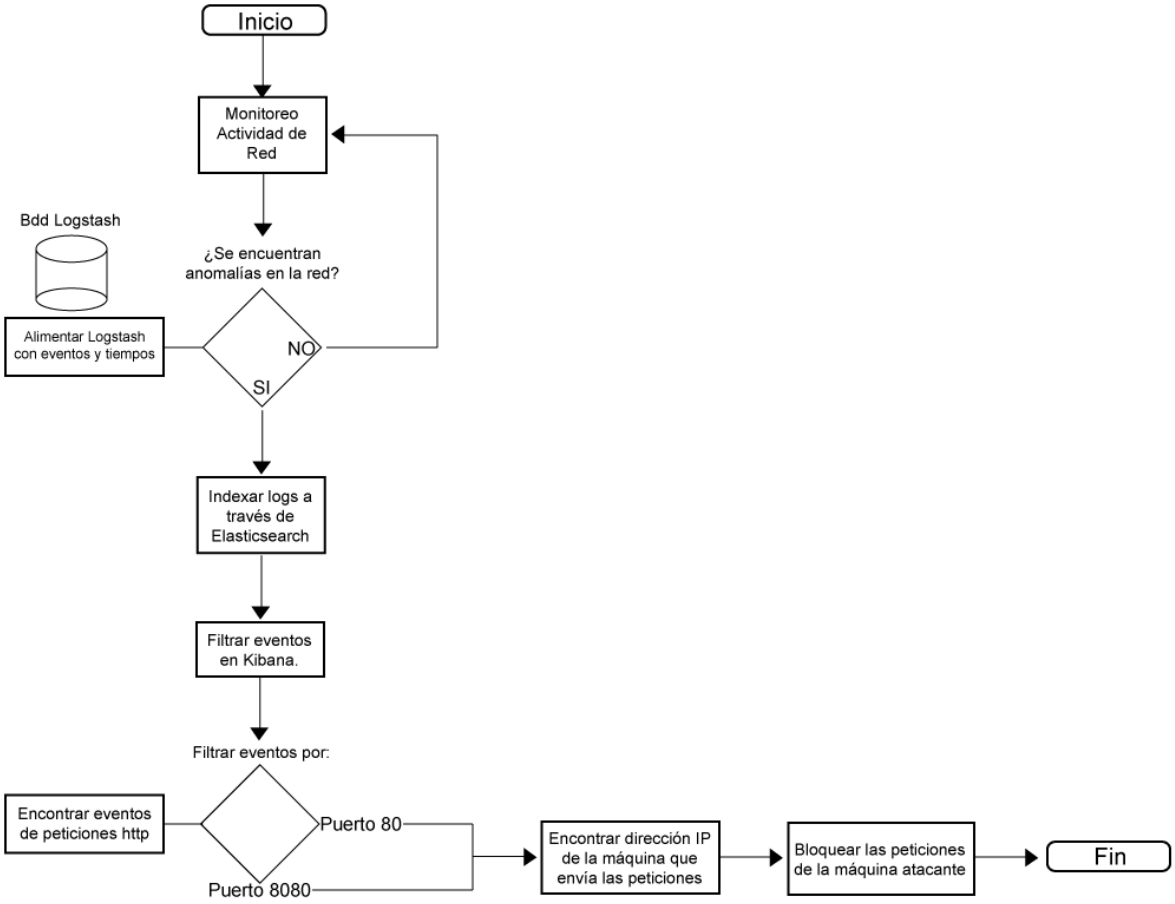

Figura 26. Diagrama de flujo del algoritmo para la detección del ataque DoS.

Detección Ataque de Fuerza Bruta.

Los ataques de fuerza bruta se caracterizan por tratar de romper la seguridad de algún programa en particular mediante la iteración de intentos hasta alcanzar precisión para romper claves. Las vulnerabilidades de sistemas con autenticación de usuario-contraseña generalmente se caracterizan, en el común de personas, de ser fáciles de adivinar pues se eligen palabras o frases muy comunes y probables de que sistemas las descubran por simple estadística.

En este caso para mostrar cómo se puede neutralizar un ataque de fuerza bruta se utiliza el software THC Hydra que permite cargar diccionarios de palabras para realizar pruebas de ingreso a un servicio de forma reiterativa, realizando las combinaciones entre usuarios y contraseñas establecidas.

Figura 27. Archivo de configuración de usuarios para utilizar en el software de Hydra.

Esta primera figura muestra los nombres de usuarios que se eligieron para comenzar a probar la seguridad en Ubuntu.

Figura 28. Archivo de configuración de contraseñas para utilizar en el software Hydra.

Esta otra figura muestra los passwords elegidos para intentar romper la clave del usuario.

El ataque se va a realizar contra las credenciales del usuario 'sudo' el de mayores privilegios en una máquina con sistema operativo Ubuntu.

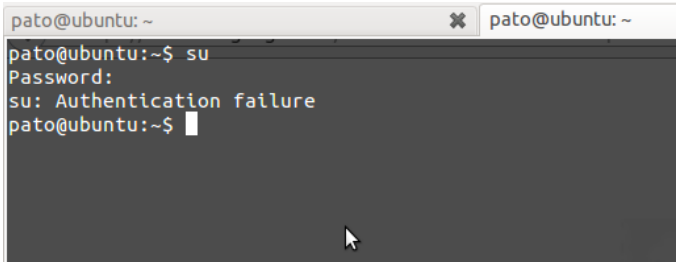

Figura 29. Autenticación de usuario ‘sudo’ en un sistema operativo Ubuntu.

Como se aprecia en la figura, el usuario administrador 'sudo' tiene autenticación

Para el respectivo monitoreo utilizando el sistema SIEM ELK lo principal es cargar el archivo de logs del kernel y de los registros del sistema.

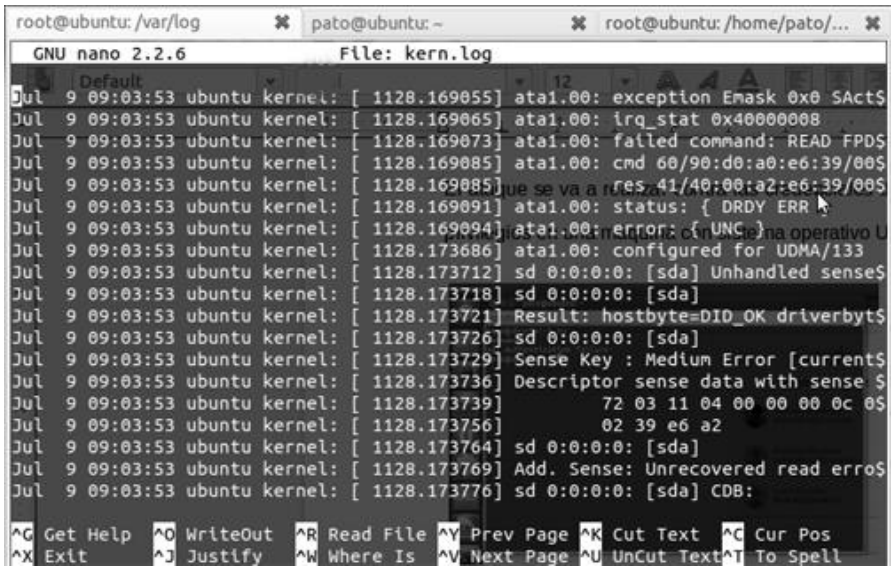

Figura 30. Archivos de registros del kernel. Kern.log.

Una vez el archivo ya cargado y listo para el indexamiento se puede mostrar en la interface gráfica de Kibana los intentos por romper la autenticación del súper usuario en Ubuntu.

