UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

Análisis e Implementación de una Solución Honeypot para un Entorno Experimental

TRABAJO DE TESIS
PARA OBTENER EL GRADO DE
Ingeniero en Redes

PRESENTA
Br. Julio Cesar Matus Chan

DIRECTOR DE TESIS
Ing. Pablo Velarde Alvarado

ASESORES

Dr. Homero Toral Cruz

Dr. Freddy Ignacio Chan Puc

L. I. Luis Fernando Mis Ramírez

M.T. Martín Antonio Santos Romero

CHETUMAL QUINTANA ROO, MÉXICO, DICIEMBRE DE 2017

UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

TRABAJO DE TESIS ELABORADO BAJO SUPERVISIÓN DEL COMITÉ DE ASESORÍA Y APROBADO COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:

INGENIERO EN REDES

Comité de Trabajo de Tesis

Universidad de Quintana Roo
DCi DIVISIÓN DE CIENCIAS E INGENIERÍA

DIRECTOR:

Ing. Pablo Velarde Alvarado

ASESOR:

Dr. Homero Toral Cruz

ASESOR:

Dr. Freddy Ignacio Chan Puc

UNIVERSIDAD DE QUINTANA ROO
SERVICIOS ESCOLARES
TITULACIONES

CHETUMAL QUINTANA ROO, MÉXICO, DICIEMBRE DE 2017

DEDICATORIA

A:

Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Mis Padres Juan Matus Morales y Rosalía Chan Pech, por darme la vida, quererme mucho, creer en mí y porque siempre me apoyaron. Mamá y Papá gracias por darme una carrera para mi futuro, todo esto se los debo a ustedes.

Mis hermanos, Juan Alberto Matus Chan, María del Carmen Matus Chan y Mariana Guadalupe Pinto Chan, por estar conmigo y apoyarme siempre, los quiero mucho.

Mi novia Ana Mildre Velasco Ruiz y los pequeños Robert Alejandro Ruiz y Oddete Ruiz, por estar en esta nueva etapa conmigo, su ayuda ha sido fundamental, han estado conmigo incluso en los momentos más turbulentos. Este proyecto no fue fácil, pero estuviste motivándome y ayudándome hasta donde tus alcances lo permitían. Te lo agradezco muchísimo, amor, gracias por su gran apoyo.

Y no me puedo ir sin antes decirles, que sin ustedes a mi lado no lo hubiera logrado, tantas desveladas sirvieron de algo y aquí está el fruto de todo este esfuerzo. Les agradezco a todos ustedes con toda el alma el haber llegado a mi vida y el compartir momentos agradables y momentos tristes, pero esos momentos son los que nos hacen crecer y valorar a las personas que nos rodean. Los quiero mucho y nunca los olvidare.

"El azar no favorece más que a los espíritus preparados"

Atentamente: Julio Matus

18 de diciembre de 2017

Página intencionalmente en blanco

AGRADECIMIENTOS

A DIOS…

A Dios por brindarme la oportunidad de obtener otro triunfo personal, darme salud, sabiduría y entendimiento para lograr esta meta, me permites sonreír ante todos mis logros que son el resultado de tu ayuda, este trabajo de tesis ha sido una gran bendición en todo sentido y te lo agradezco, y no cesan mis ganas de decir que es gracias a ti que esta meta está cumplida.

A mis Padres…

Gracias a mis padres Juan Matus Morales y Rosalía Chan Pech, por ser los principales promotores de mis sueños, gracias a ellos por cada día confiar y creer en mí y en mis expectativas, gracias a mi madre por estar dispuesta a acompañarme cada larga y agotadora noche de estudio; gracias a mi padre por siempre desear y anhelar siempre lo mejor para mi vida, gracias por cada consejo y por cada una de sus palabras que me guiaron durante mi vida. De igual manera por el apoyo incondicional, el amor recibido, la dedicación y la paciencia con la que cada día se preocupaban por mi avance y desarrollo, es simplemente único y se refleja en la vida de un hijo.

A mis hermanos…

A mis hermanos María del Carmen Matus Chan, Juan Alberto Matus Chan y Mariana Guadalupe Pinto Chan, que de una u otra forma a lo largo de nuestras vidas han estado en mi vida, para reír, llorar y solidarizarnos, a ustedes mis herman@s queridos muchas gracias.

A mi Novia…

A mi novia Ana Mildre Velasco Ruiz y dos pequeños Robert Alejandro Ruiz y Oddete Ruiz, gracias por entenderme en todo, gracias porque en todo momento son un apoyo incondicional en mi vida, son la felicidad encajada en una sola persona, fue mi todo reflejado en otra persona a la cual yo amo demasiado, y por la cual estoy dispuesto a enfrentar todo y en todo momento.

18 de diciembre de 2017

Este mismo furor y pasión con la que describo el perfecto e incondicional apoyo de ustedes, fue el mismo con el que desarrollé cada parte y punto de esta tesis, y por esto mismo puedo afirmar y pronosticar su éxito y agrado para cada uno de sus lectores.

A la Universidad de Quintana Roo…

A la Universidad de Quintana Roo por haberme aceptado ser parte de ella y abierto las puertas de su seno científico para podrá estudiar mi carrera, así como también a los diferentes docentes que brindaron sus conocimientos y su apoyo para seguir adelante cada día.

A Conacyt…

Por haberme otorgado la beca del programa asistente de investigador durante el periodo en que este trabajo de tesis se estaba realizando.

A mis Asesores…

A mis asesores el Dr. Homero Toral Cruz y el Dr. Pablo Velarde Alvarado, por haberme brindado la oportunidad de recurrir a su capacidad y conocimiento científico, así como también haberme tenido toda la paciencia del mundo para guiarme durante todo el desarrollo de la tesis.

A mis amigos…

Amigos y personas especiales en mi vida, no son nada más y nada menos que un solo conjunto de seres queridos que suponen benefactores de importancia inimaginable en mis circunstancias de humano. No podría sentirme más ameno con la confianza puesta sobre mi persona, especialmente cuando he contado con su mejor apoyo desde que siquiera tengo memoria.

Al Movimiento Antorchista….

Por haber sido pieza fundamental durante mi crecimiento en la carrera y dentro de las labores de la organización.

18 de diciembre de 2017

RESUMEN

Las redes de gran escala se enfrentan diariamente a miles de ataques de red. No importa la fuerza de los mecanismos de defensa de seguridad existentes, estas redes siguen siendo vulnerables, nuevas herramientas y técnicas están siendo desarrolladas constantemente por los piratas informáticos. Una nueva tecnología prometedora que atrae a los atacantes con el fin de supervisar sus actividades maliciosas y divulgar sus intenciones está surgiendo con Honeypots virtuales.

En este trabajo de tesis se presenta, la realización de un estudio y análisis donde se establecen fases de prueba que explican el funcionamiento de cada herramienta honeypot utilizada, llevando a cabo un análisis de los resultados extraídos mediante una serie de scripts implementados en Python, con el propósito de realizar un análisis de la red. Así mismo se examina un extenso mecanismo de seguridad basado en tres tipos diferentes de honeypots de baja interacción, de código abierto. Estos sistemas honeypot, identifican las amenazas potenciales y los métodos utilizados en contra de la red. De esta manera mediante herramientas de visualización se observarán los posibles ataques y medios por los cuales se identificaron las partes vulnerables. Se utilizarán herramientas para la intrusión de tráfico de red. Así como pruebas de testeo para verificar la eficiencia del sistema honeypot implementado.

Los datos recopilados por los honeypots revelan información valiosa sobre los tipos de ataques, los servicios de red vulnerables dentro de la red y las actividades maliciosas lanzadas por los atacantes.

PALABRAS CLAVE

Honeypot, honeyd, kippo, dionaea, Seguridad informática, Taxonomía honeypot, Honeyd-viz, Kippo-Graph, DionaeaFR, SQL, Server-Side, Client-Side.

Página intencionalmente en blanco

PROYECTO HONEYPOT

ÍNDICE DE TABLAS

Página intencionalmente en blanco

ÍNDICE DE FIGURAS

CAPÍTULO 1. INTRODUCCIÓN

"Conoce a tu enemigo y conócete a ti mismo, y saldrás triunfador en mil batallas."

Sun Tzu

La tecnología avanza día con día, logrando con esto beneficios sociales, ejemplo de esto es el rápido desarrollo de las empresas, la educación y la investigación científica y tecnológica; sin embargo, este progreso no solo beneficia a la sociedad, hay que reconocer la existencia de las amenazas y prueba de este hecho son los constantes ataques informáticos y la posibilidad de ocultar operaciones maliciosas.

La información en el ámbito social y empresarial es muy importante, es uno de los factores que impulsan el desarrollo de organizaciones e instituciones para la toma de decisiones. Hoy en la era de la información, la tecnología ha permitido adelantos significativos que están introduciéndose en las empresas con cambios imprescindibles, esto con la finalidad de que las organizaciones se adapten a las necesidades de cambio permanente, aprovechando su ventaja competitiva y respondiendo a la necesidad de ser ágil. Al referirse al aspecto de la seguridad existen investigaciones y desarrollos que en la actualidad son de ayuda para la protección de los datos, pero ¿Qué ocurre cuando los métodos de seguridad han sido violados? ¿Cómo detectar las evidencias de ataque?

La presente tesis es un documento de investigación sobre las tecnologías informáticas de seguridad en redes e investigación más innovadoras actualmente, hoy día los honeypot permiten obtener y analizar diferentes tipos de amenazas informáticas tales como troyanos y virus por mencionar ejemplos, con el uso de las tecnologías, las necesidades de comunicación y procesamiento de datos están en constante evolución, esto origina nuevos retos también a la seguridad informática.

Las políticas de seguridad en redes son creadas por empresas y organizaciones gubernamentales para proveer un marco para que los empleados sigan en su trabajo diario.

Todas las prácticas de seguridad en redes están relacionadas por la política de seguridad en redes.

Tal como la seguridad en redes está compuesta de dominios, los ataques a las redes son clasificados para hacer más fácil el aprender de ellos. Los virus, los gusanos y los troyanos son tipos específicos de ataques a las redes [1].

1.1 DESCRIPCIÓN DE LA PROBLEMÁTICA

El uso de máquinas de señuelo para dirigir la atención de los intrusos lejos de las máquinas bajo protección es una técnica importante para evitar ataques o intrusiones. Cualquier dispositivo, sistema, directorio o archivo utilizado como señuelo para atraer a los atacantes lejos de activos importantes y para recolectar comportamientos de intrusión se conoce como un honeypot. Los sistemas de monitoreo de tráfico son capaces de acceder a todo el tráfico que cruza la zona de demarcación y tratan de identificar patrones que puedan indicar un ataque. Los honeypots, son sistemas que se configuran para que parezcan posibles blancos de ataque Dado que los usuarios autorizados de la red saben que no deben usar el honeypot, cualquiera que intente acceder a este host es, por definición, un intruso.

