Universidad de Buenos Aires

Facultades de Ciencias Económicas,
Cs. Exactas y Naturales e Ingeniería

Carrera de Especialización en Seguridad Informática
Propuesta de Trabajo Final

Tema

Honeypot

Título

Honeypot como herramienta de prevención de ciberataques

Autor: Ing. Jorge Ismael Campoverde Armijos

Tutor del Trabajo Final:
Dr. Pedro Hecht

Año de presentación
2018

Cohorte del cursante
2017

Declaración Jurada de origen de los contenidos

Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Trabajos Finales de Maestría vigente y se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual.

Jorge Ismael Campoverde Armijos
Nro. Documento 1104527278
DNI (Argentina): 956993

Resumen

El presente Trabajo Final de Especialización tiene como objetivo presentar de manera general los Honeypot y analizar su importancia en la seguridad dentro de un entorno corporativo. Los Honeypot se hacen necesarios en las organizaciones debido a que los ambientes de TI son cada vez más complejos y manejan una enorme cantidad de eventos de seguridad. En estas condiciones, los Honeypot son una herramienta que permite observar los ataques en tiempo real para detectar un ciberataque, interpretarlo y crear estrategias de mitigación[1], sin exponer los servidores reales.

La importancia de estudiar los Honeypot se debe a que en la actualidad existe una gran dependencia de los sistemas informáticos, en los cuales, la información tecnológica adquiere un alto valor para la unidad de negocios en las organizaciones. Esta importancia de la información tecnológica conlleva a la proliferación de amenazas, que tienden a crecer, a ser cada vez más complejas y a elevar el nivel de los atacantes. Por ello, es de vital importancia contar con herramientas como los Honeypot, dado que permiten contrarrestar y estudiar este tipo de ataques.

Tener un Honeypot correctamente implementado dentro de la organización permite distraer a los atacantes de las máquinas “reales” para advertir al administrador de la red rápidamente que se está produciendo un ataque. A partir de la advertencia de este, el administrador puede estudiarlo a profundidad para implementar un plan de mitigación.[2]

De acuerdo con lo anterior, el objetivo final del presente trabajo es estudiar la implementación de dos Honeypot y analizar los posibles ataques que cada uno de ellos pueden recibir en un entorno corporativo. Esta investigación sobre análisis forense en Honeypot podrá ser de gran ayuda para futuros ciberataques.

Palabras clave: Honeypot, Análisis Forense, Ciberseguridad, Seguridad de Redes.

Tabla de contenido

Tabla de Figuras

Introducción

Debido a que la seguridad informática es una ciencia que abarca una gran cantidad de campos de estudio, la presente investigación describe una herramienta que permite a las organizaciones analizar la actividad ilícita que se produce desde el internet y evaluar sus riesgos. En esta medida, los Honeypot son una herramienta que permiten reconocer de manera rápida y eficiente un ciberataque y rastrearlo.

Los Honeypot son herramientas capaces de detectar ataques nunca antes vistos en su hábitat natural, desde rastrear el fraude programado con tarjetas de crédito hasta el robo de individualidades. Estas herramientas se hacen necesarias en la actualidad, debido a que los ataques cibernéticos a las organizaciones están en crecimiento. Un claro ejemplo de esto es el Ransomware[1] WananCry que infectó y cifró gran cantidad de equipos de grandes compañías como Telefónica, pidiendo un rescate monetario para la recuperación de la información.

Las razones detalladas anteriormente justifican el significativo interés en este campo de estudio. Por ello, resulta fundamental comprender cómo funcionan los sistemas en una organización y descubrir la importancia de la mejora continua en la seguridad. Esto permitirá a los administradores ser conscientes de los riesgos a los cuales está expuesta la empresa y contrarrestar los ataques a través, por ejemplo, de los Honeypot. Esta es una herramienta que permite la recopilación de información de ciencias forenses, logrando disminuir significativamente los costos vinculados a incidentes de seguridad y, consiente la ejecución de un plan de riesgos con mayores datos acerca del ataque.

Considerando los beneficios de estas herramientas y todas las posibilidades que bridan, en este trabajo nos enfocaremos principalmente en identificar los beneficios que poseen dos tipos Honeypot. Estos permiten extraer información para prevenir posteriores ataques y detectan ataques externos e internos de la red en donde se encuentran alojados.

[1]Ransomware es una forma de software malicioso (o malware) que, una vez que se toma una computadora, amenaza con dañarla, generalmente negando acceso a los datos.

La primera parte de este documento describe los antecedentes de la tecnología Honeypot, su arquitectura y, las ventajas y las desventajas de su uso dentro de una organización. De igual manera, describe el uso de la informática forense para el análisis de cada uno de los diferentes tipos de ataque sobre los Honeypot.

La segunda parte, describe la implementación de dos Honeypot en máquinas virtuales y su uso como servidores víctima de ataques. Adicionalmente, se desarrolla un análisis detallado de cada uno de estos ataques y la clasificación de sus vulnerabilidades.

En este documento se publican solamente ataques conocido hacia alguno de los Honeypot. El principal aporte realiza este estudio es el análisis de las ventajas y desventajas de cada tipo de Honeypot en la seguridad ofensiva de una organización. Esto permitirá a los futuros investigadores observar nuevos tipos de ataques para realizar un análisis posterior.

CAPITULO I: Honeypot

Marco teórico

La idea de los Honeypot comenzó en el año de 1991 con dos publicaciones “The Cuckoos EGG” escrita por Clifford Stoll en la cual cuenta su experiencia atrapando a un hacker que se encontraba en su empresa buscando información confidencial, mientras que la publicación titulada “An Evening with Breford” escrita por Bill Chewick nos cuenta la historia de un hacker informático y las trampas que le ponían sus compañeros para atraparlo. [3]

Cronológicamente estos fueron los hitos más importantes en el desarrollo y evolución de los Honeypot.[4]

- El primer Honeypot fue lanzado en el año de 1997 y se denominó “Deceptive Toolkit” y su objetivo era el de usar el engaño para atacar por la espalda.
- En el año 1998 se lanzó el primer Honeypot comercial “Cybercop Sting” el cual introduce el concepto de muchos sistemas virtuales concentrados en un Honeypot.
- En el año 2000 luego de haberse producido varios ataques con gusanos[2] y ocasionando robo de información, varias empresas adoptaron la implementación de Honeypot como medio de investigación y detección de ataques informáticos.
- En el año 2002 la tecnología del Honeypot fue compartida y fue usada por las organizaciones a nivel mundial para detectar y capturar información sobre ataques desconocidos.
- En el año 2005 el proyecto “Honeypot Filipinas” comenzó para promoverla seguridad de la información sobre las islas Filipinas.

Definición de **Honeypot**

Un Honeypot es un software cuya intención es atraer a atacantes simulando que el sistema es vulnerable o débil y que puede ser comprometido. En el ámbito de la seguridad informática es una herramienta que se usa para

[2]Es un malware que tiene la propiedad de duplicarse a sí mismo

recoger información sobre los atacantes y sus técnicas de intrusión, se lo puede considerar como un cebo electrónico.

La esencia de los Honeypot es la contrainteligencia ya que invita a los atacantes a sistemas vulnerables, pero para beneficio del administrador de la red u organización, porque los honeypot no tienen un uso real para los usuarios regulares ya que no pueden acceder a los mismos. [3][5]

En un concepto holístico[3], el uso específico de un honeypot es atrapar actividad maliciosa en la red, y obtener datos de este ataque para luego poder aplicar una acción sobre el mismo. De acuerdo al autor Mokube,I. & Adams M los honeypot se pueden dividir en varios niveles de interacción, entre los cuales tenemos los honeypot de investigación y producción.