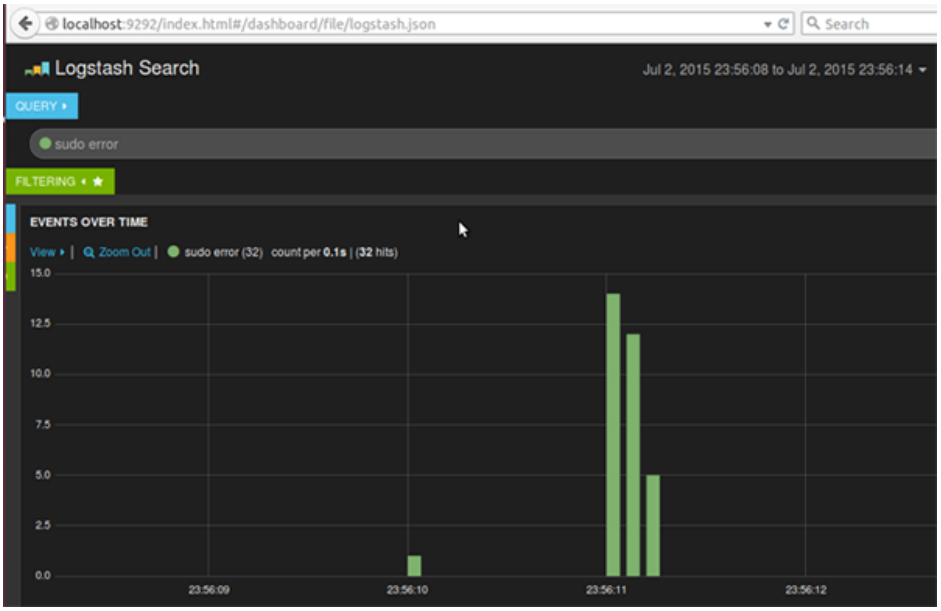

Figura 31. Filtro de eventos en Kibana durante el ataque de fuerza bruta.

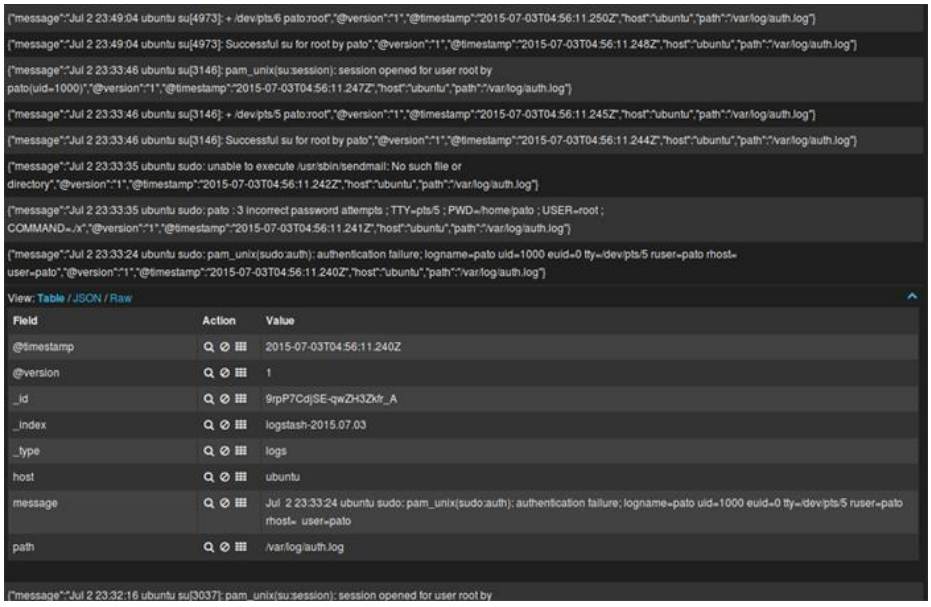

Figura 32. Detalle de los mensajes de error en la autenticación del usuario ‘sudo’.

En las figuras que se muestran se puede observar como Kibana puede filtrar los eventos de intentos por evadir la seguridad en Ubuntu. Hay varios eventos registrados que corresponden al ataque que se está llevando a cabo por el software Hydra.

Se puede filtrar no solo los intentos que no tuvieron éxito para romper la autenticación con fuerza bruta, sino también si algún intento tuvo éxito. Además se puede saber la fecha y hora exacta de cuando se lo hizo.

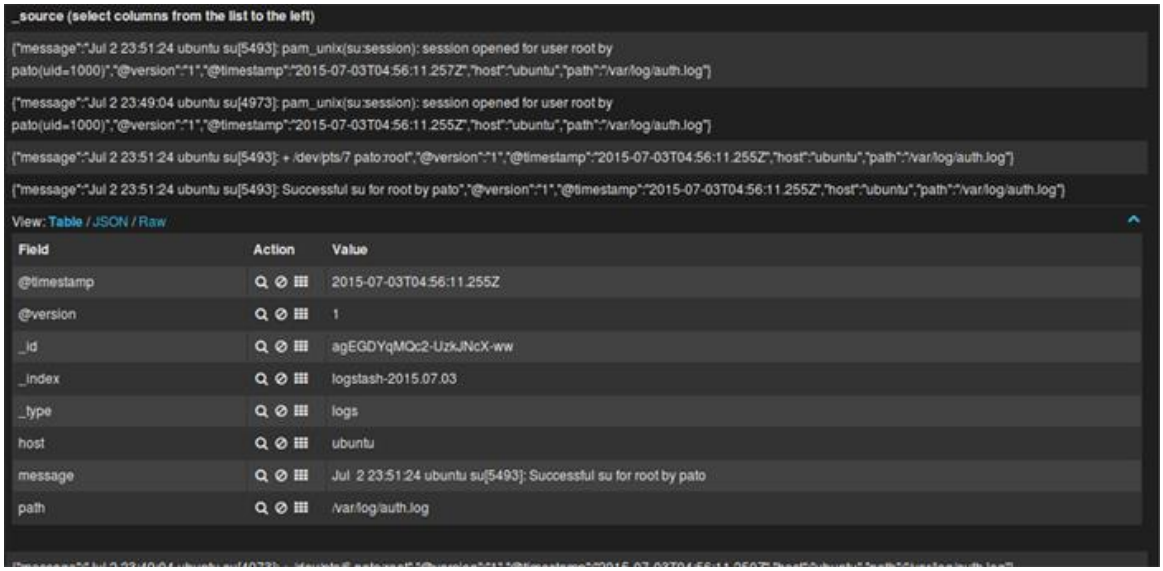

Figura 33. Exitosa autenticación del usuario ‘sudo’ en Ubuntu.

Esto es muy importante para sistemas en dónde se requiere saber que usuario tuvo permisos predominantes sobre los diferentes archivos del sistema. Si el monitoreo tiene éxito entonces se puede prevenir dichos ataques con una buena política de modificación de las credenciales del usuario por frases en otro idioma o sin mucho sentido en el idioma nativo del usuario que es el encargado de establecer el nombre de usuario y la contraseña. Esta recomendación es solo una de tantas buenas

prácticas para incentivar el uso de fuertes contraseñas mucho más complicadas de romper.

Algoritmo de detección.

Tras haber descrito claramente la técnica para la detección del ataque de fuerza bruta, se presenta un algoritmo formal de manera escrita y en forma de diagrama.

Algoritmo:

1) Buscar en la máquina (terminal o servidor) el archivo que almacena los registros de eventos del sistema. Ej: kern.log.

2) Cargar en Logstash el o los archivos correspondientes.
.
3) Realizar el idexamiento de los eventos con Elasticsearch y cargar los eventos en la interface gráfica Kibana.

4) Filtrar los eventos monitoreados en tiempo real por palabras claves: ‘advertencia’, ‘error’, ‘autenticación’. Queda al juicio del usuario.

5) Si no se encuentran eventos atípicos en grandes cantidades
    volver al paso2.

6) Si se encuentran eventos atípicos en grandes cantidades.

7) Revisar si algún intento de romper la autenticación tuvo éxito.

8) Cambiar las credenciales que se crea están siendo vulneradas.

9) Promover el uso de contraseñas difíciles de romper.
    Se pueden utilizar palabras en otro idioma o palabras sin sentido.