La importancia de los honeypot radica en lo siguiente:

- Permiten recopilar información sobre quién está tratando de comprometer el sistema. Esto es posible ya que el honeypot tiene herramientas que pueden rastrear la fuente y los destinos.
- Pueden proporcionar la información sobre qué herramientas y tácticas han sido utilizadas por el atacante para comprometer el sistema. Tal información se puede encontrar en las técnicas que se han utilizado dentro de un honeypot tales como registros del firewall, sistemas de la detección de la intrusión (IDSs), y registros del sistema. Al obtener esta información, se puede evitar ataques en el futuro, mejorando el sistema contra estos ataques conocidos. La recopilación de información sobre herramientas y tácticas se considera como el objetivo más importante de un honeypot.

1.2 JUSTIFICACIÓN

En la actualidad, las organizaciones son cada vez más dependientes de sus redes informáticas y un problema que las afecte, por mínimo que sea, puede llegar a comprometer la continuidad de las operaciones. Dado que la información ha sido desde siempre un bien invaluable, protegerla ha sido una tarea continua y de vital importancia.

A medida que se crean nuevas técnicas para la transmisión de la información, se idean otras que permitan acceder a ella sin autorización. La seguridad no es solo una aplicación de un nuevo programa capaz de proteger, es más bien un cambio de conducta y de pensar. Hay que adueñarse del concepto seguridad e incluso volverse algo paranoico para que en cada labor que se desempeñe, se piense en seguridad y en como incrementarla. La falta de medidas de seguridad en las redes es un problema que está en crecimiento, cada vez es mayor el número de atacantes y cada vez están más organizados, por lo que van adquiriendo día a día habilidades más especializadas que les permiten obtener mayores beneficios. Por tal motivo, es imperante la necesidad de utilizar técnicas proactivas de defensa orientadas a la seguridad de redes como son los honeypots.

La idea de la tecnología de los honeypots es mostrar a los atacantes un sistema virtual que parezca el sistema real, cuya intención es atraer a los atacantes simulando ser sistemas débiles o con fallas de seguridad, evidentes, o no del todo, pero si lo suficiente para atraerlos y ser un reto para sus habilidades de intrusión, de esa manera los ataques se efectúan sobre ese sistema sin causar ningún daño al sistema real.

Los honeypots son herramientas que permiten atraer y analizar a los atacantes que desean ingresar a la red sin autorización, con la finalidad de ver los movimientos de los atacantes y saber que debilidades tiene la red de la institución.

A través de este proyecto de tesis, se espera como resultado, un análisis comparativo en el desempeño entre honeypots, los cuales se implementarán y configurarán. El resultado obtenido del dicho análisis, servirá de referencia para que cualquier usuario con conocimientos básicos en seguridad pueda seleccionar alguno de ellos en función de sus recursos de hardware y de acuerdo a las necesidades de desempeño sirva como guia para la implementación y configuración de un honeypot.

1.3 OBJETIVO GENERAL

- Evaluar, configurar e implementar distintas soluciones de honeypot sobre una plataforma Linux en un entorno experimental bajo distintos escenarios de riesgo.

1.4 OBJETIVOS ESPECÍFICOS

- Realizar una revisión bibliográfica para construir el estado del arte de honeypot.
- Evaluar, instalar y configurar tres distribuciones de honeypot *OpenSource*, en un entorno de cama de pruebas.
- Crear los escenarios de ataque a los que estará expuesto el honeypot.
- Realizar ataques mediante herramientas de software libre para comprobar la funcionalidad y desempeño de los honeypots seleccionados.
- Implementar funciones de captura de datos, análisis de bitácoras y mantenimiento de las herramientas honeypot.

1.5 ORGANIZACIÓN DE LA TESIS

El resto de este documento de tesis está organizado de la siguiente manera:

En el capítulo dos se explica una breve introducción sobre los honeypots así como conceptos teóricos necesarios para el desarrollo de esta investigación, como son: la tecnología honeypot; historia y definición; taxonomía de sistemas básicos de honeypot; sistemas perimetrales existentes en la actualizada; información acerca de la ubicación de los honeypot en la red para mostrar los alcances de este trabajo.

En el capítulo tres se aborda la descripción de las diferentes distribuciones de los sistemas honeypots de baja interacción, como lo son Honeyd, Kippo y Dionaea; en cada uno de ellos se presenta la información necesaria para su instalación y configuración en un entorno experimental. De la misma manera se muestra la arquitectura en cada uno de los sistemas seguido de ello permite realizar una captura de datos, la cual proporciona la información recopilada en diferentes logs y hacer mención a un sistema de alertas y si dicho honeypot cuenta con ello. Además, se estudia el manejo de bitácoras el cual permite ver la información relativa a los intentos de conexión con cada sistema.

En el capítulo cuatro se presenta el escenario de estudio y herramientas de visualización que se utilizarán, así como la captura de información de cada uno de los sistemas mencionados en el capítulo tres para su posterior análisis. Se mencionan los sistemas de visualización como lo son: Honeyd-Viz, Kippo-Graph y DionaeaFR, para su posterior instalación en un entorno experimental. Y finalmente se explica la forma en la que estos sistemas permitirán revisar bitácoras SQL para su análisis acerca de ataques a estos sistemas honeypots.

En el capítulo cinco se muestra la implementación de los sistemas honeypots, con una instalación a detalle de cada uno ellos, así como la instalación de herramientas de visualización, para su posterior ejecución de cada uno de ellos en un entorno virtualizado.

En el capítulo seis se presenta el análisis y los resultados obtenidos de la implementación de los honeypots mencionados en el capítulo anterior, así como figuras, tablas y gráficas para el mejor entendimiento de los resultados.

Por último, en el capítulo siete se presentan las conclusiones y trabajo futuro generado por este proyecto de tesis.

Página intencionalmente en blanco

CAPÍTULO 2. TECNOLOGÍA HONEYPOT

"Las grandes obras no son hechas con la fuerza, sino con la perseverancia."

Samuel Johnson

En todo trabajo de investigación es necesario establecer los conceptos básicos, complementarios y específicos del tema que se trate, así como herramientas utilizadas, ya que estos conceptos ayudan a entender la problemática que se plantea. Es por ello que en este capítulo se abordan temas como historia y definición de honeypot, taxonomía básica y sistemas de defensa experimental.

2.1 INTRODUCCIÓN

La seguridad de las redes es ahora una parte integral de las redes informáticas. Incluye protocolos, tecnologías, dispositivos, herramientas y técnicas que aseguran los datos y reducen las amenazas. Las soluciones de seguridad en redes surgieron en los años 1960 pero no se convirtieron en un conjunto exhaustivo de soluciones para redes modernas hasta el principio del nuevo milenio.

Se han creado organizaciones de seguridad en redes para establecer comunidades formales de profesionales de la seguridad en redes. Estas organizaciones establecen estándares, fomentan la colaboración y proveen oportunidades de desarrollo para los profesionales de la seguridad.

En los últimos años la tecnología ha tenido un crecimiento desproporcionado y dicha tecnología que ha sido enfocada en la seguridad de las conexiones de Internet, datos y servicios informáticos de las compañías, ha cambiado de alguna manera u otra, el diario vivir de la mayoría de las personas en el mundo, los nuevos modelos, técnicas y herramientas enfocadas en la seguridad de las comunicaciones para contrarrestar la comunidad black hat están generando cambios en la cotidianidad de las personas, puesto que los servicios y aplicaciones que necesitan de un grado de confiabilidad están

mejorando visiblemente y con esto el acceso a actividades que anteriormente solo se realizaban personalmente por procesos y aplicaciones en el Ciber espacio.

En el arte de la guerra la mejor arma es la información, cuanto mejor conozcas a tu enemigo, habrá mejores opciones y mayor posibilidad de vencerle, esta es la frase insignia de un conjunto de investigadores "The Honey Project" que surgió con el propósito de recopilar información sobre los ataques y los atacantes que tienen como medio el Internet para llevar a cabo sus propósitos maliciosos.

Según un informe realizado por el Computer Security Institute en el año 2006 en Latinoamérica existe acerca de 90 millones de internautas y sin embargo la mayoría no tiene conocimiento acerca de las vulnerabilidades y riesgos con que cuentan sus equipos y mucho menos están en conocimiento de la comunidad Black hat y se incrementó en número de ataques y se perdieron cerca de quince millones de dólares por inyección de virus sin considerar pérdidas causadas por otros ataques [2].

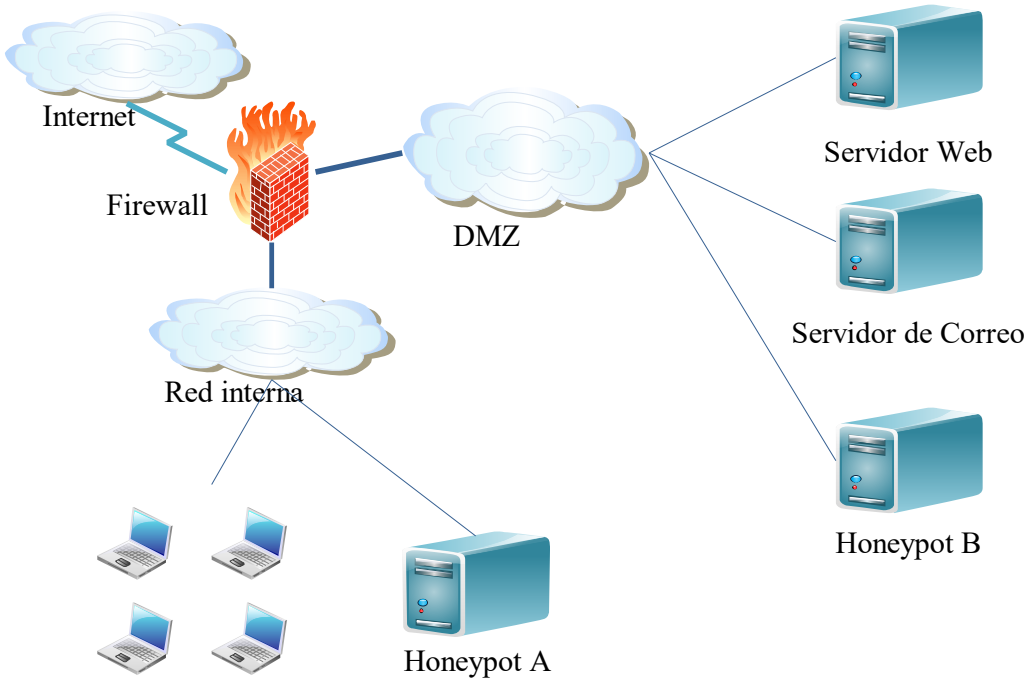


Figura 2.1 Arquitectura de los honeypots en una red.