Los honeypot de investigación son usados principalmente por bases militares de organizaciones gubernamentales , que tienen como objetivo principal el descubrir nuevas amenazas y aprender las nuevas técnicas de ataque , mientras que los honeypot de producción se implementan en las organizaciones sobre un ambiente productivo para mejorar la seguridad del mismo.[6]

Los Honeypot se encuentran ejecutando varios servicios sobre el servidor en el cual están implementados , entre los cuales encontramos: Telnet[4] (puerto 23),servidor de Protocolo de transferencia de hipertexto (HTTP) (puerto 80), protocolo de transferencia de archivos (FTP[5])(puerto 21), Secure Shell (SHH[6]) (puerto 22) entre otros servicios que permiten la comunicación entre dispositivos .[7]

Ventajas y Desventajas de los Honeypot

Ventajas

- Los Honeypot son simples de entender, configurar e instalar ya que no poseen algoritmos complejos, la simplicidad de la implementación es tan sencilla como conectarlo a la red y listo, son buenos en el uso de

[3] "Del todo o que considera algo como un todo"
[4] Es un protocolo de red utilizado para proporcionar una interfaz de línea de comando para comunicarse con un dispositivo
[5]Es un protocolo de red estándar utilizado para la transferencia de archivos informáticos entre un cliente y un servidor en una red informática
[6]Es un protocolo de red criptográfica para operar servicios de red de forma segura a través de una red no segura.

recursos ya que no necesita grandes cantidades de CPU para el procesamiento.

- En lo que corresponde a costos, no hay necesidad de adquirir nuevas tecnologías o hadware adicional, ya que cualquier computadora puede ser usada como un honeypot.
- Los Honeypot permiten el estudio de nuevos tipos de ataques, lo cual presenta para el administrador de la red una gran ventaja, ya que permite comprender los ataques que se producen en la organización y en base a esto y crear un plan de mitigación.
- La principal ventaja de un Honeypot es que permite descubrir ataques que no son detectados por otros sistemas de seguridad como los IDS[7] (Intrusión Detection System) el Honeypot no necesita una base de datos con firmas actualizadas.
- Los Honeypot no son voluminosos en términos de captura de datos ya que no crean grandes cantidades de datos y solo están tratando con el tráfico malicioso entrante y al centrarse únicamente en este tipo de tráfico hace que la investigación sea mucho más fácil. [8][3][9]

Desventajas

- Son elementos pasivos, por lo que si no están colocados correctamente en la red no lograrán capturar ningún dato y no tendrán ningún valor.
- Al ser vulnerables intencionalmente, pueden ser de gran atracción para los atacantes novatos ScriptKiddies[8] , que usan herramientas públicas automáticas para tratar de vulnerar los sistemas, y si no posee un ambiente controlado el honeypot puede ser utilizado como puente para ataques a otras redes internas.
- Tienen una visión limitada, por lo que se afirma que los Honeypot no sustituyen ningún mecanismo de seguridad, sino que trabajan en conjunto para mejorar el perímetro de seguridad de la organización.

[7]Es un dispositivo o aplicación de software que monitorea una red o sistemas en busca de actividad maliciosa o violaciones de políticas.
[8]Usuarios sin habilidades que usan herramientas ya desarrolladas para tratar de vulnerar sistemas

- El honeypot se puede usar como un zombie[9] para llegar a otros sistemas y ponerlos en peligro[6][4]

Tipos de Honeypot

Existen dos tipos de Honeypot, los de bajo nivel de interacción y los de alto nivel de interacción.

Los Honeypot de bajo nivel de interacción son los más usados debido a su bajo costo de implementación y mantenimiento, porque cuando el atacante logra penetrar el sistema no llega a nada, es decir el Honeypot no tiene un sistema operativo instalado, por ejemplo:

- Un servicio FTP emulado, es el que está escuchando en el puerto 21, y que está emulando un login FTP o probablemente soportará algunos comandos de FTP adicionales, pero no es un riesgo para la seguridad, ya que lo más probable es que no esté ligado a ningún servidor FTP real y el atacante aunque consiga las credenciales de acceso no podrá obtener ningún archivo del servidor. La principal desventaja de este tipo de Honeypot, radica en que registran únicamente una información limitada.[9]

Los Honeypot de alto nivel de interacción son más costosos porque necesitan una infraestructura, hardware y un grupo de ingenieros especializados para poder realizar la implementación. La principal ventaja de este tipo de Honeypot es que tiene la capacidad de capturar grandes cantidades de información referentes al modo de operación de los atacantes, debido a que los intrusos se encuentran frente a un sistema real.

Existe también un Honeypot con un nivel medio de interacción, pero su implementación no está masificada en los entornos de seguridad, aunque brinden un poco más de información con respecto a los de baja interacción, tienen una característica especial y permiten al atacante ejecutar comandos propios del sistema operativo.

[9] Es la denominación asignada a computadores personales que, tras haber sido infectados por algún tipo de malware, pueden ser usados por una tercera persona para ejecutar actividades hostiles

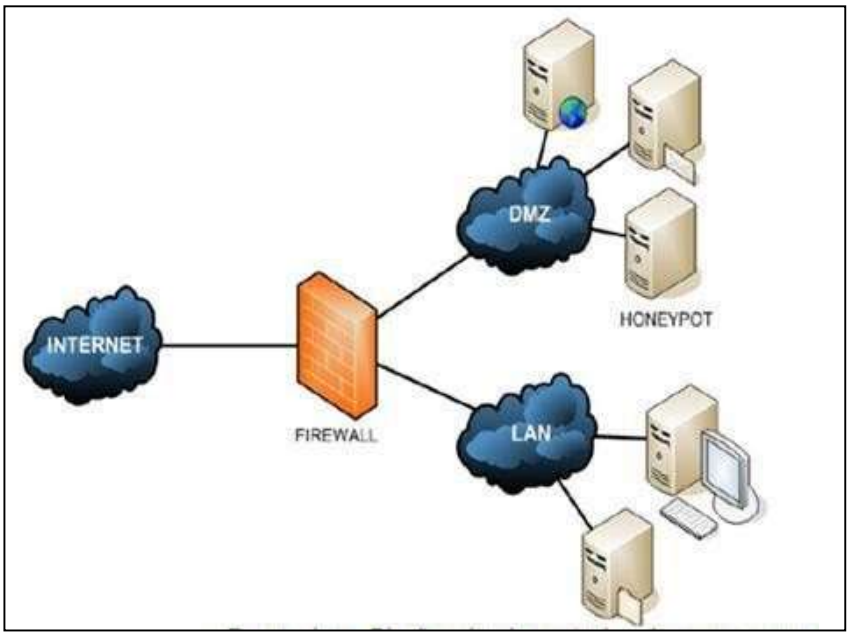

Figura 1.Honeypot de media Interacción[2]

CAPITULO 2: Informática Forense

Historia de la informática forense

Desde 1984, el Laboratorio del FBI y otras agencias que persiguen el cumplimiento de la ley, empezaron a desarrollar programas para examinar evidencia computacional para poder tener pruebas de los ataques que se realizaban contra las empresas y así poder llegar hacia los atacantes[4]. La informática forense comenzó a evolucionar hace más de 30 años en los EEUU, cuando las fuerzas del orden y los investigadores militares comenzaron a ver a los criminales ponerse técnicos.

En marzo del año 1998, a través del subgrupo de trabajo denominado “The High Tech Crime”, conocido como el Grupo de Lyón , encargó a la IOCE[10] el desarrollo de una serie de principios aplicables a los procedimientos de pruebas digitales, así como la armonización de métodos y procedimientos entre las naciones que garanticen la fiabilidad en el uso de las mismas.[10]. Y es así que la informática forense hace su aparición como una disciplina auxiliar de la justicia moderna, para enfrentar los desafíos y técnicas de los intrusos informáticos, para servir como garante de la verdad alrededor de la evidencia digital que se pudiese aportar.