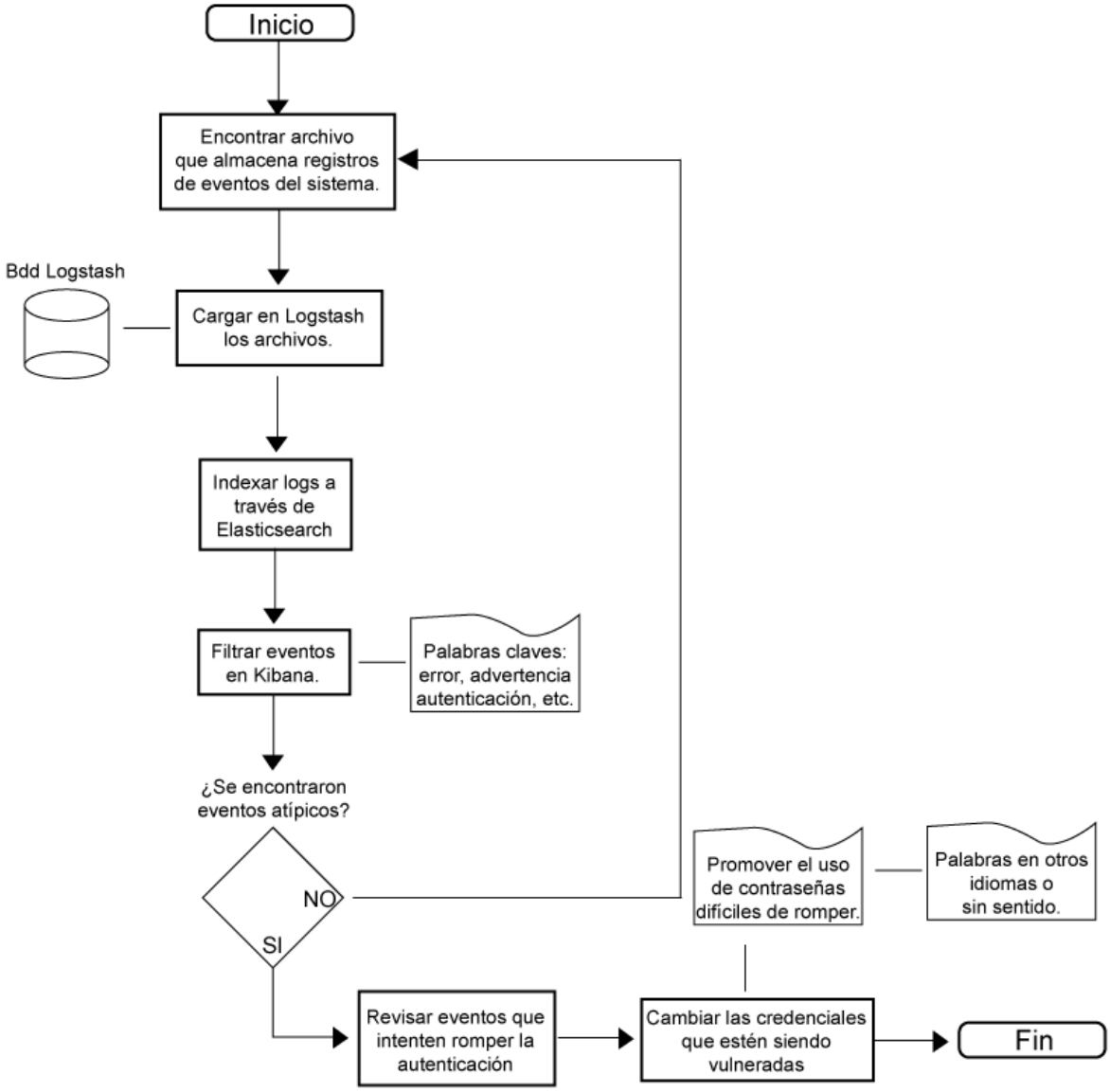

Figura 34. Diagrama de flujo de la detección del ataque por fuerza bruta.

Detección Ataque de Defacement.

El ataque de defacement se refiere al tipo de ataques en dónde un usuario sin autorización y de manera anónima modifica archivos cargados en algún hosting y

cambia cierto código para mostrar algo diferente a la versión oficial en el back-end de una página web. Para hacer la simulación de este ataque se elige una página cualquiera alojada en el servidor local de la máquina y se le saca una copia idéntica en algún otro directorio también del mismo servidor local.

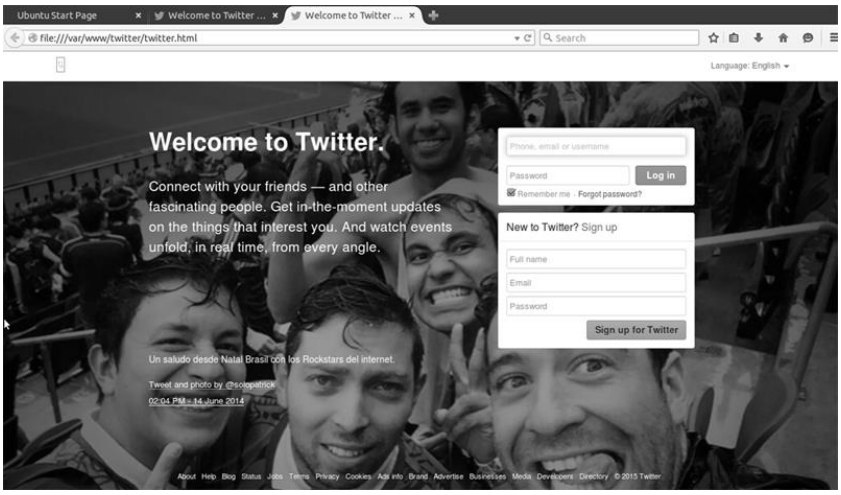

Figura 35. Página de inicio de un sitio web al cual se le someterá al ataque de defacement.

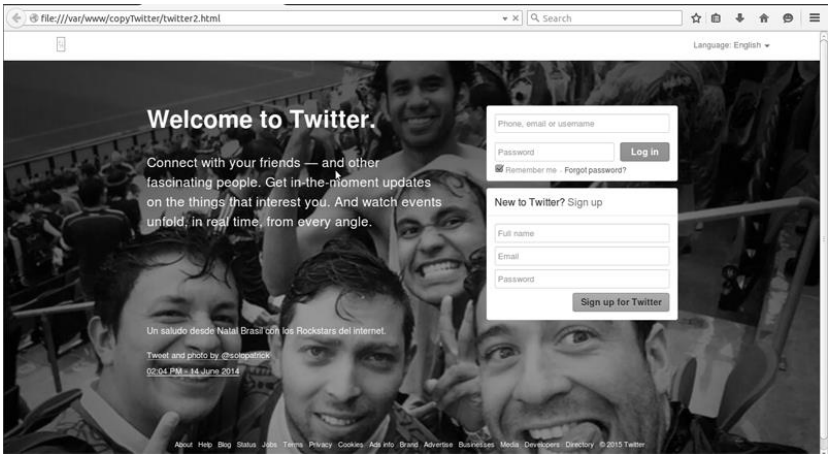

Figura 36. Página de inicio del sitio de respaldo para evaluar el ataque de defacement.

Para este ejemplo se tomó como referencia el sitio web de twitter.com. Se descargó los archivos y se hizo una copia exacta en el servidor local. De esta ya copia de

Twitter original se sacó una segunda copia de este mismo directorio con todos sus archivos. Así se muestra en las fotos:

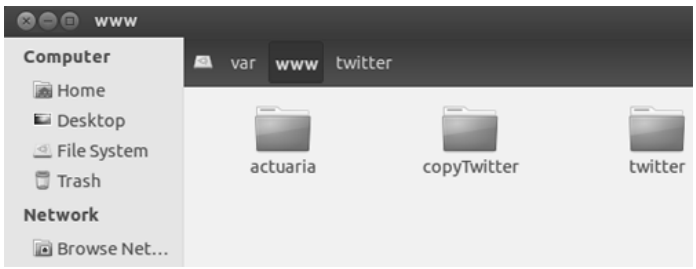

Figura 37. Directorio de los sitios web alojados en el servidor local.