2.2 HISTORIA Y DEFINICIÓN DE LOS HONEYPOTS

Para definir honeypot se utilizará la definición realizada por [3] como:

> *Un recurso de seguridad destinado para ser atacado o comprometido por atacantes. Su finalidad no es ningún caso, resolver o arreglar fallas de seguridad. Sino que se encarga de proporcionar información valiosa sobre posibles atacantes en la red antes de que comprometan sistemas reales.*

A continuación, se enumeran algunas de las muchas referencias históricas:

- 1990/1991 – Clifford Stolls the Cuckoos Egg and Bill Cheswicks: La primera publicación referente al concepto de honeypot.
- 1997 – Versión 0.1 de Fres Cohen of Deception Toolkit: La cual fue lanzada como una de las primeras soluciones honeypot dentro de la comunidad de la seguridad.
- 1998 – Cybercop Sting: Se comenzó el desarrollo de uno de los primeros honeypots comerciales que fueron vendidos al público. Cybercop Sting introduce el concepto de múltiples sistemas virtuales concentrados en un solo honeypot.
- Marty Roesch y GTE Interworking: Se comenzó el desarrollo de una implementación honeypot que se convirtió en NETFACADE, el cual originó el concepto Snort.
- 1999 – Honeynet Project: Se formó el proyecto Honeynet, así como una serie de importantes publicaciones “Know your Enemy” este proyecto y publicaciones fueron las que hicieron crecer el valor e importancia de esta tecnología. Este proyecto se inició informalmente en la lista de correo “Wargames” gracias a los correos cruzados entre varios expertos en seguridad informática, que terminaron con el desarrollo formal del proyecto antes de finalizar dicho año.
- 2000 – El honeypot del proyecto fue atacado y comprometido por un famoso grupo de Hackers, lo que permitió el estudio y análisis del comportamiento de este grupo, así como demostrar la viabilidad y utilidad de esta nueva herramienta. Este incidente elevo el concepto honeypot como la última tendencia en seguridad de redes, llegando a convertir su libro en un best-seller en la seguridad informática.

- Comienzos del 2001 – El proyecto se convirtió en una organización sin ánimo de lucro dedicada al estudio de los hackers, que actualmente está compuesta por más de treinta miembros permanentes.
- 2000/2001 – El uso de honeypots para la captura de información con el objetivo de estudiar la actividad de gusanos informáticos. Muchas organizaciones adoptaron los honeypots como medio de investigación y detección de ataques informáticos.
- 2002 – Los honeypots son usados para detectar y capturar información sobre ataques desconocidos.

2.3 TAXONOMÍA BÁSICA DE LOS HONEYPOTS

Los honeypots se pueden clasificar en base a dos criterios fundamentales e independientes, tipo de recursos atacados y nivel de interacción. Esta taxonomía es muy básica y se ajusta a todas las taxonomías de honeypot más complejas.

Primer Criterio - Tipo de recursos atacados - Describe si los recursos de un honeypot se explotan en modo servidor o cliente. Un honeypot del lado del servidor utiliza servicios de red como SSH o NetBIOS, escuchando en sus puertos y supervisando cualquier conexión iniciada por clientes remotos. Por el contrario, un honeypot del lado del cliente empleará un conjunto de aplicaciones cliente, como un navegador web, que se conectan a servicios remotos y supervisan toda la actividad generada.

El segundo criterio - Nivel de interacción - Determina si el honeypot es un recurso real de alta interacción o sólo una emulación baja interacción. Un tipo mixto de honeypot que combina ambas funcionalidades se llama honeypot híbrido. [4]

2.3.1 Server-Side honeypots

Honeypots diseñados para detectar y estudiar los ataques a los servicios de red se denominan Server-Side. Los honeypots de este tipo actúan como un servidor, exponen un puerto abierto, múltiples puertos o aplicaciones enteras y escuchan pasivamente las conexiones entrantes, establecidas por clientes remotos probablemente maliciosos, en la siguiente figura se puede ver un honeypot en modo servidor.

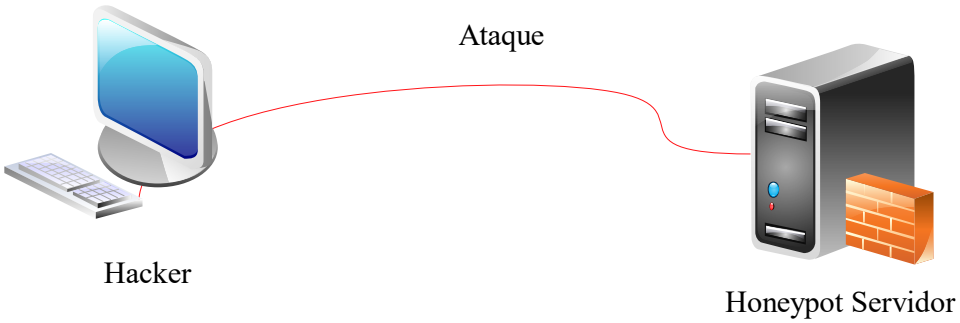


Figura 2.2 Honeypots en modo servidor.

A menudo, estos tipos de honeypots detectan amenazas que utilizan el escaneado como medio para identificar a las víctimas potenciales, como por ejemplo el escaneo de gusanos o bots. Los honeypots del lado del servidor se consideran los honeypots tradicionales, y a menudo el término honeypots se asocia por defecto. [4]

2.3.2 Client-Side honeypots

Honeypots diseñados para detectar ataques a aplicaciones son llamados Client-Side. Una aplicación cliente es una pieza de software que establece una conexión a un servidor e interactúa con ella. El tipo más popular y el más específico de las aplicaciones del lado del cliente son los navegadores web, junto con las extensiones y complementos asociados, en la siguiente figura se muestra un honeypot en modo cliente.

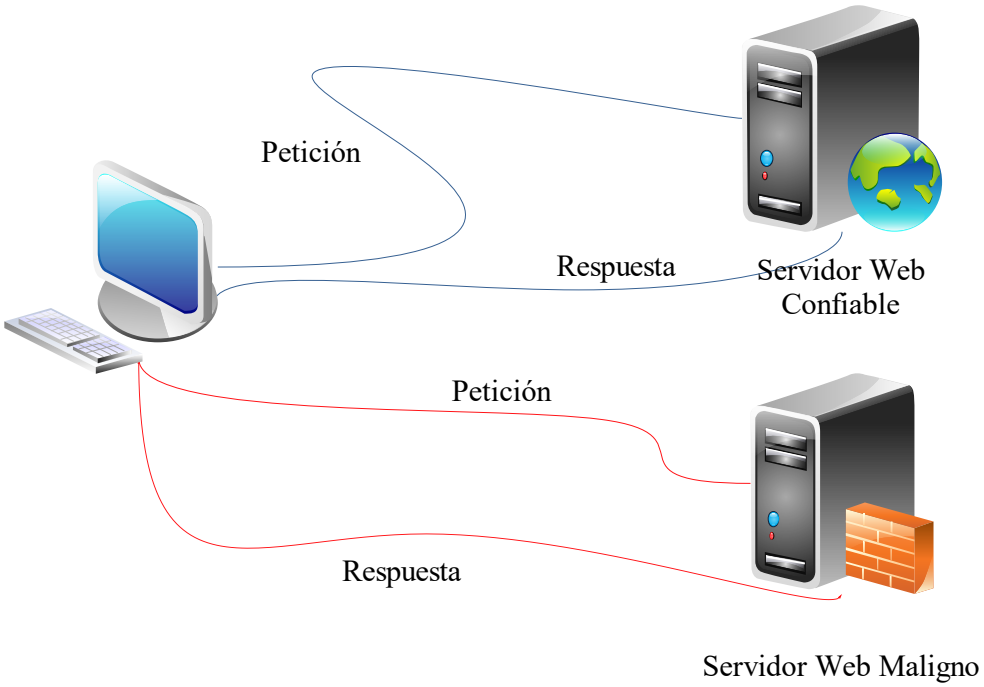


Figura 2.3 Honeypots en modo cliente.

Los honeypots del lado del cliente son muy diferentes en su funcionamiento de servidores. Honeyclients establece activamente conexiones a servicios para detectar comportamientos maliciosos del servidor o del contenido que sirve.

Los honeyclients más populares son aquellos que detectan ataques a navegadores web y sus complementos, propagados a través de páginas web. También tienen la capacidad de mirar varias formas de archivos adjuntos, y ha habido intentos de crear mensajes instantáneos honeypots también. [4]

2.3.3 Honeypots de baja interacción

Los honeypots de baja interacción normalmente emulan servicios y sistemas operativos. La actividad del atacante se encuentra limitada al nivel de emulación de dicho honeypot. La ventaja de un honeypot de baja interacción radica en su simplicidad, ya que estos tienden a ser fáciles de utilizar y mantener con un riesgo prácticamente nulo.

Por lo general, el proceso de implementación de un honeypot de baja interacción, se basa en una instalación de un software de emulación de sistema operativo, utilizando herramientas como *Virtual box* o *VMWare*, elegir el sistema operativo y el servicio a emular. Este proceso, de naturaleza similar al *plug and play*, hace que la utilización de este tipo de honeypot sea sencilla. Los servicios emulados mitigan el riesgo de penetración en el sistema, conteniendo la actividad del atacante que nunca tiene acceso al sistema operativo real donde podría atacar o dañar otros sistemas.

La principal desventaja de este tipo de honeypots, radica en que registran únicamente una información limitada, y aunque están diseñadas para capturar una actividad predeterminada. Por lo tanto, es relativamente sencillo para un atacante detectar un honeypot de baja interacción, ya que un intruso hábil puede detectar que se trata de una emulación con el paso del tiempo [5].

2.3.4 Honeypots de alta interacción

Los honeypot de alta interacción constituyen a una solución mucho más compleja, son más difíciles de implementar y mantener, porque los sistemas y servicios que brinda no son emulados, son reales montados sobre sistemas operativos y hardware lo que aumenta el riesgo de uso. De igual manera, estos sistemas también pueden quedar sin parches después de la instalación para permitir al atacante explotar las vulnerabilidades conocidas al atacar el honeypot. [6]

Son capaces de detectar ataques, permiten al atacante interactuar con todas las capas del modelo OSI e incluso permitirle entrar en el sistema. Esto permite capturar las pulsaciones de teclas de los atacantes, rootkits, herramientas y patrones de ataque. Esta información se puede utilizar para comprender los motivos, niveles de habilidad y otros detalles de los atacantes. Como el atacante está interactuando con un sistema real también será capaz de registrar un comportamiento nuevo, inesperado o desconocido. Sin embargo, los sistemas reales pueden utilizarse como una plataforma para que un atacante lance nuevos ataques contra sistemas no honeypot (dentro o fuera de la organización), lo que introduce un cierto riesgo al desplegar un honeypot. Además, son más complejos que los honeypots de interacción baja ya que necesitan ser construidos y configurados para su

tarea. Con esta creciente complejidad, también hay mayores requisitos de mantenimiento [7].