[10]Organización Internacional para la Cooperación en Evaluación

Introducción a la informática forense

El análisis forense digital se basa en un conjunto de técnicas destinadas a extraer información valiosa de discos, sin alterar el estado de los mismos lo que permite buscar datos que son conocidos previamente, tratando de encontrar un patrón o comportamiento determinado para obtener información que se encontraba oculta.

Se puede utilizar la informática forense con finalidades de prevención, es decir, analizar un conjunto de equipos conocer a qué riesgos potenciales están expuestos y mitigar así un posible ataque o incidente informático[11]

La informática forense requiere un conocimiento perfecto de las técnicas de los conocidos como hackers[11], así como del funcionamiento del software en general. La rama de ciencia forense que se usa para la recolección de evidencia digital para luego realizar una análisis de los datos es la informática forense, que tiene una definición semejante a la de ciencia forense, pero con la gran diferencia de que vez los dispositivos electrónicos están involucrados en nuestras investigaciones, entre los cuales tenemos tarjetas de memoria, unidades de almacenamiento , asistentes digitales personales ,es decir todos los dispositivos que tienen la capacidad de aceptar entradas, proporcionar salida, y almacenar datos.[12]

Hay dos áreas diferentes que deben tenerse en cuenta al recoger evidencia forense digital:

1. El proceso de recopilar la evidencia sin alterar su contenido y asegurando que sea admisible en el tribunal por algún litigio.
2. El uso de prácticas forense sanas de cumplimiento de la ley que resulta en que la colección sea admisible en el tribunal[13]

Fases de la informática forense

Existen varias fases de un análisis forense digital entre las cuales tenemos las siguientes.[14]

[11]Un “hacker” o pirata informático es cualquier experto informático experto que utiliza sus conocimientos técnicos para superar un problema

Figura 2-Fases de la informática forense[12]

Identificación del Incidente: Búsqueda y recopilación de evidencias

La primera fase del análisis forense comprende el proceso de identificación del incidente, que lleva consigo la búsqueda y recopilación de evidencias. Si se sospecha que un sistema ha sido comprometido primero se debe revisar que no fue un problema de hardware o software antes de comenzar el análisis forense en búsqueda de evidencia del ataque.

Para iniciar una primera inspección se debe tener en mente la premisa de que se debe conservar la evidencia, por ello no se debe realizar nada que pueda modificarla [11][15]

Recopilación de evidencias

La segunda fase de la metodología de análisis forense consiste en la recopilación de evidencias luego de corroborar que los sistemas informáticos han sido atacados. En este punto se debe valorizar la prioridad del accionar con respecto a dos premisas.

A.- Tener nuevamente operativos los sistemas rápidamente.

B.- Realizar una investigación forense detallada del ataque.

La opción A, permite devolver el sistema a su estado normal cuanto antes, pero este procedimiento hará que pierda casi todas las evidencias que los atacantes hayan podido dejar en “la escena del crimen”.

La opción B en la cual el análisis forense es su prioridad, se basa en una serie de pasos en-caminados a recopilar evidencias que le permitan determinar el método de intrusión al sistema.[11]

Preservación de la evidencia

Para realizar la preservación de la evidencia , se realiza dos copias de las evidencias obtenidas, se generar una suma de comprobación de la integridad de cada copia mediante el empleo de funciones hash[12] tales como MD5[13] o SHA1[14]. [16]

A estas firmas se les debe incluir una etiqueta en cada copia de la evidencia sobre el propio CD o DVD, se incluye también en el etiquetado el cual consta de lo siguiente:

- ¿Dónde, cuándo y quién manejo o examinó la evidencia, incluyendo su nombre, cargo, un número identificativo, fechas y horas, etc.?
- ¿Quién estuvo custodiando la evidencia, durante cuánto tiempo y dónde se almacenó?
- ¿Cuándo se cambie la custodia de la evidencia también deberá documentarse cuándo y cómo se produjo la transferencia y quién la transportó??

Todas estas medidas harán que el acceso a la evidencia sea muy restrictivo y quede claramente documentado, posibilitando detectar y pedir responsabilidades ante manipulaciones incorrectas e intentos de acceso no autorizados[11]

Análisis de la evidencia

Luego de obtener las evidencias digitales y almacenarlas de forma adecuada , se procede a la siguiente fase que es las más laboriosas, el Análisis Forense propiamente dicho, cuyo objetivo es reconstruir con todos los datos disponibles la línea temporal del ataque o también llamada timeline, determinando la cadena de acontecimientos que tuvieron lugar desde el

[12]Una función hash es cualquier función que se puede usar para asignar datos de tamaño arbitrario a datos de un tamaño fijo

[13]El algoritmo MD5 es una función hash ampliamente utilizada que produce un valor hash de 128 bits.

[14](Secure Hash Algorithm 1) es una función hash criptográfica que toma una entrada y produce un valor hash de 160 bits (20 bytes) conocido como mensaje diges.

instante inmediatamente anterior al inicio del ataque, hasta el momento de su descubrimiento.[11][17]

Documentación del Incidente.

Tan pronto como el incidente haya sido detectado, es muy importante comenzar a tomar notas sobre todas las actividades que se lleven a cabo. Cada paso dado debe ser documentado con fecha y hora desde que se descubre el incidente hasta que finalice el proceso de análisis forense, con el objetivo de que existan menos posibilidades de error a la hora de gestionar el incidente.[11][17]

CAPITULO 3: Honeypot Cowrie y Glastopf

Honeypot Cowrie

Introducción

Un método muy frecuente para tener acceso de forma remota a cualquier sistema operativo o servidor de forma remota es a través de Secure Socket Shell SSH[15], que es un protocolo de red que proporciona a los administradores una manera criptográficamente segura para acceder al sistema. La principal característica de este protocolo es que proporciona una fuerte autentificación y asegura las comunicaciones de datos cifrados entre los equipos que se conectan, para poder ejecutar comandos o mover archivos de un ordenador a otro[18]

Para estudiar las actividades ejecutadas por los atacantes después de ingresar a un sistema a través de SSH, se usa un honeypot. El honeypot Cowrie es un honeypot de interacción media, entre las principales características tenemos que nos permite crear un sistema de archivos falso simulando un sistema operativo Debían y el intruso puede navegar entre los directorios, pero no puede destruir nada y cada una de estas acciones se irá registrando para luego poder ser analizada por el administrador de la red.[19]

Características

Entre las características principales del honeypot tenemos las siguientes

[15]Es un protocolo de red criptográfica para operar servicios de red a través de una red no segura.

- Integra un sistema de archivos completamente artificial similar a un mecanismo de Debian 5.0 que tiene la capacidad de construir y eliminar archivos.
- Permite obtener contenidos a un archivo de texto, simulando que son contenidos reales de un sistema en producción. Esto atrapará al atacante que intentará usar el comando "cat" para archivos como /etc/ passwd
- Almacena los registros en el formato UML, de esta manera, podremos estudiar detalladamente todos los pasos que ha realizado un posible atacante.
- Permite ver cuáles fueron los archivos que se descargaron a lo largo de la sesión SSH (simulando los comandos de descarga que son wget) para un análisis posterior.[20][21]
- Los intentos de autentificación se guardan en un archivo para un procesamiento fácil de gestión de registros[22]

Directorios de Cowrie

Entre los directorios más importantes tenemos los siguientes.