La copia de seguridad se hace con la intención de poder monitorear que siempre el original tenga el mismo código que la copia, caso contrario entonces el administrador del sitio web sabrá que alguna línea de código cambió y entonces puede bajar la página del servidor hasta encontrar el error.

Este tipo de ataque es muy común cuando atacantes intentan mostrar algo en diferentes sitios web. Teniendo ya la copia de seguridad se pasa a crear un script que esté monitoreando cada cierto periodo de tiempo entre las dos páginas.

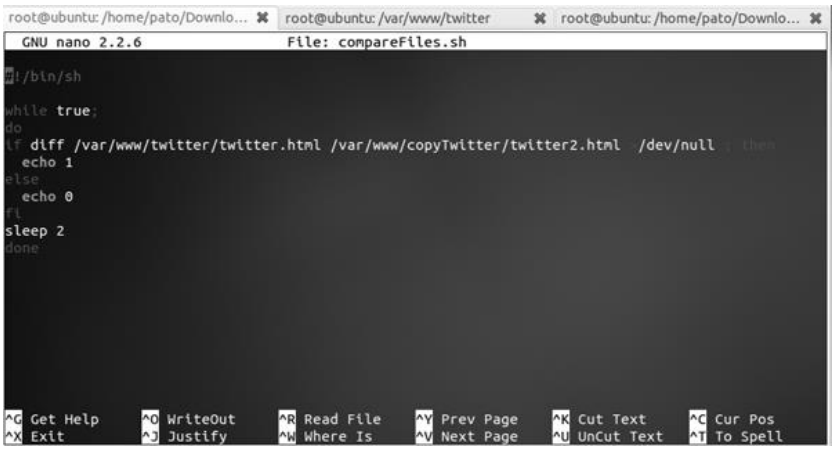

Figura 38. Script utilizado para la comparación de dos archivos idénticos.

En este caso se elige que se escriba un 1 o 0 dependiendo si las páginas son iguales o no. El script escribe estos resultados reiterativos en un archivo .log para luego poder cargar en Logstash. Se eligió que el script se esté ejecutando cada dos segundos. Si no se pondría un tiempo de espera, con la velocidad de un computador el archivo log luego de unos cuantos segundos ocuparía demasiado espacio en disco. El comando 'diff' compara los archivos.

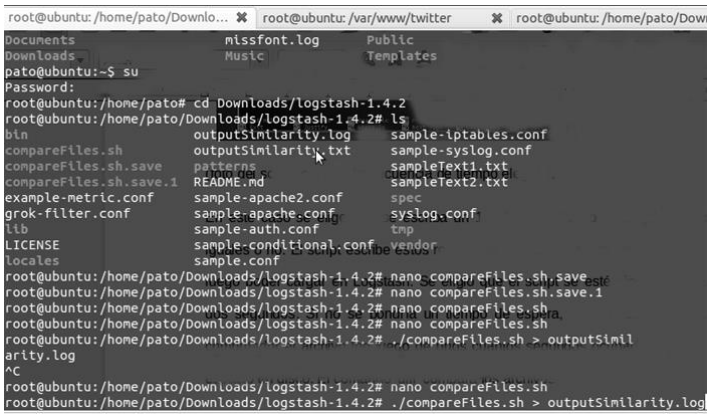

Figura 39. Sentencia que ejecuta el script en el servidor local.

Al ejecutar el script (luego de haberle dado permisos para que pueda escribir en otros archivos) se le dice que todo el output que escriba lo ponga en el archivo 'outputSimilarity.log'.

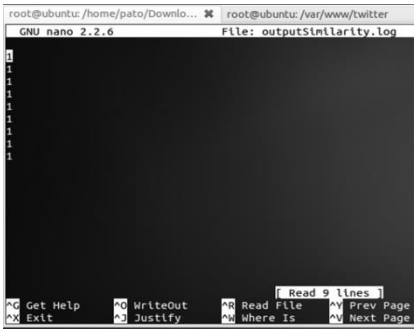

Figura 40. Archivo outputSimilarity.log que almacena la comparación de los archivos.

El resultado de la figura es la comparación de los dos archivos. El 1 significa que la comparación es exacta y por lo tanto no ha habido ningún cambio en el código de ninguno de los dos archivos.

El archivo de configuración de Logstash es el siguiente:

Figura 41. Archivo de configuración de Logstash para identificar el ataque de defacement.

Una vez que el sistema SIEM ya tiene los registros necesarios se puede comenzar a filtrar el archivo para encontrar si la página fue modificada o no.

Por el momento la imagen nos muestra que todo funciona de acuerdo a lo planeado. Entonces, se planea hace un cambio en el archivo original que va a ser detectado por el script que se creó y que se va a ver reflejado en el indexamiento de los logs. Al ser detectado el cambio del archivo original pues el script comenzaría a escribir 0 en el archivo de comparación, el administrador de la red o del sitio web debe bajar de inmediato el archivo que cambió por terceros (atacantes). Lo importante de detectar este tipo de ataques es que se interfiere rápidamente en el propósito del atacante: mostrar modificada alguna página (generalmente la página de inicio) en el sitio web.

Si el administrador del sitio quita rápidamente esa página mostrada en vivo, el ataque queda neutralizado.

Algoritmo de detección.

Tras haber descrito claramente el procedimiento para la detección del ataque de defacement, se presenta un algoritmo formal de manera escrita y en forma de diagrama.

Algoritmo:

1) Crear copias de todos los archivos html a proteger.

2) Crear un script que compare la copia y el archivo original línea por línea en un intervalo de tiempo de 60 segundos.

3) Cada instante que el script se esté ejecutando, alimentar logs con eventos y tiempos.

4) Si el archivo original no ha cambiado,
Continuar con la comparación mientras el archivo original se encuentra en el servidor y accesible desde Internet.

5) Si el archivo original si ha cambiado,
Notificar mediante un correo electrónico al administrador del sitio web.

Este debería:
Tener acceso y hacer al recurso modificado no accesible hasta verificar los cambios.