Debido al alto nivel de libertad que un atacante tendrá en el honeypot puede estar completamente comprometido, por lo tanto, debe ser monitoreado y observado para detectar cualquier acción y cambios hechos al sistema. Cuando el honeypot ha sido comprometido puede tomar horas o incluso días para analizar los eventos del ataque. Debido a esta complejidad y alto mantenimiento hace que sea difícil desplegar honeypots de alta interacción a gran escala. [8]

2.3.5 Honeypots híbridos

Los honeypots híbridos combinan tanto las herramientas de baja interacción como las de alta interacción para obtener los beneficios de ambas. En SGNET, se utiliza un honeypot del lado del servidor de alta interacción para aprender a manejar tráfico desconocido. Cómo emular nuevos protocolos. Después de este proceso de aprendizaje, el tráfico similar adicional se redirige a la baja interacción del lado del servidor. Esta combinación aumenta tanto el nivel de detección de amenazas como el rendimiento.

SurfCERT IDS utiliza múltiples honeypots de servidor de baja interacción y Argos, una solución de alta interacción. Del mismo modo en HoneySpider Network un honeyclient de baja interacción filtra los sitios web, mientras que todos los demás son analizados de nuevo - esta vez con honeyclients de alta interacción. [4]

2.3.6 Honeypots virtuales

Los honeypots virtuales nacen de la necesidad de tener un gran espacio de direcciones IP, es casi imposible implementar un honeypot por cada IP por razones en espacio físico y económico. En una máquina física, se puede levantar varios honeypots como máquinas virtuales, los honeypots no constituyen una máquina real, pero pueden proporcionar todo el nivel de interacción como un honeypot Físico de Alta Interacción, la única diferencia es que está corriendo bajo algún software de virtualización y comparten los recursos físicos de la máquina real, inclusive la conexión a Internet, permitiendo tener conectadas

a toda una red de honeypots con sus respectivas IPs dentro de una máquina física, facilitando la movilidad y reduciendo enormemente la cantidad de hardware usado.

El sentido básico en su despliegue es la escalabilidad y fácil cuidado. Uno puede desplegar más de cientos de honeypots sólo en una sola máquina física. El atractivo de los honeypots virtuales es que son muy razonables para desplegar y fácilmente disponibles en Internet para cualquiera que quiera tenerlo. [7]

2.4 SISTEMAS DE DEFENSA PERIMETRAL

A medida que Internet surgió y se convirtió en una importante plataforma comercial; la necesidad de una defensa perimetral entre la red corporativa y otras redes, en particular, la Internet pública, surgió el desafío de la defensa perimetral esto generó una sucesión de mecanismos de seguridad de red diseñados para restringir las rutas permitidas y la inspección del tráfico de red [9].

A continuación, se enmarcan cuatro sistemas de defensa perimetral y en cada uno de ellos se explica la forma en la que actúan sobre una red, como son:

- Honeywall
- Firewall
- NIDS
- DMZ

2.4.1 Honeywall

La tarea del Honeywall es separar el honeypot o honeynet del resto de la red para mitigar el riesgo de dañar los sistemas que no son honeypot. Como con un firewall, todo el tráfico del honeypot debe pasar a través del Honeywall tal como se muestra en la figura 2.1.

El Honeywall es importante cuando se implementan honeypots de alta interacción, o hay manera de controlar a un atacante en un sistema completamente comprometido. En cambio, el atacante debe ser controlado por el Honeywall que evita que se lancen nuevos ataques desde el honeypot. Sin embargo, una arquitectura de este tipo es muy difícil de

implementar, ya que bloquearla demasiado hará que el atacante sospeche y puede revelar que está siendo monitoreado, mientras que tenerlo abierto puede permitir al atacante lanzar ataques en otros sistemas [3].

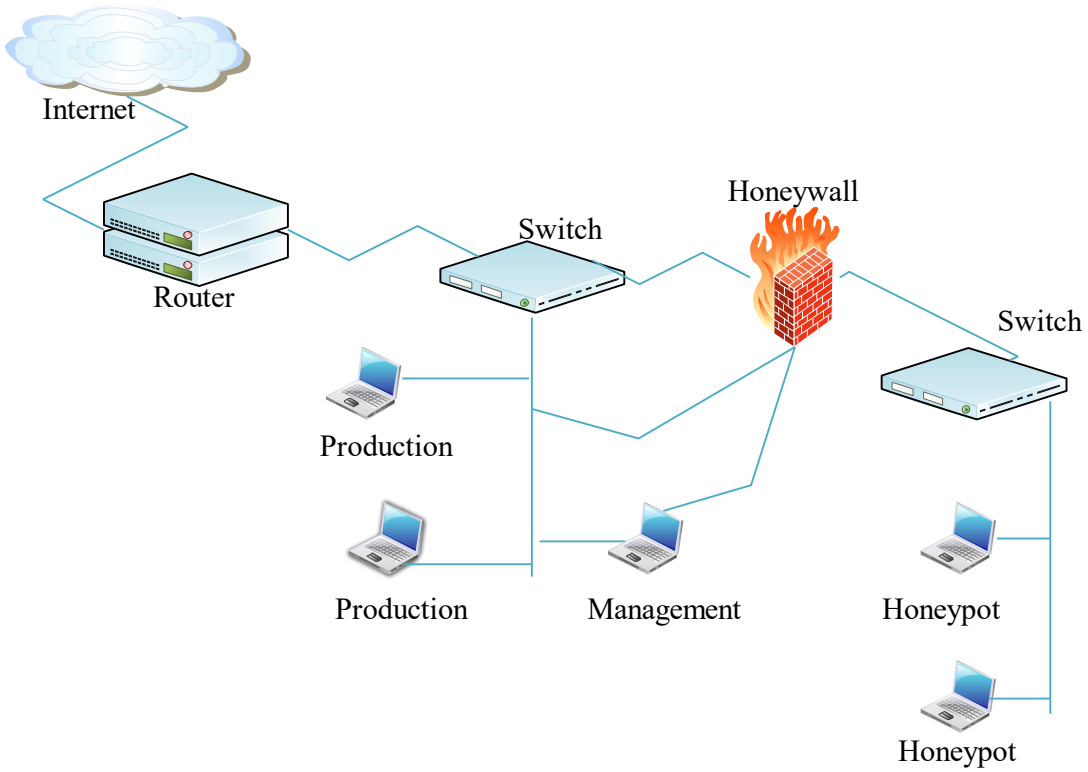


Figura 2.4 Representación gráfica del sistema perimetral de un honeypot.

2.4.2 Firewall

Un firewall es un sistema ubicado entre dos redes como lo muestra la figura 2.2, el cual permite o deniega el tráfico que circula entre ellas obedeciendo una política de seguridad implementada a través de una lista de control de acceso *Access Control List*. Es el mecanismo encargado de proteger una red confiable de una que no lo es, como Internet. El firewall solo sirve para defensa perimetral de las redes, no defiende de ataques o errores provenientes del interior, como tampoco ofrece protección una vez que el intruso lo traspasa. [10]

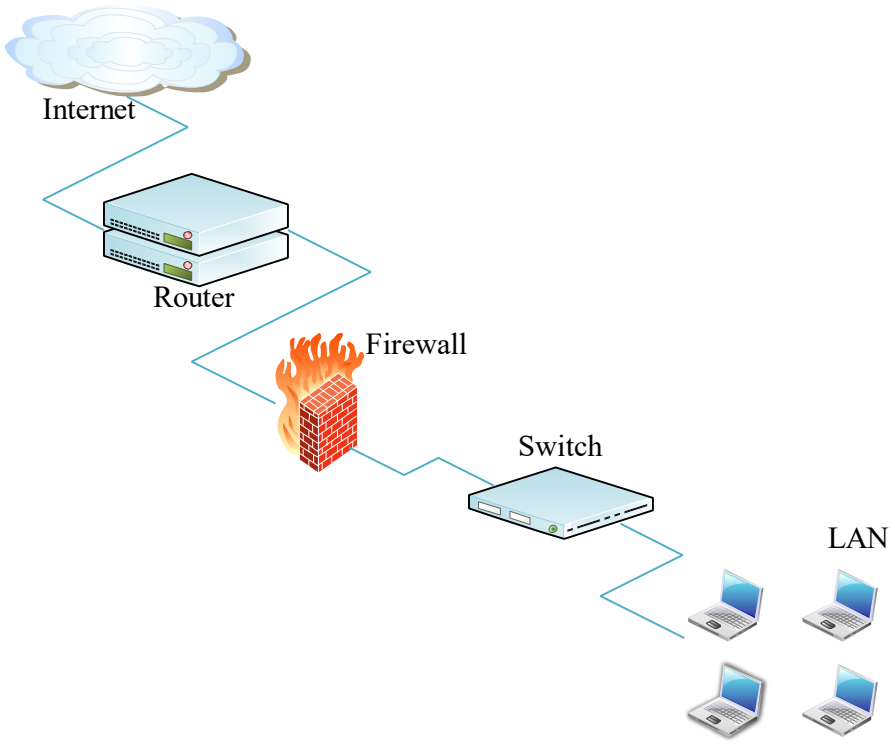


Figura 2.5 Representación gráfica del sistema perimetral firewall.

2.4.3 NIDS (Network Intrusion Detection System)

La función del sistema de detección de intrusiones de red es alertar cuando ocurre una actividad sospechosa con el perímetro de la red como lo muestra la figura 2.3. Actúan sobre una red capturando y analizando paquetes, buscando patrones que supongan algún tipo de ataque.

Cuando los mecanismos de seguridad de red funcionan correctamente, pueden analizar grandes redes y su impacto en el tráfico suele ser pequeño. [11].

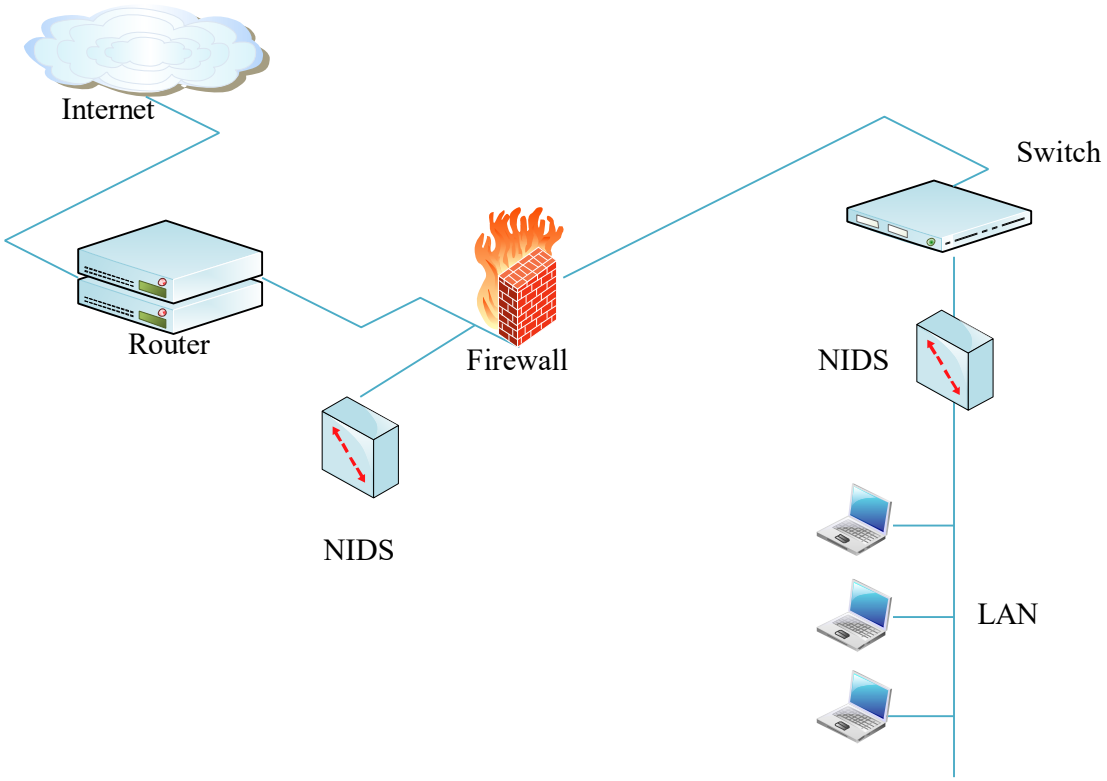


Figura 2.6 Representación gráfica del sistema perimetral NIDS.