- cowrie.cfg - archivo de configuración de Cowrie
- data / fs.pickle - sistema de archivos falso
- data / userdb.txt - credenciales permitidas o no permitidas para acceder al honeypot
- dl / - archivos transferidos desde el atacante al honeypot
- honeyfs / - contenido de archivos para el sistema de archivos falso
- log / cowrie.json - salida de transacción en formato JSON
- txtcmds / - contenido del archivo para los comandos falsos[19]

Requisitos de instalación.

El software necesario para utilizar e honeypot Cowrie es el siguiente: Un sistema operativo (puede ser plataformas Debian, CentOS y Windows)

- Python 2.7+ (Python 3 no es soportado)
- Dependencias de Python
- Ambientes virtuales de python
- Interfaz Zope[20]

Honeypot Glastopf

Introducción

El proyecto de Glastopf fue fundado en el año 2008 por Lukas Rist, desde entonces alrededor de diez personas han contribuido al proyecto, la mayoría solicitando características o sugiriendo nuevas ideas que podrían ser implementadas.

En la actualidad se presume que el 80% de los intentos de ataque hacia una organización son contra las aplicaciones Web, por lo que las empresas no pueden correr el riesgo de que exista alguna filtración de información o que sus sitios se vean comprometidos y se instale malware y este sea entregado a sus clientes lo que causaría un daño irreparable para la organización. Teniendo en cuenta estas estadísticas y al conocer las posibles consecuencias que podrían ocurrir si una aplicación web es comprometida, es necesario el uso de un honeypot para conocer el vector del ataque y así tener una respuesta ante estos incidentes de seguridad.[8]

El honeypot Glastopf es capaz de emular cientos de vulnerabilidades para reunir datos que apunten hacia al ataque a una aplicación sobre una organización, una de las principales características de Glastopf con respecto a otros Honeypot es que se centra en responder con la respuesta correcta al atacante que explota la aplicación web objetivo y no al a vulnerabilidad en particular. [8][23]

El principio detrás de esto es muy simple: responder al ataque usando la respuesta que el atacante espera de su intento de explotar la aplicación web.

Características y Tipos de ataque

Tipos de Ataques.

Figura 3.Tipos de ataque del Honeypot Glastopf[24]

- Inclusión local de archivos (Local File Inclusion): Es conocido también como LFI, y es el proceso de incluir archivos que están alojados en el servidor a través de la explotación de procedimientos de inclusión vulnerables.
- Inclusión remota de archivos (Local Remote Inclusion): Es conocido también como LRI, y es el proceso de incluir archivos que están alojados en algún otro servidor a través de los cuales se puede realizar la explotación de procedimientos de inclusión vulnerables.
- Inyección SQL (SQL Injection): También conocido como SQLi, consiste en la inyección de una consulta SQL a través de los datos de entrada de la aplicación del cliente, a un formulario a través de variables GET o variables POST.

Características de Glastopf:[8]

- El honeypot Glastopf no emula la vulnerabilidad en sí, sino que emula el tipo de ataque, es decir reacciona dependiendo de los comandos que son ejecutados por parte del atacante.
- Tiene un diseño modular para incluir nuevas capacidades de registro o manejadores del tipo de ataque.
- Tiene un registro HPFeeds que facilita la recopilación centralizada de datos
- Posee un módulo que permite la emulación del ataque conocido como inclusión remota de archivos a través de un sandbox integrado desarrollado en el lenguaje PHP. Los atacantes suelen hacer uso de los motores de búsquedas y solicitudes para localizar fallas dentro de las aplicaciones de las organizaciones , por lo cual Glastopf para mitigar este tipo de ataques , extrae esas palabras clave conocidas como dork[16].

[16] Dork es la palabra que hace referencia a la configuración avanzada de Google para buscar páginas que sean vulnerables a ciertos ataques

Perspectiva general de funcionalidad del honeypot Glastopf

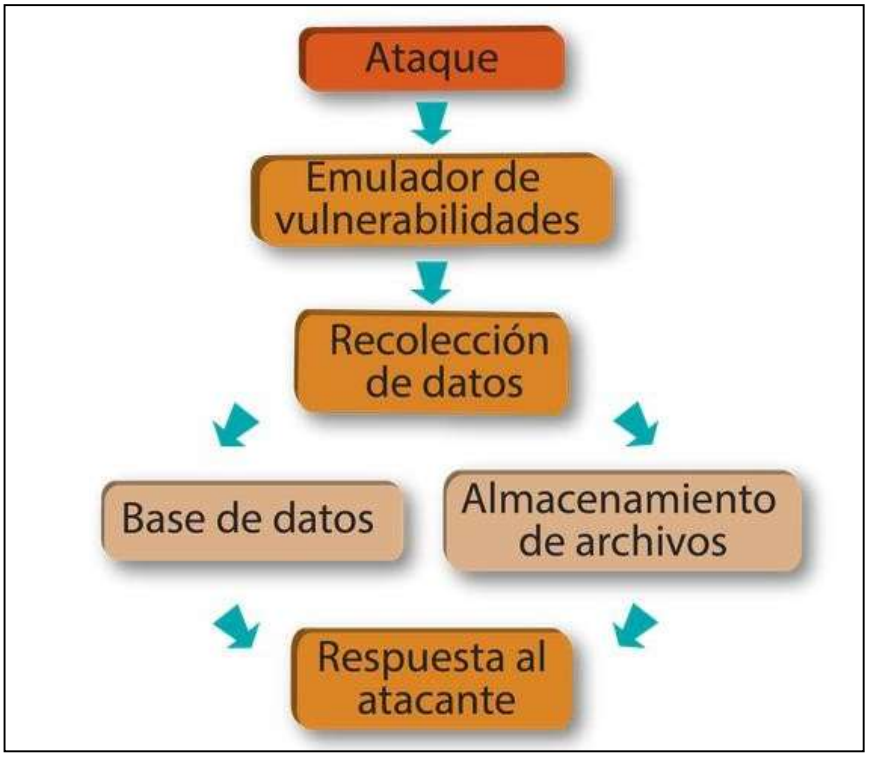

Figura 4. Panorama general de funcionalidad. Lukas Rist (Traducción de UNAM-CERT) [25]

Requisitos de instalación

- Instalar las dependencias requeridas del sistema operativo
- Instalar las dependencias de PHP
- Instalar pylibinjection

CAPITULO 4: Simulación de ataques a los Honeypot

Ataques al honeypot Cowrie.

Con el objetivo de comprobar la funcionabilidad del Honeypot Cowrie, se desarrolló una arquitectura virtual que se muestra en la figura 5, en la cual se evidencia un escenario conceptual de cómo se realizará el ataque. El honeypot Cowrie está instalado sobre una máquina virtual con un sistema operativo Ubuntu 14.04 con una IP=192.168.52.143 y está escuchando en el puerto 22, la máquina atacante tiene un sistema operativo Kali Linux con una dirección IP=192.168.52.141 y la máquina host tiene una IP=192.168.52.10.