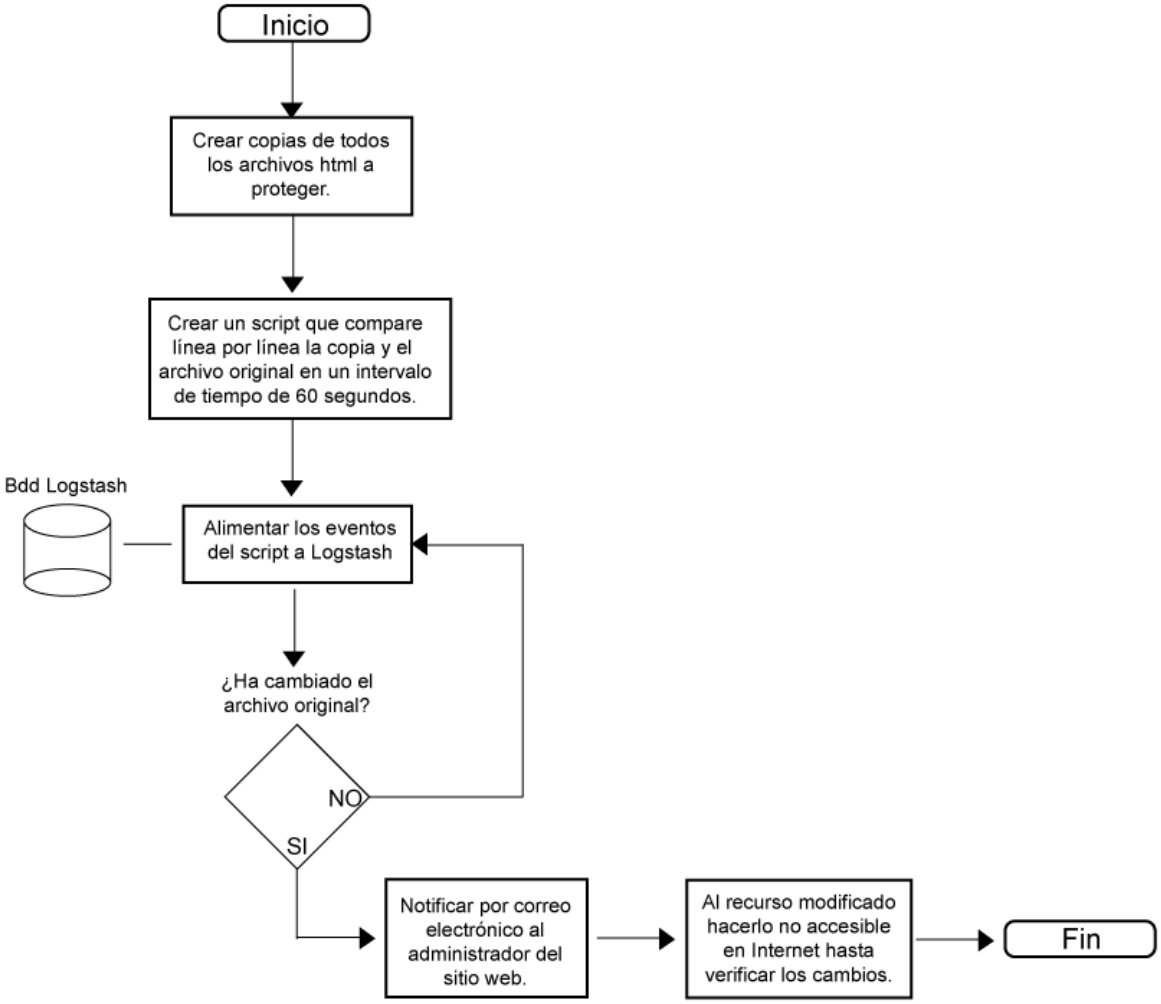

Figura 42. Diagrama de flujo del algoritmo de detección del ataque de Defacement.

Detección Ataque DDoS.

Existe mucha confusión en los temas referentes a la seguridad informática en cuánto se refiere a los ataques de DoS y de DDoS. Aunque en concepto son el mismo tipo de ataque, se diferencian pues el ataque DDoS es distribuido, de allí la primera letra de su acrónimo. Esto significa que no se realiza desde una sola máquina el ataque sino de varias y lo que intentan es quitar la disponibilidad de la

página web para los usuarios que requieran ese recurso. Si es que el recurso se vuelve temporalmente no disponible para usuarios comunes entonces el ataque es considerado exitoso.

Para realizar un ataque de DDoS se utilizaron dos máquinas además de la que se encuentra trabajando como servidor.

En casos prácticos este ataque de DDoS es el más común para ataques de 'Denial of Service' debido a que en ataques reales para lograr la inhabilitación de recursos en un servidor no basta con realizar el ataque desde una o un par de máquinas, sino desde varias máquinas ya programadas para difundir el ataque en sincronización. A continuación se muestran las imágenes de dichos ataques:

Figura 43. Ataque de DDoS desde una computadora con sistema operativo Windows.

Computador que colabora con el ataque desde un sistema operativo Windows. Como se aprecia en la figura, se ataca la IP 192.168.1.136 desde la IP 192.168.1.124 que se encuentra en la misma red de área local (LAN).

Se puede también monitorear la red y se encuentra como los paquetes de descarga por el ataque exceden y por mucho a los paquetes de 'upload' (subida).

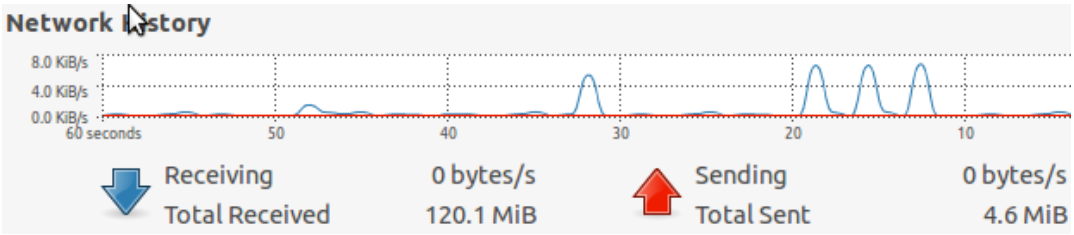

Figura 44. Monitoreo de red mientras se desarrolla un ataque DDoS en otro computador.

Los picos en esta figura en el gráfico de 'Network History' muestran el incremento en la actividad.

Mientras el ataque se está llevando acabo, se realiza el mismo procedimiento que para un ataque DoS. Se tiene que cargar el archivo de configuración de Logstash, habilitar la interface web (Kibana) e indexar los logs con Elasticsearch.

Algoritmo de detección.

Tras haber descrito claramente el procedimiento para la detección del ataque de Denial of Service (DoS), se presenta un algoritmo formal de manera escrita y en forma de diagrama.

Algoritmo:

1) Monitorear la actividad de red en busca de anomalías.

2) Si no se encuentran comportamientos extraños
Se continúa con el monitoreo (paso 1)
.
3) Si se llegan a encontrar anomalías en los picos de paquetes recibidos en la red, se prosigue con el paso 4.
Cualquiera que fuera el caso, alimentar Logstash con logs correspondientes de tiempos y eventos.

4) Indexar los logs correspondientes a través de Elasticsearch.

5) Filtrar los eventos en Kibana para encontrar el origen de las anomalías y tener los reportes de manera gráfica
.
6) Filtrar los eventos por:
Puerto 80 (http)
Puerto 8080 (https)

7) Encontrar eventos repetitivos y notorios de peticiones http hacia cualquiera de los dos puertos descritos anteriormente.

8) Encontrar las direcciones IP de las máquinas que envían las peticiones.

9) Bloquear las IPs necesarias para que las peticiones no sean acogidas por la máquina víctima del ataque.