2.4.5 DMZ (Demilitarized Zone)

La zona desmilitarizada es el punto medio entre la Internet no confiable y la LAN interna de confianza como lo muestra la figura 2.4. Es un punto en el servidor de seguridad donde se colocan los servidores necesarios para estar accesibles desde Internet. Los servidores típicos que se ponen en la DMZ son servidores de correo electrónico o web. El propósito es mantener a los servidores públicos completamente separados de los servidores privados en la LAN. [10]

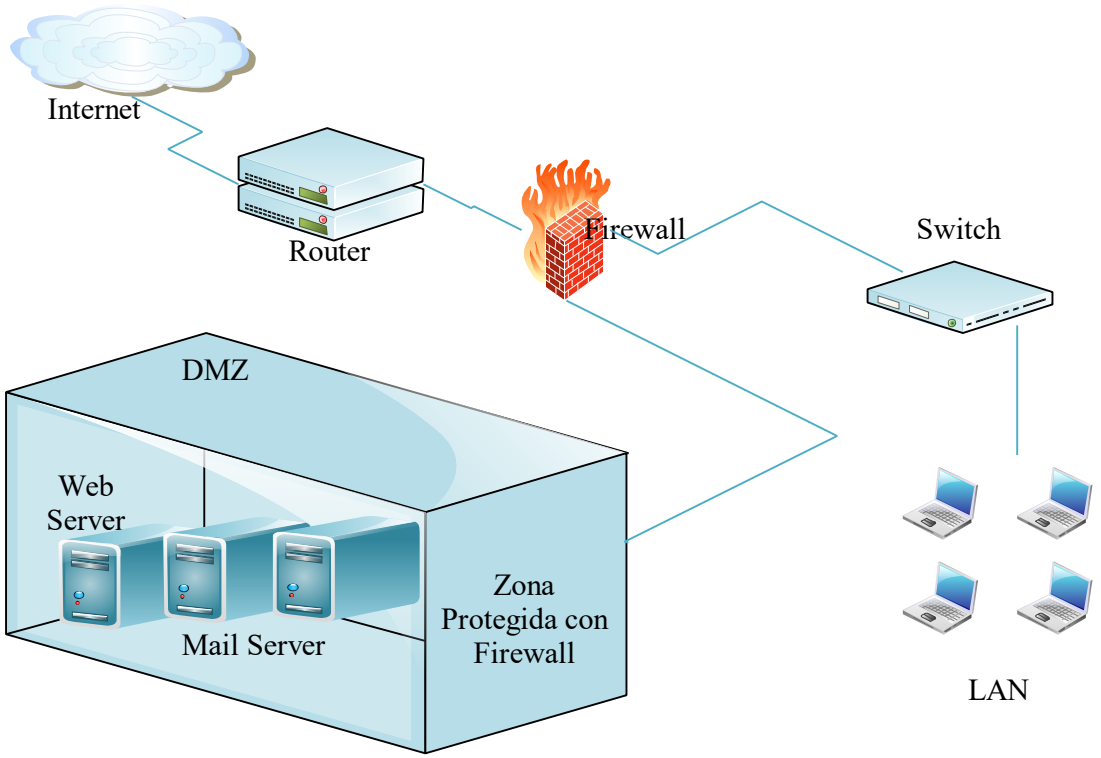


Figura 2.7 Representación gráfica del sistema perimetral DMZ.

2.5 UBICACIÓN DE LOS HONEYPOTS

Hay tres ubicaciones principales para colocar un sistema de honeypot, donde cada ubicación tiene sus ventajas y desventajas dependiendo del objetivo de la implementación de honeypot:

- Interno detrás del firewall
- Exteriores frente a Internet
- En la DMZ

2.5.1 Ubicación interna

El honeypot interno se coloca dentro de la red con un firewall entre él e Internet. La principal ventaja de colocarlo en la red interna es que puede exponer ataques que han hecho pasar las defensas de red, así como la captura de amenazas internas al mismo tiempo como lo muestra la figura 2.5. El honeypot interno podría, en este ejemplo,

advertir a los administradores del sistema que el gusano ha pasado por el firewall y está probando equipos internos.

Como el honeypot está protegido por el firewall, también recibirá escaneos y debido a esto recogerá mucho menos datos que un honeypot externo. Esto puede ser tanto una ventaja como una desventaja. Si el objetivo del honeypot es reunir tanta información como sea posible, es una clara desventaja. Sin embargo; La baja cantidad de datos recibidos también facilitará el trabajo de monitoreo y mantenimiento del honeypot. El principal inconveniente de los honeypots colocados internamente es la amenaza que representa si está completamente comprometida, ya que los ataques pueden lanzarse libremente en otros nodos internos. [6]

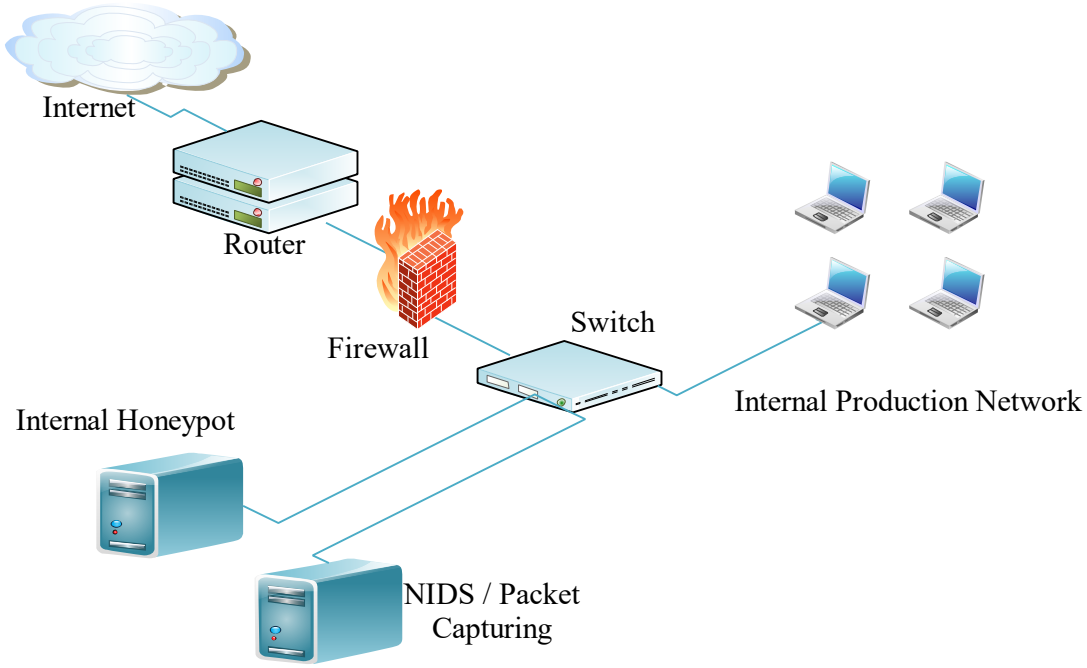


Figura 2.8 Internal honeypot.

2.5.2 Ubicación externa

Con colocación externa no hay firewall que proteja el honeypot de ninguna manera. Sin filtrar lo que llega al honeypot, estará libremente expuesto a ataques totalmente y aumentará el número de escaneos que recibe como lo muestra la figura 2.6.

Si el número de direcciones IP públicas es limitado, las unidades de supervisión y registro pueden colocarse en la misma LAN que el honeypot conectado a través de un network tap. Esto les permitirá supervisar cualquier tráfico que vaya desde y hacia el honeypot. Debido a la falta de un firewall o algún otro tipo de defensa, esta configuración plantea el mayor riesgo de que el honeypot comprometido se utilice como una plataforma para atacar la red de producción o redes externas. [6]

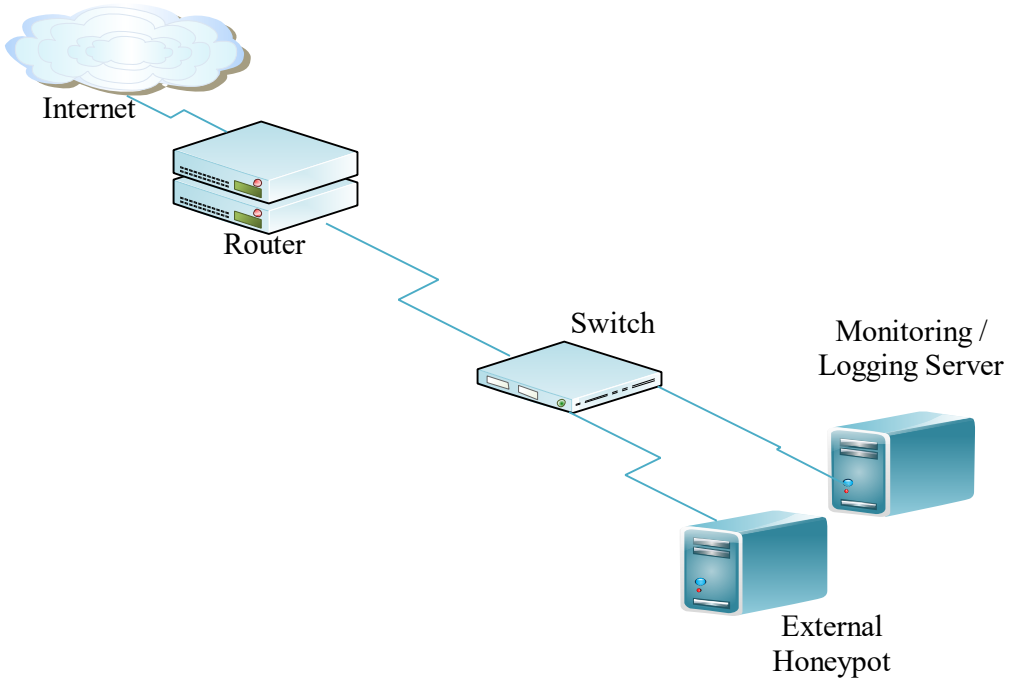


Figura 2.9 External honeypot.