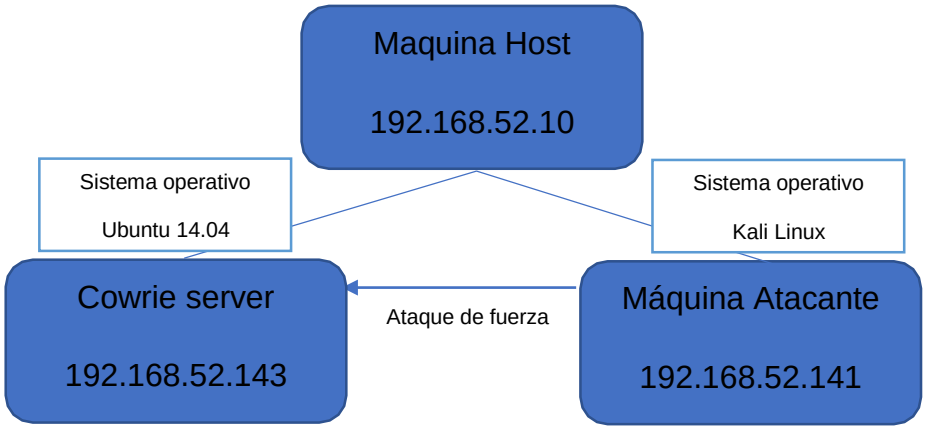

Figura 5. Escenario del ataque

Para realizar este ataque, primero se ejecutará un escaneo de la maquina víctima para revisar cuales son los puertos y servicios que están corriendo, para lo cual ejecutaremos el siguiente comando en nuestra máquina atacante que tiene el sistema operativo Kali Linux con permisos de superusuario

root@kali:# nmap –sV –p 1-1024 192.168.52.143

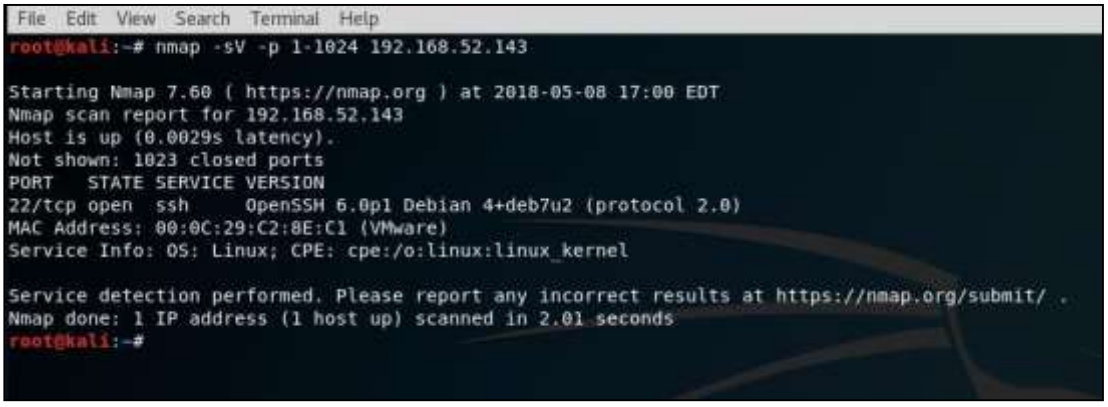

Figura 6 Resultado del escaneo con nmap

En la figura 6 se puede observar que el puerto 22 está abierto ejecutando un servicio ssh, ya que en la máquina escaneada se encuentra ejecutando el honeypot Cowrie.

Luego de la etapa de reconocimiento y detectar que existe un vector de ataque, se procederá usar la herramienta medusa que es usada para obtener acceso a servidores ssh a través de fuerza bruta. Medusa es una herramienta desarrollada para llevar ataques del tipo fuerza bruta de manera paralela y modular, se destaca en su uso por la rapidez en que realiza dichos ataques.

El aplicativo permite llevar a cabo múltiples ataques a diferentes sistemas con diferentes usuarios[25]

Para ejecutar la herramienta medusa, se necesita conocer varios parámetros como la ip de la máquina víctima, un diccionario con nombres de usuario, un diccionario con una lista de passwords y el protocolo que se usará para conectarse. En esta prueba de concepto, supondremos que el usuario es el que viene por defecto y omitiremos el diccionario de usuarios en la ejecución del comando.

El comando que usaremos es el siguiente:

root@kali: # medusa –u root –P rockyou.txt –h 192.168.52.143 –H ssh

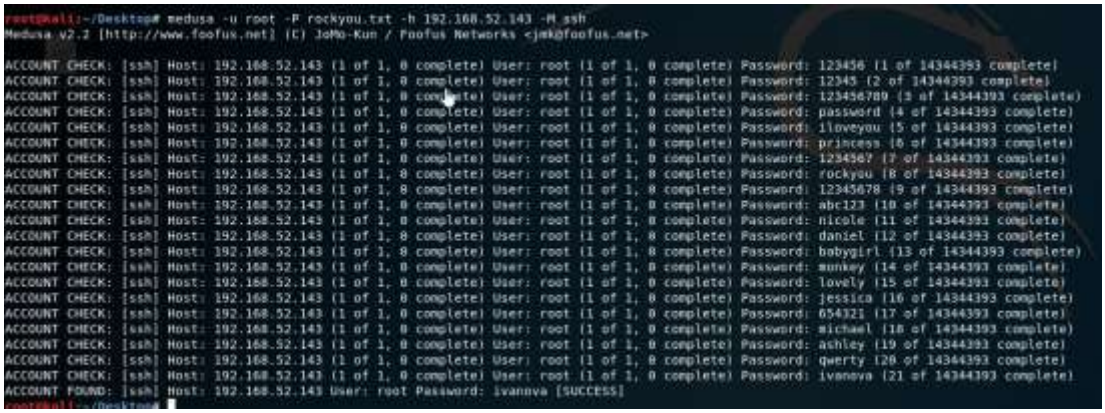

Figura 7. Descubrimiento de Usuario y Password del servidor

En la figura 7, se observa que luego de varios intentos fallidos, el ataque es exitoso encontrando el usuario/password correcto para acceder al sistema, mostrándonos que las credenciales correctas son usuario=root y clave=ivanova.

Luego de obtener las credenciales, el atacante procederá a verificar si estas son correctas, y tratará de entrar en el servidor ingresando la clave que acaba de crackear a través de fuerza bruta, encontrado que son correctas y cayendo en el señuelo de un falso servidor.

Figura 8 .Sistema de Archivos Cowrie

En la figura 8, se puede observar que el atacante se encuentra dentro del honeypot Cowrie y está con permisos root, por lo cual no es necesario usar un exploit[17]para aprovechar una vulnerabilidad en el kernel para escalar privilegios ,ya que es posible acceder a cualquiera de los directorios.

Una de las ventajas de este honeypot es que permite al atacante crear carpetas y borrar archivos, lo que supone que está navegando en un servidor productivo por lo cual lo primero que buscará es el archivo passwd y shadow que es en donde se encuentran los usuarios y passwords de los usuarios del sistema respectivamente, para lo cual ejecutará el siguiente comando.

```
root@kali: # cat /etc/passwd
```

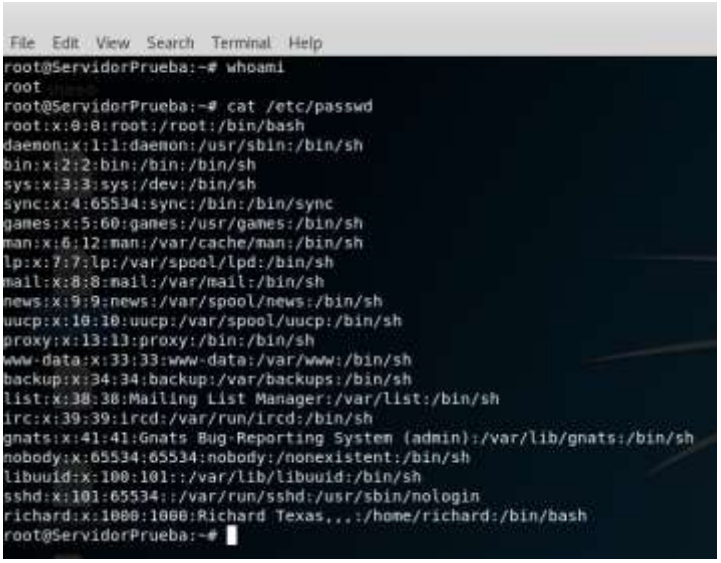

Figura 9 Contenido del archivo /etc/passwd

Para acceder al archivo que tiene las claves encriptadas usaremos el siguiente comando

```
root@kali:# cat /etc/shadow
```

[17] Una herramienta de software diseñada para aprovechar una falla en un sistema informático, generalmente con fines maliciosos.