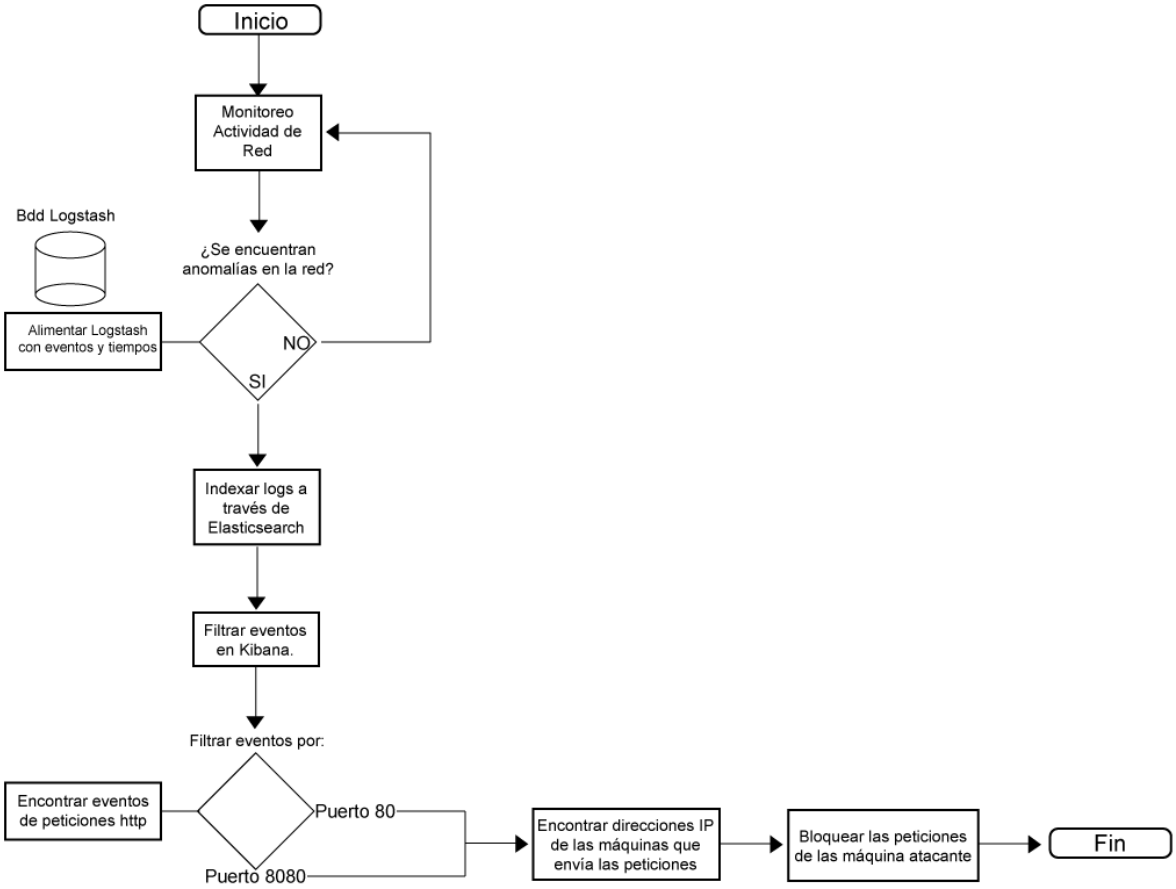

Figura 45. Diagrama de flujo del algoritmo de detección del ataque DDoS.

Síntesis

Queda en evidencia en esta sección de desarrollo del trabajo los procedimientos a seguir para la detección de algunos ataques de red a través del sistema SIEM ELK (Elasticseach – Logstash – Kibana). Es importante resaltar que este procedimiento puede variar dependiendo el sistema SIEM con el que se maneje. Aunque en esencia muchos aspectos de los procedimientos descritos serán muy parecidos.

La mayoría de las herramientas que se pueden utilizar para la detección de ataques son open-source (de libre uso) y esto es una ventaja para invertir en el monitoreo de redes en las distintas industrias que existen. Con el acceso tan penetrante a tecnología nueva en la actualidad, es importante armar estrategias de contingencia para estar preparados para cualquier ataque a las redes. La interconectividad actual en el mundo permite que el negocio de los atacantes a las redes sea muy cotizado en el mercado.

En las siguientes secciones de este trabajo se analiza a mayor profundidad los procedimientos para la detección de los ataques a la red; así como también las posibles recomendaciones para futuros trabajos que estén orientados a investigar diferentes procedimientos para la detección de los ataques mencionados, o ataques distintos. Esto, debido a que la gama de ataques a las redes es muy amplia y hay varios ataques que en este trabajo no se llegan a cubrir.

CAPÍTULO 5.- ANÁLISIS Y CONCLUSIONES

Análisis y Conclusiones

Al comienzo de la presente investigación la finalidad principal era detectar ataques informáticos en los sistemas SIEM. Existen muchas herramientas para la detección de estos ataques, aunque muchas veces los procedimientos pueden variar dependiendo el software que se esté utilizando. Para este caso particular se utilizó el sistema ELK que es una combinación de dos softwares más una interface gráfica para un análisis mucho más completo y visual de los registros de un computador o servidor.

A continuación se presentan algunas resoluciones en base a lo presentado en el desarrollo del trabajo con respecto a los procedimientos para la detección de los ataques a la red. Además se evalúa el análisis descrito de dichos procedimientos y que tan eficientes pueden llegar a ser si se los aplica en distintos contextos informáticos.

La importancia de los procedimientos.

Es muy relevante para el análisis de este trabajo el analizar la descripción de los procedimientos que se propusieron en la sección del desarrollo. Aunque como

todo trabajo de titulación hay mucha investigación, este trabajo en particular también tiene su parte práctica que puede ser sin muchos inconvenientes de compatibilidad aplicado por diferentes lectores.

La definición formal de un procedimiento es la siguiente: “Un **procedimiento** es un conjunto de acciones u operaciones que tienen que realizarse de la misma forma, para obtener siempre el mismo resultado bajo las mismas circunstancias”. El conjunto de acciones en este caso que hay que evaluar es las formas posibles para alcanzar un mismo resultado. Cada uno de los cuatro ataques mencionados en el desarrollo del trabajo tienen su procedimiento distinto y único para ser detectados por los sistemas de seguridad: SIEM. Un procedimiento debe apoyarse en distinto material didáctico como lo son: textos e imágenes en este caso. Las imágenes apoyan lo descrito y muestran de una manera visual como se logra el resultado de la detección de los ataques.

Es por esto, que se recalca la importancia de la descripción de los procedimientos pensando en futuros trabajos y artículos que puedan hacer mención a como, de una manera práctica se pueden detectar ataques a las redes informáticas.`

Dificultades para detectar ataques.

En el capítulo de introducción del presente trabajo se habló de algunos ataques que en el desarrollo no se describe el procedimiento. La razón por la que no se tomaron en cuenta todos los ataques fue porque para muchos de esos no es

necesario tener solo el software de un sistema SIEM para lograr detectarlos, sino una combinación de distintos componentes de computación.

Por ejemplo, para detectar un ataque de phishing muchas veces la víctima en el momento en que se dispone a abrir el enlace falso que el atacante le ha enviado puede darse cuenta que el ataque es inminente. No sería necesario todo un sistema SIEM que se encuentre monitoreando la sesión en Internet de la víctima. Además no solo sería necesario cualquier sistema de seguridad sino que además se debería tener conexión con las bases de datos actualizadas para que esos sitios fraudulentos sean detectados de manera oportuna. Una analogía sería la comparada con cómo funcionan los antivirus en general. Un antivirus puede estar preparado para detectar software malicioso únicamente si dicha muestra del software malicioso se encuentra registrado en la base de datos. Virus que son fabricados y salen al mercado como nuevos no tienen forma de ser detectados pues todavía nadie los ha puesto en una base de datos y ha compartido dicha información con los distintos clientes antivirus.

Para el ataque de Man-in-the-Middle (hombre en el medio) en cambio se tendría que realizar el proceso de tener adaptado un sistema de seguridad en un servidor. Para este caso no sería aplicable, pues los ataques detectados y descritos son simulados en una máquina que no es servidor, sino que utiliza un software (Apache) para funcionar como servidor local. Además, para este tipo de ataques software llamados 'middleware' son especializados para encontrar intrusos en la red. Un ejemplo es el programa Wireshark.

ELK como el sistema SIEM utilizado.

¿Por qué se eligió Logstash como sistema SIEM? La respuesta tiene dos partes. Antes de la existencia de Logstash ya existían otros sistemas de seguridad para redes como Splunk, Syslog-ng, Chukwa, Scribe, entre otros. La ventaja principal de Logstash (ELK) es que es un software open-source, no muy pesado y de fácil integración. Además el complemento perfecto de Logstash, pero software independiente es Elasticseach. Este software en cambio se encarga del indexamiento de esos logs ya registrados siendo un software RESTful. Por último, queda resaltar la herramienta gráfica de Logstash: Kibana. Permite generar gráficos de todo tipo en base a filtros personalizados y peticiones a los indexamientos de los logs. Además es un front-end mucho más amigable a las anteriores interfaces gráficas que se utilizaba en Logstash.

Por un lado el software sin costo para desarrollo. Por el otro herramientas independientes pero compatibles para dar un funcionamiento más volátil y veloz a los usuarios.

Software Open-Source no implica gratuidad.