2.5.3 Ubicación en DMZ

La tercera ubicación disponible para colocar un honeypot se encuentra en el firewall DMZ. Todos los nodos que están posicionados en la DMZ ya están expuestos a sondas e intentos de ataques desde el exterior como lo muestra la figura 2.7.

Tener un honeypot en la DMZ puede proporcionar advertencias tempranas de cualquier violación de seguridad en estos servidores expuestos. Colocar el honeypot en la DMZ protegerá la red interna contra ataques lanzados desde dentro del honeypot. Por otro lado, no será capaz de servir como un indicador para cualquier compromiso de red interna. [6]

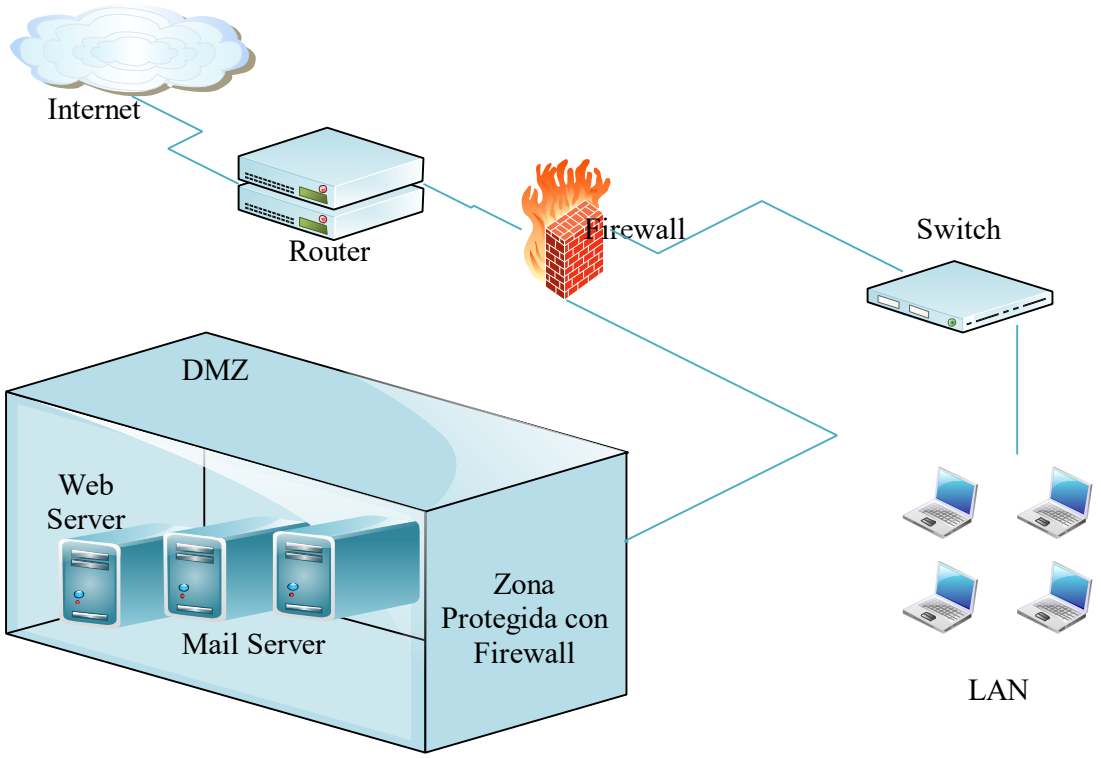


Figura 2.10 DMZ Honeypot.

2.5.4 Ventajas y desventajas de los honeypots

Las principales ventajas y características que ofrecen los honeypots son las siguientes:

- Generan un volumen de tráfico pequeño de datos, que al contrario que los demás sistemas de seguridad firewall, IDS, etc. Generan un gran volumen de datos,

incluso de información que no es necesaria, para el caso de los honeypots, estos generan una información con muy pocos datos, pero de alto valor.

- Los honeypots son equipos que ningún usuario o sistema normal debe acceder a ellos. Permitiendo de esta forma, revelar cualquier tipo de acceso, al atacante o una configuración errónea del sistema, sin llegar a tener prácticamente falsos positivos.
- Se necesitan recursos mínimos ya que, a diferencia de otro tipo de sistemas de seguridad, sus requisitos son mínimos. No consume ancho de banda, memoria o CPU. No necesita complejas arquitecturas o un gran número de equipos centralizados, cualquier equipo conectado a la red puede realizar el trabajo de un honeypot.
- Es un tipo de sistema que sirve tanto para atacantes internos como externos. De tal forma, se evita poner nombres a las maquina como honeypot o attack-me; muchas veces ni tan siquiera están dadas de alta en los servicios de DNS. Su objetivo es pasar de manera desapercibida en una red como una maquina más.

Por otro lado, como todo tipo de sistema tiene también una serie de contrapartidas o desventajas, las cuales son:

- Son elementos toralmente pasivos, de esta forma, si no reciben ningún ataque no sirven para gran cosa.
- Son fuentes potenciales de riesgo para la red. Debido a la atracción que ejercen sobre los atacantes, de tal manera que si no calibramos perfectamente el alcance de un honeypot y lo convertimos en un entorno controlado y cerrado, este puede ser utilizado como fuente para ataques a otras redes o incluso a la propia red. Una posible solución a esta desventaja es la de incorporar el honeypot en una red DMZ.
- Tienen una visión limitada, ya que solo ellos pueden rastrear y capturar actividad destinada a interactuar directamente con ellos.

2.5.5 Ventajas y desventajas según su ubicación

La siguiente tabla muestra las ventajas y desventajas de los honeypots de acuerdo a su ubicación [6].

Tabla 2.1 Ventajas y desventajas de los honeypot según su ubicación.

Ubicación	Ventajas	Desventajas
Externa	• Alta exposición en Internet. • Fácil de configurar. • Bajo número de dispositivos de red necesarios.	• Control de datos deficiente. • Mayor riesgo para la red de producción.
Interna	• Bueno para imitar activos de producción. • Lo mejor para monitorear empleados internos. • Sistema de alerta temprana para respaldar otras defensas.	• Configuración más compleja • Control de datos cuestionables. • Necesidad de decir que puertos permitir / redirigir.
DMZ	• Bueno para imitar activos en producción. • Posibilidad de control de datos.	• Configuración más compleja. • Necesidad de decir que puertos permitir / redirigir. • No es el sistema de alerta interna más sólida.

2.6 Tecnología anti-honeypot

Los Honeypots se utilizan para engañar a los atacantes y mejorar la seguridad dentro de las grandes redes de equipos. A medida que esta actividad creciente se convierte en una nueva tendencia, los atacantes siempre intentan derrotar la eficacia del honeypot. Los honeypots son muy eficaces y los atacantes están trabajando para encontrar formas de explotarlos y evitarlos [2].

Problemas de la Red.

Es posible que los atacantes no quieran atacar a una computadora que está siendo utilizada para atraparlos, y tal vez no quieran ser monitoreados porque esto podría revelar su identidad, sus métodos y sus herramientas.

Por ejemplo, usando una explotación de *zero day* contra un honeypot que registra todo; un atacante probablemente perdería el secreto valorado de sus técnicas. Así, un atacante tratará de encontrar el sistema de víctimas como un honeypot o sistema real. Incluso si los atacantes tienen acceso a un honeypot, por ejemplo, a través de una Shell o a través de algún Shellcodes personalizado, que todavía puede utilizar la capa de red con el fin de determinar si han comprometido un honeypot en lugar de una máquina real.

Al hacer esto, podrían revelar sus técnicas utilizadas para imprimir un honeypot a través de la capa de red. Para entonces, los defensores que operan el honeypot ya tendrán un registro de la actividad maliciosa del atacante como una especie de alarma antirrobo [12].

Técnicas para la detección de Honeypots

La aparición de este sistema de detección de honeypot, en asociación con otras herramientas de spam emergentes, sugiere dos tendencias importantes:

- Honeypots están afectando a los spammers.
- La tecnología honeypot actual es detectable, y es probable que haya más sistemas de identificación de honeypot.

La capacidad de detectar un honeypot es poco probable que permanezca limitada a los spammers. Otros grupos hostiles o maliciosos podrían beneficiarse de sistemas de identificación. En un esfuerzo por crear sistemas honeypot indetectables, se crea una mejora significativa en las tecnologías honeypot de hoy en día.

Para aparecer como un objetivo tentador, los honeypots ofrecen una variedad de servicios aparentemente vulnerables. A pesar de la complejidad de los servicios honeypot estos varían drásticamente, por lo general caen en una de cuatro tipos: mínimo, restringidas, simulado y completo. De baja y alta complejidad.

- Los servidores mínimos proporcionan un puerto de servicio abierto.
- Los servidores restringidos proporcionan interacciones básicas.
- Los servidores simulados proporcionan interacciones complejas.
- Los servidores completos proporcionan soporte funcional completo.

Niels Provos y sus colegas tienen una web aparece como un servidor de trabajo completo, pero en realidad, registra las acciones en lugar de realizar operaciones externas. Servidores simulados aceptan INS de registro y solicitudes, y generan respuestas conocidas y mensajes de error. Los ejemplos de servidores simulados incluyen secuencias de comandos que emulan completa SMTP y los servidores web IIS de Microsoft.

A diferencia de estos pseudoservicios, los servicios de honeypot son raros. No sólo gestionan las solicitudes, sino que también permiten que las entidades malintencionadas interactúen totalmente e incluso comprometan el sistema simulado. Muchos honeypots completos también permiten conexiones externas limitadas, lo que hace que el servicio aparezca completamente funcional al mismo tiempo que evita que participe en DoS [13].

Cuestiones relacionadas con el sistema.

A medida que los honeypots se están desplegando, los atacantes han comenzado a diseñar técnicas para detectar, eludir y desactivar los mecanismos de registro utilizados en los honeypots. Se presentarán varias técnicas y algunas herramientas diversas que ayudarán a los atacantes a descubrir e interactuar con honeypots. La sección tiene como objetivo mostrar a los equipos de seguridad y los profesionales que quieren configurar o endurecer

sus propias líneas de defensa basada en el engaño de lo que la limitación de la investigación basada en honeypot es actualmente [14].

Es un problema difícil para desplegar honeypots que no pueden ser detectados por los hackers. La tecnología honeypot sólo es efectiva si un atacante no sabe que está atacando una "trampa" en lugar de un sistema real. Por lo tanto, es críticamente importante para los profesionales de la seguridad que despliegan honeypots para ser conscientes de los métodos que los hackers blackhat utilizan para identificarlos.

El ataque a un honeypot se puede llevar a cabo cualquier nivel como se discute en problemas de red y problemas del sistema. Hay muchas técnicas presentes para la impresión de la presencia de honeypot en la red. Varias técnicas para la detección de honeypot también se discute junto con alguno de los cazadores de honeypot que muestra potencialmente todos los tipos de honeypot pueden ser detectados. Por último, se proponen varias obras para las contramedidas para la detección de honeypot.