Figura 10 Contenido del archivo etc/shadow

Análisis de los eventos en el Honeypot Cowrie

Los eventos que se generan en el honeypot Cowrie se guardan dentro del directorio de logs del sistema de archivo, en la figura 11 se observa los logs en un formato poco entendible ya que se encuentran en un formato de archivo de tipo json[18] , para una mejor visualización de los mismos se usa a un módulo llamado cowrie-logviewer el cual nos permite observar los logs de una manera interactiva, para tener una respuesta rápida a un ataque de fuerza bruta.

Figura 11. Eventos en el honeypot Cowrie

[18]Es un formato liviano de intercambio de datos. Es fácil para los humanos leer y escribir

Luego de instalar las dependencias correspondientes para usar el módulo Cowrie-logviewer, se levanta el servicio de Cowrie-logviewer para lo cual

root@kali: # python cowrie-logviewer.py

ejecutaremos el siguiente comando.

```
cowrie@ubuntu:~/cowrie/cowrie-logviewer$ python cowrie-logviewer.py
* Serving Flask app "cowrie-logviewer" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Figura 12. Servicio Logviewer ejecutándose

Una de las principales características del honeypot, es su función de reproducción de sesiones en tiempo real , ya que permite visualizar cada vez que el atacante entra al servidor y navega alrededor de los directorios del sistema, y cada comando que ejecute se graban como una película, lo que permite al administrador de la red analizar las actividades realizadas por el atacante paso a paso, como se puede observar en la figura 13.

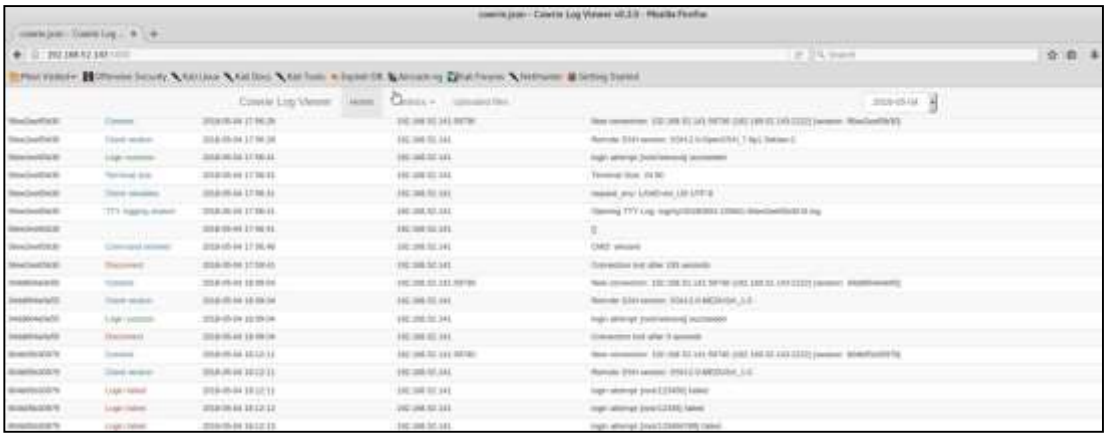

Figura 13. Visualización de los ataques en Cowrie Log Viewer

En la figura 14 se puede observar todos los comandos y ver las actividades exactas de lo que hizo el atacante en el servidor, se observa que ingreso comandos como cat /etc/shadow, nano, sysinfo para obtener información del sistema, y el administrador de la red obtiene la hora exacta del ataque lo que le permite realizar un plan de mitigación para manejar los incidentes.

Figura 14 . Comandos ejecutados en el FileSystem falso

Ataques al Honeypot Glastopf.

Con el objetivo de comprobar la funcionabilidad del Honeypot Glastopf, se desarrolló una arquitectura virtual que se muestra en la figura 15, en la cual se detalla un escenario conceptual de cómo se realizará el ataque. El honeypot Glastopf está instalado sobre una máquina virtual con un sistema operativo Kali Linux con una IP=192.168.52.146 y está escuchando en el puerto 8080, la máquina atacante tiene un sistema operativo Ubuntu con una dirección IP=192.168.52.143 y la máquina host tiene una IP=192.168.52.10.

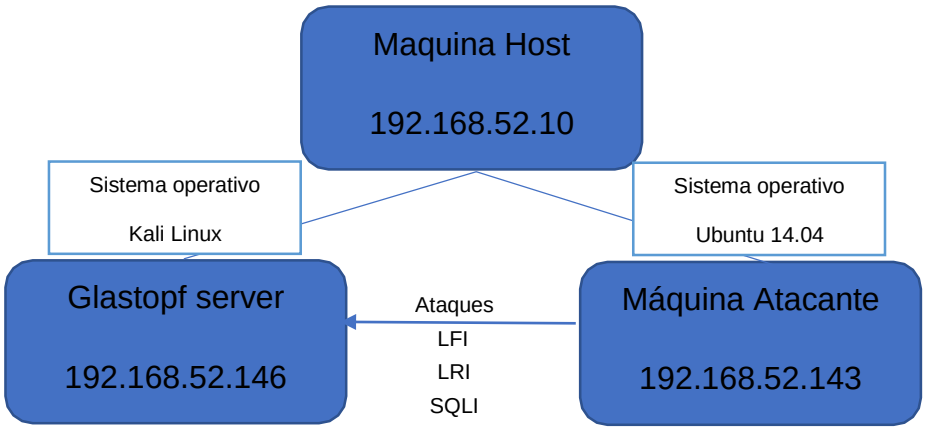

Figura 15 Escenario del Ataque

Para testear el honeypot, el atacante tratará de acceder al servidor web desde la máquina atacante que se encuentra en la misma red LAN, pudiendo acceder correctamente como se muestra en la figura 16.

Figura 16. Visualización de la Web desde la máquina atacante

Desde la máquina atacante, se realizará un ataque de inclusión de archivo local para poder acceder ilegalmente al archivo que se encuentra en el directorio /etc/passwd. En la figura 17 y 18 respectivamente se puede ver que

```
curl –k http://192.168.52.146:8080/x?id=../../../etc/password
```

el atacante ataca el servidor web Glastopf que se ejecuta en la dirección IP 192.168.56.146 a través del puerto 8080 y logra obtener exitosamente los archivos password y shadow del Sistema de Archivos del honeypot.

Para obtener el archivo passwd, se ejecuta el siguiente comando.

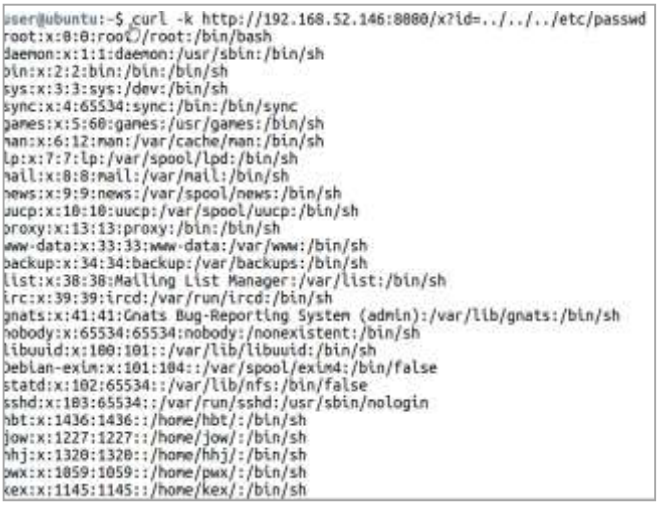

Figura 17. Archivo passwd obtenido a través de ataque LFI

Para obtener el archivo shadow, se ejecuta el siguiente comando.