El software **Open-Source** es un tipo de software que puede ser cambiado, modificado y compartido de manera libre. Este tipo de software es desarrollado por una comunidad y distribuido bajo licencias de la definición de Open-Source. Se hace referencia a lo que significa este tipo de software pues el sistema SIEM de seguridad

ELK es open source (de código abierto). Existe mucha confusión en la actualidad acerca de lo que significa que un programa de software sea de código abierto. Muchas veces se asume que el significado se refiere a que el costo del desarrollo fue $0, cuando esa asunción está muy alejada de la realidad. Aunque no está especificado en los objetivos primarios o secundarios de este trabajo, como autor intenté realizar el análisis de la detección de ataques con un sistema de código abierto para crear conciencia de que aunque la distribución del sistema no tenga un costo para los usuarios, eso no necesariamente implica que el costo del desarrollo fue nulo. La herramienta puede ser 'gratis' pero el trabajo empeñado por los distintos co-autores del software tiene un costo sin duda. Incluso para este trabajo el sistema operativo de la máquina que actuó como usuario de ELK es de código abierto. Una distribución de Linux: Ubuntu versión 12.04.

En el medio ecuatoriano es a veces difícil hacer comprender a empresarios que los sistemas que se basan en programas o software open source, no implica que el costo del desarrollo necesariamente será menos costoso. Es una generalización que continuamente le hace mal a la industria pues el trabajo de muchos desarrolladores se ve menospreciado y no permite el desarrollo de sistemas más eficientes y bien construidos.

De procedimientos construir algoritmos.

Los procedimientos que se definen en la sección de desarrollo deben ser también evaluados en base a qué tan sencillo es transformarlos a algoritmos. La

diferencia entre procedimiento y algoritmo básicamente es el lenguaje que se utiliza para la definición de ambos. Mientras que un procedimiento puede estar escrito en un lenguaje común, los algoritmos se caracterizan por su lenguaje más formal y científico de acuerdo a la rama a la que se refieren. De ahí que el objetivo común de ambos es el cumplir con algún objetivo en particular.

Si nos referimos a los procedimientos descritos en la sección de desarrollo, de los cuatro ataques a la red simulados que son: 1) denegación de servicios (DoS), 2) denegación de servicios distribuidos (DDoS), 3) ataques por modificación de páginas (defacement) y 4) Ataque de diccionario (brute-force dictionary attack); se puede apreciar claramente procedimientos bien escritos y descritos con ayuda de imágenes para lograr el objetivo que es la detección de los ataques simulados.

En este trabajo se logra tener estos procedimientos para en un futuro poder escribirlos como algoritmos de sistemas SIEM más especializados y específicos para el propósito que se tenga. Por ejemplo, en el desarrollo de este trabajo los ataques son descritos de manera más generalizada pero específicos en cuanto a los sistemas que se utilizó. Para este tipo de sistema los procedimientos bien pueden ser útiles para alcanzar algoritmos en lenguaje más formal y específico en temas de seguridad e informática.

Concordancia con los objetivos.

Revisando los objetivos descritos en secciones anteriores vale la pena hacer un balance general de cuáles se han cumplido y cuáles no. Uno de los objetivos era el de entender cómo funciona un sistema SIEM de información de sistema y manejo de eventos por sus siglas en inglés. Ya en la introducción se abordó ese tema así como también en el desarrollo se especifica cómo utilizar el software Logstash y Elasticseach que es uno de los tantos sistemas que se utilizan en la actualidad.

También es importante resaltar que de los ataques descritos en la sección de desarrollo, se lograron obtener resultados positivos en cuánto como a detectarlos y especificar el procedimiento a seguir. Además, en la introducción hacia los sistemas SIEM y de los ataques, se enumera y se describe los métodos más comunes utilizados por atacantes al momento de la arremetida hacia alguna red en particular. Conocer los métodos más comunes y socializarlos es muy importante para que cada vez este mercado sea menos apetecible para gente mal informada y que conoce mucho de informática.

Síntesis.

Expuestas las conclusiones en base a toda la investigación y desarrollo del trabajo queda por último el presentar ciertas sugerencias para los distintos lectores de este trabajo que pretendan ampliar el espectro de alcance del proyecto. Sugerir

mejoras, así como también otros temas para la profundización en la investigación y algunos otros ataques para analizar más a fondo en el desarrollo.

Recomendaciones

En el presente trabajo aunque la mayoría de objetivos se cumplieron y todos se analizaron a profundidad en la sección anterior, es importante resaltar ciertos temas que pueden servir para profundizar en temas relacionados a la detección de ataques de red utilizando sistemas SIEM. A continuación se da pautas de cómo se podría ampliar el alcance del trabajo a ciertos temas discutidos en la introducción o en el desarrollo pero que no fueron tomados en cuenta para cumplir los objetivos del trabajo.

1) Planificar como desarrollar la detección de ataques que se presentan en la introducción de este trabajo

Cómo se mencionó anteriormente algunos ataques a la red no fueron planteados en el desarrollo. Esto debido a que el software utilizado (Logstash) no era suficiente como para la detección de otros ataques bien descritos en la introducción a los ataques de red. Ataques como el de Man-in-the-Middle, SQL Injection o Phishing pueden ser abordados de manera distinta sin necesidad de desarrollar algún procedimiento específico en base a un sistema SIEM.

Por poner un ejemplo concreto, para el análisis de la detección de ataques de SQL Injection sería muy interesante abordar los temas de cómo proporcionar integridad y

seguridad en motores de bases de datos. Con rutinas y validación de datos podría abordarse el cómo detectar este tipo de ataques. Pues, aunque los métodos nombrados son más de prevención, se podría en base a estadística encontrar cuando se intentó burlar la validación de datos y transmitir esos datos en información relevante para la creación de patrones de seguridad.

Para los ataques de suplantación de identidad (phishing) la víctima del ataque puede confiar en el antivirus pagado que tenga para que le advierta de ciertos sitios fraudulentos que tienen como objetivo el robar credenciales a usuarios comunes. No es necesario tener un sistema SIEM monitoreando todas las conexiones de red del usuario, pues aunque detecte todas las conexiones entrantes y peticiones de otros servidores, si no tiene la base de datos actualizadas con los sitios sospechosos por realizar suplantación de identidad, entonces no lanzará advertencias al usuario. Una de las maneras de detectar este tipo de ataques es que cuando usted sea víctima de este tipo de intento de estafa informe a las autoridades competentes, la Asociación de Internautas. Dicha Asociación creó hace varios meses un enlace a través del cual los internautas pueden denunciar los correos que simulan ser entidades bancarias, web falsas o cualquier tipo de estafa por medio del uso de phishing en internet. Para esto solo se debe mandar un correo a phishing@internautas.org adjuntando el mail recibido o la web que intenta el robo de datos. El propósito de la Asociación de Internautas es evitar y erradicar de una vez los posibles intentos de estafas realizado mediante el uso de phishing

Para los ataques de hombre-en-el-medio (Man-in-the-Middle) se podría utilizar software conocidos como midware. Un ejemplo es Wireshark. El analizador de protocolos de red más famoso del mundo. Un proyecto de código abierto que se emprendió en 1998. Es decir, se puede ampliar la gama de estudio de ataques a la red no solo con software de seguridad sino con otros tipos de software que se han descrito en esta sección.

2) Establecer procedimientos generales que puedan adaptarse a distintos SIEM.

Aunque durante todo el desarrollo completo de este trabajo se intentó mostrar detección de ataques enfocados a un sistema SIEM en particular, sería muy interesante que diversos autores que puedan utilizar esta tesis de pregrado como fuente, la amplíen en cuanto a los procedimientos de detección de ataques y los hagan adaptables a distintos sistemas SIEM. Es decir, que sean procedimientos generalizados para no limitar herramientas al momento de realizar monitoreo de seguridad a redes.

Partir de los procedimientos descritos en este cometido sería un buen inicio para cualquier otra investigación. Sin embargo, habría que tomar en cuenta que muchos ataques evolucionan, se reinventan para evadir seguridades y por esto el análisis debe comprender investigación en fuentes más recientes.