Así, tanto el atacante como el defensor del honeypot ya están listos con sus armaduras y armas y en los últimos tiempos puede haber herramientas y técnicas más potentes de Internet que pueden ser inventadas en ambos lados [15].

CAPÍTULO 3. DESCRIPCIÓN DE LAS DISTRIBUCIONES DE HONEYPOTS

"El coraje es una disposición a sentir grados pertinentes de temor y confianza en situaciones desafiantes."

Aristóteles

3.1 HONEYD

Honeyd es un honeypot de propósito general. La característica principal que lo diferencia de la mayoría de los honeypots es que es capaz de emular simultáneamente multitud de sistemas operativos, cada uno con sus propios servicios y asociando una dirección IP a cada uno de ellos a través de una única interfaz de red. Para la asignación IP se ayuda de una aplicación que realiza spoofing de un rango de red determinado, *Farpd*.

Otra de las características importantes de este honeypot, es la posibilidad de distraer a los atacantes, que estos ven todas las vulnerabilidades de este honeypot y realizan sus ataques hacia este, olvidando a los verdaderos sistemas desplegados en la red, por lo tanto, también sirve para mejorar la seguridad de una red.

Este honeypot ha sido elegido para su implementación, ya que es importante dotar este sistema de un honeypot de tipo genérico, es decir que no detecte un solo tipo de ataque, sino que detecte un gran número de ataques. Hay que decir, que Honeyd es una herramienta que solo detecta movimiento en los puertos habilitados en los hosts emulados, por lo tanto, resulta esencial la creación de un sistema de clasificación de estos ataques detectados, ayudando a la detección de este honeypot [8].

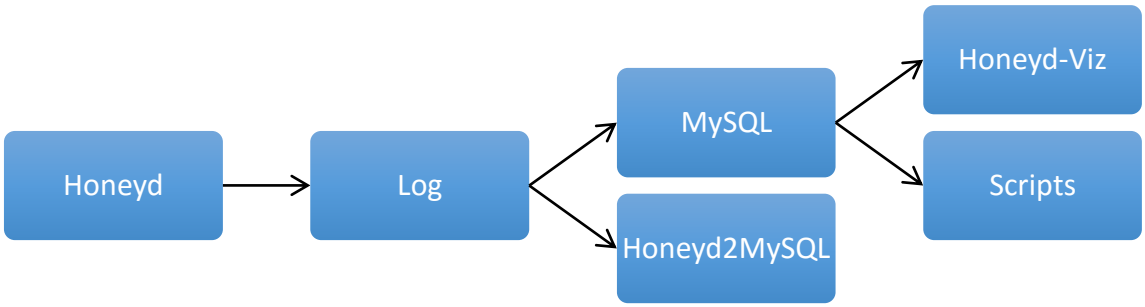


Figura 3.1 Componentes del sistema honeyd

3.1.1 Instalaciones y configuración de Honeyd

Este honeypot está montado sobre un sistema operativo Ubuntu Server 12.04 y cuenta con los repositorios necesarios para instalar *Honeyd* sin problemas. La opción de instalación mediante compilación de código también se encuentra disponible. Una vez instalado *Honeyd*, hay que editar el archivo de configuración con el fin de personalizar los sistemas a emular.

3.1.2 Arquitectura de Honeyd

La configuración de los sistemas operativos emulados y sus servicios se realiza mediante la edición de un archivo denominado *honeyd.conf*. La estructura básica de un sistema y sus recursos se basa en bloques de configuración formando una plantilla para cada tipo de sistema. Cada platilla consta de los siguientes elementos principales:

- Nombre plantilla que identifica al sistema a emular.
- Identificación del sistema operativo.
- Puertos TCP/UDP/ICMP.
- Estado de cada puerto *open, reset, block, etc*. o un script que gestione la conexión.

Honeyd contiene una gran cantidad de scripts para simular la aplicación, aunque permite la generación de scripts personalizados en casi cualquier lenguaje de programación. A continuación, se mencionan estos scripts en la siguiente tabla:

Tabla 3.1 Scripts de simulación para honeyd.

• apache.sh	• finger.sh	• exchange-nntp.sh
• cyrus-imapd.sh	• lpd.sh	• exchange-pop3.sh
• echo.sh	• rcp.sh	• exchange-smtp.sh
• ident.sh	• squid.sh	• iis.sh
• qpop.sh	• syslogd.sh	• ldap.sh
• sendmail.sh	• wuftpd.sh	• msftp.sh
• ssh.sh	• ftp.sh	• vnc.sh
• telnetd.sh	• proxy.pl	• iis-0.95
• bo.sh	• smtp.pl	• wen.sh
• discard.sh	• exchange-imap.sh	

Uno de los aspectos más interesantes es el método que utiliza *Honeyd* para proporcionar un conjunto de huellas sobre un sistema operativo cuando este es analizado por un atacante. Este método utiliza archivos que contienen huellas o patrones de comportamiento asociados a un sistema operativo en concreto que lo hacen único en la mayoría de los casos. De esta forma, Honeyd puede usar estos archivos para recrear el comportamiento de la pila TCP/IP de los paquetes de red de manera que consigue una identificación específica para el sistema emulado, el cual se muestra en la siguiente figura:

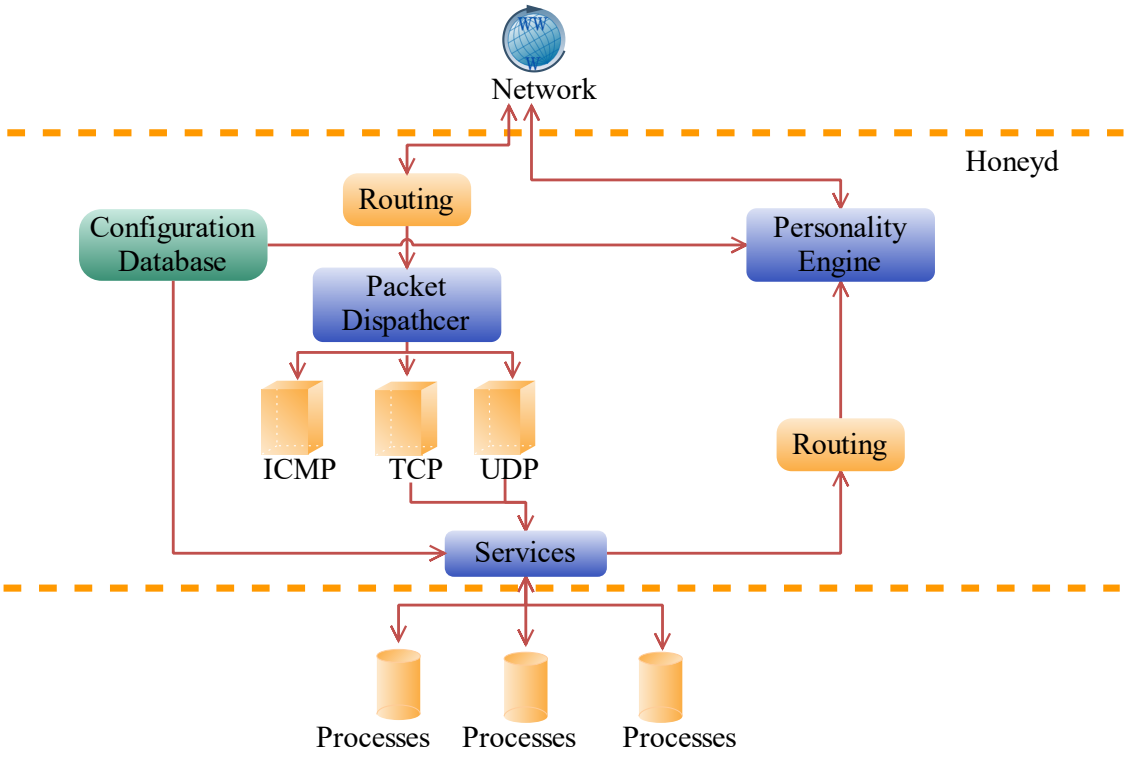


Figura 3.2 Arquitectura básica de honeyd

3.1.3 Captura de datos con Honeyd

Honeyd almacena la información recopilada en los siguientes archivos:

- **Syslog:** Registra toda la información acerca de los establecimientos de conexión en el archivo de log del sistema mediante *syslogd*, la herramienta de gestión de logs en sistemas UNIX/Linux.
- **honeyd.log:** Es otro archivo donde se almacena la información sobre la actividad del honeypot. Este archivo es opcional y se puede activar incluyéndolo como un parámetro en el comando de inicialización de *Honeyd*. El contenido es el mismo que el volcado en *syslog*.
- **Otros:** Cada servicio programado con scripts puede llevar su propio log independiente donde registra la actividad relacionada con el servicio, incluyendo más información, además, de la conexión realizada.

Respecto al sistema de alertas, este honeypot no cuenta con ningún sistema de avisos cuando se detecta un intento de intrusión o interacción con él. Al hacer uso de terceras herramientas que supervisen los logs y generan alertas cuando sean detectadas.

En cuanto a la generación de informes y obtención de la información de los logs, es necesario utilizar *plugins* y herramientas comentadas en el punto anterior. Mediante estas herramientas se supervisará la actividad del *honeypot* y extraerá la información de los logs de una forma más cómoda, ya que la lectura directa de los logs de texto no es práctica.

3.1.4 Manejo de bitácoras

Honeyd recopila información de los siguientes parámetros del tráfico relacionados con los intentos de conexión:

- Direcciones IP de origen y destino.
- Puertos TCP/UDP.
- Estampas de tiempo.
- Estado de la conexión.
- Información sobre el estado de la conexión, la identificación del sistema operativo del cliente o la herramienta utilizada para el escaneo.

Si se ha utilizado algún script para la emulación de un servicio, este contendrá información diversa, por ejemplo, el script MSFTP almacena información en un log sobre todos los comandos introducidos, incluyendo usuarios y contraseñas utilizada en el login del servicio.

3.2 KIPPO

Kippo es un honeypot de baja interacción que emula servicios SSH para detectar intrusos que intentan acceder al sistema a través de puerto 22, por lo tanto, utilizan para ello técnicas de ataques de fuerza bruta.