```
curl –k http://192.168.52.146:8080/x?id=../../../etc/shadow
```

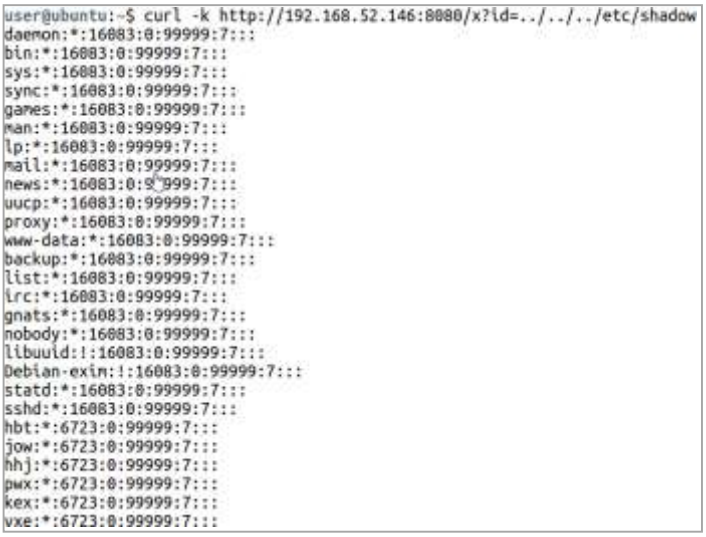

Figura 18 Archivo shadow obtenido a través de LFI

Análisis de los eventos en el Honeypot Glastopf

Como se observa en la figura 19 , en los registros del honeypot se puede evidenciar que el ataque fue realizado desde un equipo con la dirección IP 192.168.52.143, la hora en la que se realizó y el método utilizado que en este caso es GET[19] y los archivos que el atacante quiere obtener que son el archivo passwd y el shadow del directorio /etc.

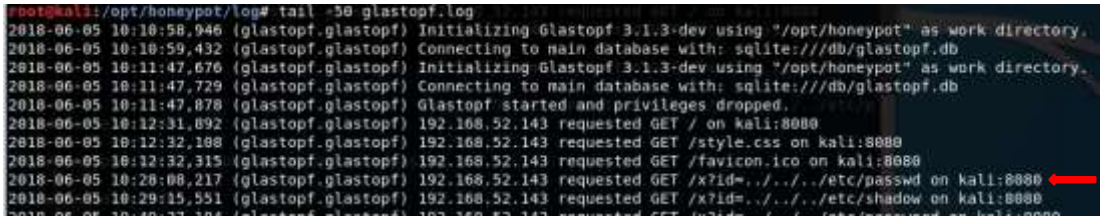

Figura 19 . Registro de ataques en el Honeypot

El honeypot Glastopf además de darnos un registro completo de los datos del atacante, nos permite identificar el tipo de ataque que realizó, para lo cual usaremos la base de datos sqlite3que se instala de forma conjunta con el honeypot y tiene las siguientes tablas.

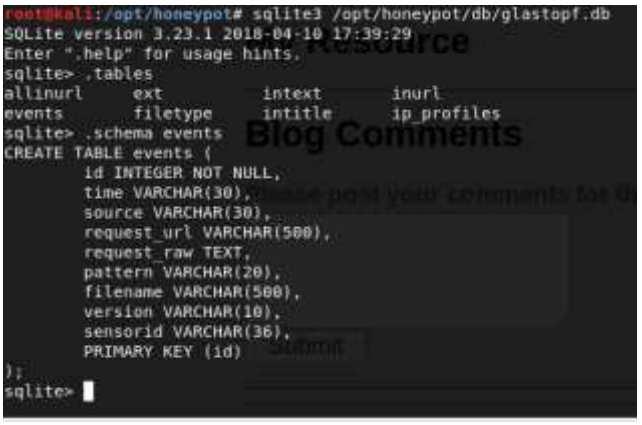

Figura 20 . Campos de la tabla eventos

El honeypot Glastopf nos permite extraer un identificador, el agente a través del cual se ejecutó el ataque, y el tipo de ataque que se realizó para posteriormente hacer una auditoría, para lo cual usaremos el siguiente comando:

sqlite >select id, time, source, request_url, patrón de eventos;

[19] Información se envía de forma visible al servidor a través de la URL

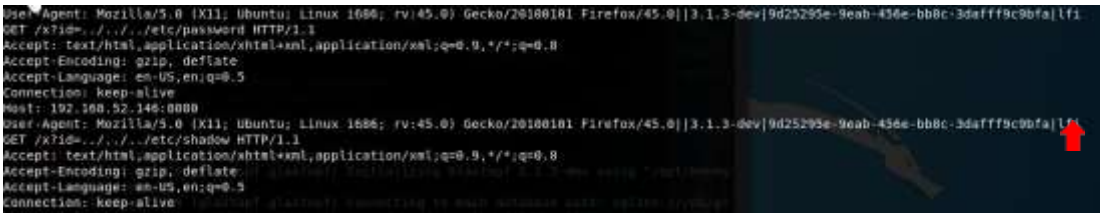

Figura 21 Análisis de los registros de la base de datos sqlite3

En la figura 21, se observa que al final del registro se imprime "lfi", lo cual indica que se trata de un ataque de inclusión local de archivos (LFI) en el servidor web, el cual se efectuó desde el un Agente de Usuario Mozilla/5.0, lo que nos permite saber que lo hicieron directamente desde el navegador y nos muestra que la máquina atacante es un Ubuntu Linux i686.

Los ataques LFI se pueden ejecutar desde diferentes agentes de usuario , como se observa en la figura 22 , el tipo de agente usado es curl[20]que a través del protocolo http logra extraer exitosamente el archivo shadow.

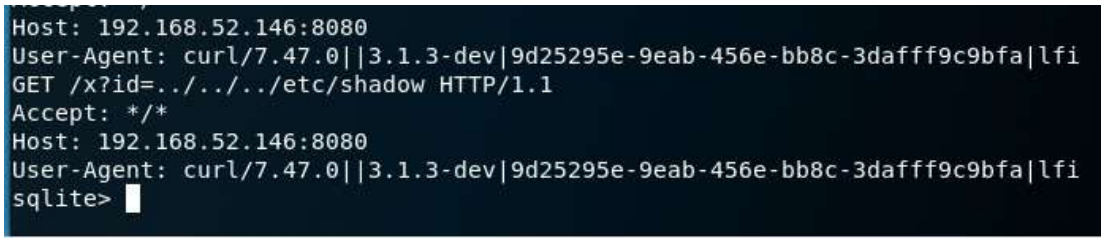

Figura 22 Ataque de Inclusión local de archivos a través del agente de usuario Curl

En los registros del honeypot, podemos observar que se ejecutó también un ataque de inclusión remota de archivos (RFI) en el cual el atacante intenta incluir archivos remotos a través de la explotación de procedimientos de inclusión vulnerables implementados en la aplicación web, como se puede observar en la figura 23.

Figura 23 Ataque de Inclusión remota de archivos

[20]CURL es un proyecto de software que proporciona una biblioteca y una herramienta de línea de comandos para transferir datos utilizando varios protocolos

El honeypot Glastopf permite también registrar los eventos que son generados por herramientas de pentesting[21] automáticas como sqlmap que es es una herramienta de código abierto que automatiza el proceso de detección y explotación de fallas de inyección SQLa través de los datos de entrada de la aplicación del cliente, por ejemplo: formularios, variables , etc y esto permite leer , modificar o eliminar la base de datos del servidor. Para el uso de esta herramienta se necesita conocer la ip del servidor web, por lo cual se ejecuta el siguiente comando.