3) El análisis de los ataques debe replantearse

La capacidad que se tiene en la actualidad para recopilar información hace posible que cada vez los estudios científicos en la rama de la informática sean de mayor especialización y realizados con mayor frecuencia. Por esta razón sería imperativo el que nuevos trabajos a partir de este tenga como prioridad el encontrar nuevas fuentes de ataques populares para el tiempo de cuando se comience a escribir el análisis. Aunque en esencia los conceptos de los ataques deberían ser los mismos, puede ser que haya ciertas modalidades que los atacantes utilicen más que otros. Por esto y más una investigación profunda de los ataques debe considerarse en cualquier trabajo posterior.

4) Aplicar contexto en un trabajo posterior de seguridad informática

Muchos lectores y personas ajenas a la seguridad informática podrán creer muy lejana la posibilidad de enfrentar ataques por delincuentes informáticos. Aunque en la vida real este tipo de amenazas no sean tomados con la seriedad que lo amerita hay que recordar siempre que se debe hablar en la base de algún contexto en particular. Por poner un ejemplo actual, en el Ecuador se ha desatado un escándalo en el sector público debido a la supuesta contratación por parte del SENAIN (Secretaría Nacional de Inteligencia) a la empresa italiana Hacking Team, especialista en proveer servicios de monitoreo en telecomunicaciones y espionaje. El intentar conectar este tipo de temas especializados en la informática con el contexto, en este caso, político ayuda a que el lector común se interese por este tipo de temas.

Para futuros trabajos, sin duda que se puede aplicar el contexto en cualquier ámbito para de ahí crear un análisis de seguridad informática, como el presente trabajo, más acorde a lectores que no vean este riesgo de los ataques a redes tan cercano a su realidad.

REFERENCIAS

[1] Afzaal, M. Di Sarno, C. Dantonio, S. Romano, L. (2012). *An Intrusion and Fault Tolerant Forensic Storage for a SIEM System.* Dept. of Technol., Univ. of Naples Parthenope, Naples, Italy. November.

[2] Asanger, S. Hutchison, A. (2013). *Experiences and Challenges in Enhancing Security Information and Event Management Capability Using Unsupervised Anomaly Detection*. Dept. of Comput. Sci., Univ. of Cape Town, Cape Town, South Africa.

[3] Avireddy Srinivas, Perumal Varalakshmi, Gowraj Narayan, Srivatsa Ram, Thinakaran Prashanth, Ganapathi Sundaravadanm, Gunasekaran Jashwant, Prabhu Sruthi. (2012). *Random4: An Application Specific Randomized Encrpytion Algortihm to prevent SQL injection*. Dpt. of Information Technology. Madras Institute of Technology. Anna University, TamilNadu, Chennai.

[4] Azodi, A. Jaeger, D. Feng Cheng. Meinel, C. (2013). *A New Approach to Building a Multitier Direct Access Knowledgebase for IDS/SIEM Systems*. Plattner Inst. (HPI), Univ. of Potsdam, Potsdam, Germany.

[5] Azodi, A. Jaeger, D. Feng Cheng. Meinel, C. (2013). *Pushing the Limits in Event Normalisation to Improve Attack Detection in IDS/SIEM Systems*. Hasso Plattner Inst. (HPI), Univ. of Potsdam, Potsdam, Germany. December 2013.

[6] Bellovin Steven M, Merritt Michael. (1992). *Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks*. AT&T Bell Laboratories. Murray Hill, NJ, USA.

[7] Bertino, E. (2012). *Data Protection from Insider Threats*. Morgan & Claypool. Edition 1, 2012.

[8] Cárdenas, A. Manadhata, P. Rajan, S. (2013). *Big Data Analytics for Security*. Univ. of Texas at Dallas, Dallas, TX, USA.

[9] Chen Zhe, Guo Shize, Zheng Kangfeng, Li Haitao. (2009). *Research on Man-in-the-Middle Denial of Service Attack in SIP VoIP*. National Engineering Laboratory for Disaster Backup and Recovery. Beijing University of Posts and Telecommunications, BUPT. Beijing, China. 2009.

[10] Dairinram, P. Wongsawang, D. Pengsart, P. (2013). *SIEM with LSA technique for Threat identification*. Fac. of Inf. & Commun. Technol., Mahidol Univ., Bangkok, Thailand.

[11] *Elasticsearch: Search & Analyze Data in Real Time*. (2015). – Elasticsearch. USA. https://www.elastic.co/products/elasticsearch

[12] Fouladi Ramin F, Seifpoor T, Anarim E. (2013). *Frequency Characteristics of DoS and DDoS Attacks*. Dpt. Of Electrical and Electronics Engineering. Bogazici University, Istanbul, Turkey.

[13] Fu Zhang. (2011). *Mitigating Distributed Denial-of-Service Attacks: Application-defense and Netowrk-defense Methods.* Dpt. of Computer Science and Engineering. Chalmers University of Technology, Gothemburg, Sweden.

[14] Gabriel, R. Hoppe, T. Pastwa, A. Sowa, S. (2009). *Analyzing Malware Log Data to Support Security Information and Event Management: Some Research Results*. Advances in Databases, Knowledge, and Data Applications. DBKDA '09. First International Conference on 1 – 6 March. Business Inf., Ruhr-Univ. of Bochum, Bochum.

[15] García, M. Neves, N. Bessani, A. (2013). *An intrusion-tolerant firewall design for protecting SIEM systems*. Fac. of Sci., Univ. of Lisbon, Lisbon, Portugal.

[16] *Kibana: Explore & Visualize Your Data*. (2015). – Elasticsearch. USA. https://www.elastic.co/products/kibana

[17] Kirushnaamoni R. (2013). *Defenses to curb Online Password Guessing Attacks*. Dpt. Of Computer Science and Engineering. Mepco Schlenk Engineering College, Sivakasi, India.

[18] Kotenko, I. Chechulin, A. (2012). *Common Framework for Attack Modeling and Security Evaluation in SIEM Systems*. Green Computing and Communications (GreenCom), IEEE International Conference on 20-23 Nov. Besancon.

[19] Kotenko, I. Polubelova, O. Saenko, I. (2012). *The Ontological Approach for SIEM Data Repository Implementation*. Lab. of Comput. Security Problems, St. Petersburg Inst. for Inf. & Autom. (SPIIRAS), St. Petersburg, Russia.

[20] Kotenko, I. Chechulin, A. (2013). *Computer attack modeling and security evaluation based on attack graphs*. St. Petersburg Inst. for Inf. & Autom., St. Petersburg, Russia.

[21] Liang Zhenkai, Sekar R. (2005). *Automatic Generation of Buffer Overflow Attack Signatures: An Approach Base don Programa Behavior Models*. Dpt. Of Computer Science. Stony Brook Universitiy, Stony Brook, NY, USA.

[22] *Logstash: Collect, Enrich & Transport Data.* (2015). – Elasticsearch. USA. https://www.elastic.co/products/logstash

[23] Major, S. Fekovic, E. (2014). *Securing intelligent substations: Real-time situational awareness*. A&L Consulting, Melbourne, VIC, Australia.

[24] Novikova, E. Kotenko, I. (2013). *Analytical Visualization Techniques for Security Information and Event Management*. Lab. of Comput. Security Problems, St. Petersburg Inst. for Inf. & Autom., St. Petersburg, Russia.

[25] Orman Hilarie. (2013). *The Compleat Story of Phish*.

[26] Satoh Akihiro, Nakamura Yutaka, Ikenaga Takeshi. (2012). *SSH Dictionary Attack Detection based on Flow Analysis*. Kyusu Institute of Technology. Kitakyushu, Japan.

[27] Shahriar Hossain, Zulkemine Mohammad. (2011). *Information Source-based Classification of Automatic Phishing Website Detectors*. School of Computing. Queen's University, Kingston, Canada.

[28] Vianello, V. Gulisano, V. Kimenez-Peris, R. Patino-Martínez, M. Torres, R. Díaz, R. Prieto, E. (2013). *A Scalable SIEM Correlation Engine and Its Application to the Olympic Games IT Infrastructure*. Fac. de Inf., Univ. Politec. de Madrid, Madrid, Spain.