```
root@:# sqlmap –u 192.168.52.143
```

Figura 24 Ataques a través de software sqlmap

En la figura 24 se observa como el honeypot detecto cual es el tipo de ataque y el Agente de usuario del cual fue ejecutada sqlmap/1.1.12 que es la herramienta sqlmap.

En la auditoria de los registros del honeypot también se detectó un ataque a través de un escáner de vulnerabilidades, en este caso se usó el escáner de vulnerabilidades Nikto. En la figura 25 se puede observar que en agente de usuario es Nikto/2.1.6 con un tipo de ataque xss[22] que es una vulnerabilidad de seguridad informática que normalmente se encuentra en las aplicaciones web y permite a los atacantes inyectar scripts del lado del cliente en páginas web vistas por otros usuarios.

[21]Una prueba de penetración, coloquialmente conocida como una prueba de pluma, es un ataque simulado autorizado en un sistema informático, realizado para evaluar la seguridad del sistema.

[22] Es un tipo de inyección, en la que los scripts maliciosos se inyectan en sitios web benignos y confiables.

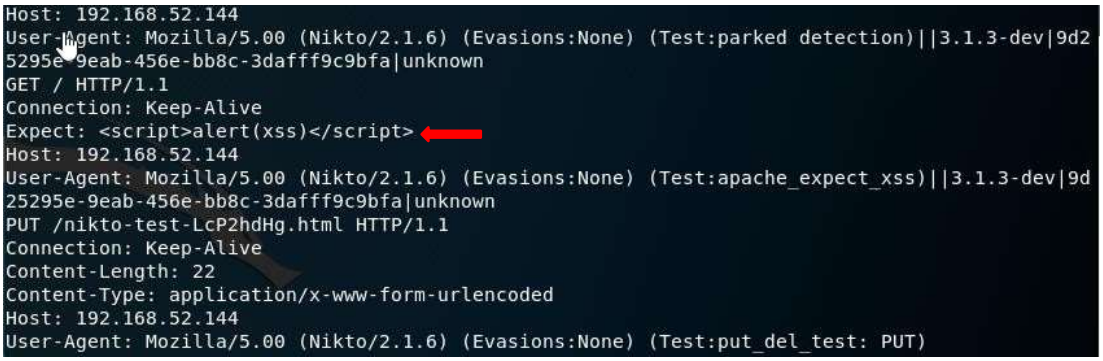

Figura 25 Ataques a través de herramienta Nikto

Conclusiones

Con este trabajo de fin de especialización se ha pretendido realizar una incursión en el apasionante y novedoso mundo de los Honeypot. Se han expuesto en detalle sus características, particularidades y ventajas al ser implementados dentro de una organización en un ambiente productivo. Esto promete llevar la seguridad ofensiva a un nivel más complejo en lo que se refiere a ciberseguridad.

Podemos afirmar que se cumplieron los objetivos propuestos en este trabajo, dado que se implementaron dos tipos de Honeypot en una arquitectura de red virtual, se recabó información de los posibles tipos de ataques, se usó una herramienta externa para mostrar estadísticas y obtener información acerca del tipo de ataque que se ejecutaron. Conocer al atacante en la fase de escalación de privilegios es fundamental para los administradores de la red. Esto les permite construir un perfil del atacante para detectar las vulnerabilidades conocidas como *0 days*, puesto que estas son una de las principales causas de hackeo masivos a organizaciones.

El desarrollo de herramientas como los Honeypot ha sido de gran utilidad no solo en el ámbito académico, sino también en el ámbito laboral, ya que, al trabajar en Ciberseguridad, el conocimiento de este tipo de herramientas me permitió mejorar mis habilidades en lo que se refiere a seguridad defensiva. En este sentido es importante indicar que los Honeypot son solo una herramienta y que su efectividad depende de los administradores de la red porque cada organización tiene sus propias necesidades.

En conclusión, este trabajo permitió verificar el funcionamiento de dos Honeypot implementados en una arquitectura virtual y testear completamente sus funcionalidades. Se puede decir que estas herramientas cumplen un rol muy importante en una organización al momento de detectar intrusos y ataques en la red interna y externa, permitiendo que estas sean más seguras.

Bibliografía

[1] E. Esteban and C. Ignacio, “Honeynets como herramienta de prevención e investigación de ciberataques,” 2013.

[2] R. E. J. Sutton, “How to build and use a Honeypot,” 2005.

[3] T. Philippine, “1 . 2 Definition of a Honeypot : 1 . 1 History of Honeypot : 2 . THE IDEA OF HONEYPOT :,” 1991.

[4] E. Superior and I. E. Secci, “Escuela Superior de Ingeniería Química Sección de Estudios de Posgrado e,” 2007.

[5] L. Spitzner, “Honeypot: Catching the insider threat,” *Proc. - Annu. Comput. Secur. Appl. Conf. ACSAC*, vol. 2003–Janua, pp. 170–179, 2003.

[6] A. Sharma, “Honeypot in Network Securoty,” *Int. J. Tech. Res. Appl.*, vol. 1, no. 5, pp. 7–12, 2013.

[7] R. Rehman, *Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID*. 2003.

[8] O. Yousuf, “Master Thesis Analysis and Deployment of Honeypot Solutions on Single Board Computers,” *Mastet Inf. Technol. Frankfurt Univ. Appl. Sci.*, no. April, 2016.

[9] S. E. Luis, T. José, and E. López, “MONITORIZACIÓN Y PREVENCIÓN DE ATAQUES,” 2017.

[10] F. Rodríguez and A. Doménech, “La Informática Forense: El Rastro Digital del Crimen,” *Derecho Y Cambio Soc.*, vol. 25, no. 1, pp. 1–9, 2011.

[11] M. López Delgado, “Análisis Forense Digital,” *Copyr. 2.006 - 2.007*, vol. Segunda Ed, p. 40, 2017.

[12] G. Rivas, “Metodología para un análisis forense,” 2014.

[13] I. Resendez, P. Martinez, and J. Abraham, “An Introduction to Digital Forensics,” *ACET J. Comput. Educ. Res.*, vol. 6, no. 1, 2010.

[14] J. Giordano and C. Maciag, “Cyber Forensics: Una perspectiva de operaciones militar.”

[15] D. Pinto, “Metodología de análisis forense orientada a incidentes en dispositivos móviles,” *Maskana*, vol. 6, no. Ed. Esp., pp. 31–41, 2014.

[16] R. G. Lerena and Info-lab, “Informática Forense,” pp. 1–12, 2016.

[17] C. Döring, “Improving network security with Honeypot,” *Honeynet Proj.*, p. 123, 2005.

[18] A. Eduardo and C. Quezada, “Autopsy 3,” 2015.

[19] T. Mitchell, S. Cyber, and S. Engineer, “Defend Your Infrastructure from Evil with Kippo / Cowrie Honeypot.”

[20] Micheloosterhof, “Cowrie,” 2018. [Online]. Available: https://github.com/micheloosterhof/cowrie.

[21] J. L. Vives, “Universidad Autónoma de Madrid 1,” no. 1, pp. 1–19, 2016.

[22] Takhion, “Use the Cowrie SSH Honeypot to Catch Attackers on Your Network,” 2018. [Online]. Available: https://null-byte.wonderhowto.com/how-to/use-cowrie-ssh-honeypot-catch-attackers-your-network-0181600/.

[23] S. A. T. Balderas, “Glastopf: Honeypot de aplicaciones web – I.” [Online]. Available: https://revista.seguridad.unam.mx/numero25/glastopf-honeypot-de-aplicaciones-web-i.

[24] C. S. Vetsch, M. Koßin, and M. Mauer, “Know Your Tools : Glastopf,” *Honeynet Proj.*, pp. 1–29, 2010.

[25] L. Rist, “Glastopf - Looking for trouble ?,” 2011.