Por la gran capacidad de este honeypot para recopilar información para su posterior análisis, donde la gran mayoría de los ataques de fuerza bruta realizados sobre una

maquina expuesta en la red, tienen un fin más allá de la penetración del sistema, estos pueden ser desde la suplantación de la identidad de la víctima hasta la descarga de software malicioso desde la propia terminal. [8]

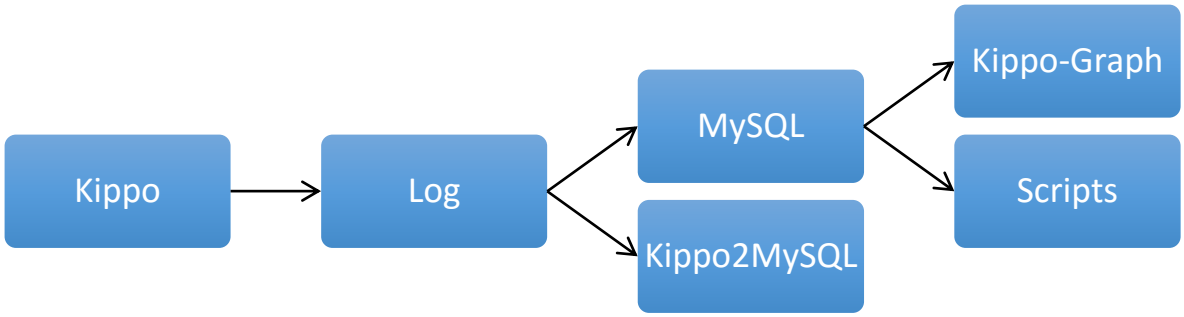


Figura 3.3 Componentes del sistema Kippo

3.2.1 Instalaciones y configuración de Kippo

Se ha escogido Ubuntu server 12.04 como sistema operativo base. La instalación de *Kippo* es rápida y sencilla ya que en los repositorios se puede encontrar todas las dependencias necesarias, se ha optado por instalar el honeypot a través de su repositorio SVN (*Subversión*) para obtener la última versión disponibles, aunque también es posible hacerlos descargando el paquete con el código fuente del mismo.

Kippo tiene un archivo de configuración muy básico donde se pueden modificar las localizaciones de los archivos de log y de la aplicación, conexiones a la base de datos y al protocolo XMPP, puerto de escucha, etc. Ya que simula un servidor SSH, comúnmente utilizado para la administración de la máquina, *Kippo* viene configurados con el puerto de escucha por defecto 2222 en lugar del puerto bien conocido 22 para SSH.

Este hecho puede ocasionar un problema, ya que debería escuchar en el puerto 22 como sería lógico. Esta modificación se puede realizar en el archivo de configuración del honeypot, pero en sistema UNIX/Linux, tan solo el usuario root puede utilizar los puertos inferiores al 1024.

Teniendo en cuenta que no es deseable ejecutar el honeypot bajo la cuenta de root, se proponen dos opciones para solventarlo:

- **IPTables:** Mediante una regla de NAT, se puede redirigir el tráfico destinado al puerto 22 de escucha del honeypot 2222.
- **Authbind:** Es una aplicación que permite al usuario sin permisos acceder a recursos de red privilegiados, como es el caso de los puertos bien conocidos. Estos permisos son concedidos por el usuario *root*.

Para las pruebas realizadas, en el caso expuesto sobre los puertos utilizados y se toma por defecto el puerto configurado en *Kippo,* 2222.

3.2.2 Arquitectura de Kippo

Al tratarse de un honeypot de uso específico, el único servicio disponible es el ya comentado servicio de conexión remota SSH. Dado que la función de SSH es brindar una conexión segura para conectarse a una maquina remota y ofrecer una terminal que permita trabajar como si fuera un usuario local, para entender como gran parte de este honeypot se basa en proporcionar una interacción adecuada posterior a la conexión, de ahí la implementación del sistema de archivos y comando que proporciona *Kippo.*

Aunque este honeypot solo exponga el servicio SSH, también es necesario valorar la suite de utilidades completa que dotan de realismo a este servicio y que han conseguido que *Kippo* sea uno de los honeypots más interesantes de su categoría.

3.2.3 Captura de datos con Kippo

Kippo mantiene varios sistemas de *logging,* estos son:

- **Logs de texto.** Se mantiene un log de texto que almacena toda la actividad relacionada con el honeypot, incluyendo la interacción por consola de un intruso, el archivo de log es *Kippo.log.* También cabe la posibilidad de habilitar otro módulo de *logging* de formato similar, pero algo más sencillo y centrado en las intrusiones, si se habilita, la información será registrada en *Kippo-textlog.log.*

- **MySQL.** Registra la actividad en archivos de logs, también es posible almacenar toda la información en una base de datos. Esta opción es muy recomendable, ya que facilita la extracción de los datos para generar estadísticas y muchas utilidades extras trabajan a través de esta base de datos.
- **XMPP.** *Extensible Messaging and Presence Protocol,* este módulo permite la notificación en tiempo real a través de canales de chat y servidores *Jabber.* Este servicio también está integrado en *Dionaea* como se describió en el análisis del mismo.

Kippo tampoco cuenta con un sistema de alertas, por lo tanto, habrá que volver a recurrir a herramientas de terceros o a la programación de scripts personalizados que accedan a alguno de los métodos de logging descritos y, emitir las alertas adecuadas.

Respecto a la generación de informes, es de gran ayuda tener habilitado el módulo para registrar la actividad en la base de datos, así, mediante consultas SQL, se puede extraer la información requerida. *Kippo-Graph*, permite establecer filtros y obtener graficas de calidad para poder utilizarlas en un informe ejecutivo.

3.2.4 Manejo de bitácoras

La información de recopila *Kippo* en sus logs es la siguiente:

- Direcciones IP de Origen.
- Puertos TCP/UDP.
- Registros de usuarios y contraseñas introducidos.
- Comandos introducidos por el atacante en la consola.
- Información de depuración del honeypot, como el establecimiento y cierre de una conexión, intercambio de claves, etc.

Kippo ha demostrado que es un honeypot del que se pueden obtener muchos beneficios, gracias a su alto grados de simulación y a la captura de los binarios descargados por un atacante. Es recomendable su uso para aquellos que quieran capturar muestras de nuevos *rootkits* y tener un registro de cada uno de los pasos trazados por el intruso en el sistema virtual.

3.3 DIONAEA

Dionaea es un honeypot de baja interacción y de propósito general, que ofrece una variedad de servicios de redes. Se desarrolló con el fin de poder recolectar malware para ser analizado posteriormente. Está escrito en lenguaje C, pero usa Python embebido como lenguaje de script para desarrollar los servicios de forma modular, [8].

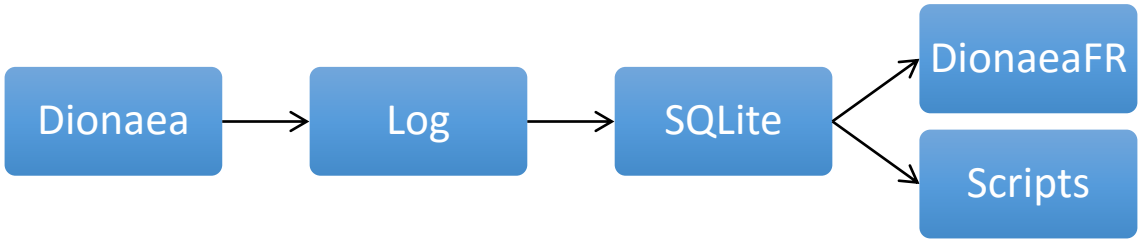


Figura 3.4 Componentes del sistema Dionaea

3.3.1 Instalaciones y configuración de Dionaea

Se ha escogido Ubuntu server 12.04 como sistema operativo para instalar *Dionaea* por ser el recomendado por los desarrolladores del honeypot, ya que simplifica enormemente el proceso de instalación y reduce las dependencias entre librerías.

Aunque existe la posibilidad de instalarlo mediante la compilación de las fuentes, se ha optado por usar los servicios de un repositorio que ya contiene las fuentes compiladas, automatizando prácticamente la instalación, sin duda, la opción más aconsejable.

Tras la instalación del honeypot, está listo para ser ejecutado sin necesidad de realizar ninguna configuración previa. Aunque la configuración por defecto puede ser suficiente, es posible realizar modificaciones a través de un archivo de configuración sencillo. En cambio, si se necesita desarrollar o modificar el funcionamiento de algún servicio, es necesario tener conocimientos de Python, ya que, aunque cada servicio se desarrolla en un módulo independiente para su mejor implementación, su programación esta poco documentada.

3.3.2 Arquitectura de Dionaea

Dionaea trae por defecto una serie de servicios activados a la espera de conexiones entrantes como se muestra en la siguiente tabla.

Tabla 3.2 Servicios ofrecidos por Dionaea.

Servicio	Puerto	Descripción
HTTP	80 TCP	Servidor web
HTTPS	443 TCP	Servidor web seguro
TFTP	69 UDP	Transferencia de archivos
FTP	21 TCP	Transferencia de archivos. Modo pasivo
SMB	445 TCP	Compartición de archivos e impresoras
SIP	5060 TCP/UDP	Comunicaciones en vivo de voz y video
SIP	5061 TCP	Comunicaciones en vivo de voz y video
MSSQL	1433 TCP	Bases de datos Microsoft
MYSQL	3306 TCP	Bases de datos MySQL

3.3.3 Captura de datos con Dionaea

La información recopilada por el honeypot es almacenada en varios archivos:

- **Dionaea.log:** Registra toda la actividad sospechosa que es detectada por el honeypot.
- **Dionaea-error. log:** Acumula información sobre los errores de la aplicación.
- **Logsql.sqlite:** Almacena toda la información referente a los ataques producidos al igual que en *Dionaea.log*, pero en una base de datos para un mejor tratamiento de los datos.

Dionaea no cuenta con un sistema de alertas, sino que hay que inspeccionar los archivos de logs para poder detectar si está ocurriendo una actividad sospechosa o ilícita. Empleando comandos de la consola Linux es posible monitorizar el contenido de estos logs, aunque siempre es posible desarrollar un módulo o scripts que generen alertas por SMS o email, por ejemplo. Otra forma de generar alertas es mediante el módulo logXMPP

descrito anteriormente, donde varios clientes recibirán notificaciones de los ataques ocurridos. Existen varias formas de recuperar la información almacenada en los logs. Una opción es revisar los logs de texto manualmente, pero es algo no deseada porque contienen demasiada información. Otra opción es recuperarla de la base de datos mediante consultas SQL.

Dionaea proporciona sentencias ya estructuradas para obtener esta información, evitando tener que estudiar la estructura de la base de datos, para poder generar sus propias consultas. Con el módulo GnuplotSQL se pueden generar gráficas para incorporarlas a un informe rápidamente. Otra forma es utilizar otras herramientas como DionaeaFR, que facilita enormemente la obtención de la información mediante una interfaz web muy visual y mucho más cómoda.

3.3.4 Manejo de bitácoras

Dionaea recopila prácticamente todos los datos acerca de un ataque, revisando los logs se puede ver información acerca de:

- Direcciones IP de origen y destino.
- Puertos TCP/UDP.
- Registros de usuarios y contraseñas introducidos en los servicios.
- Vulnerabilidades aprovechadas por los exploits.
- Binarios capturados.
- Shellcodes y llamadas del sistema.
- Comandos introducidos para interactuar con los servicios.
- URLs para la descarga de malware.

Como se puede observar, *Dionaea* proporciona información muy valiosa que permite a los administradores de sistemas conocer con detalle cualquier actividad que se produzca en el honeypot.

Página intencionalmente en blanco

CAPÍTULO 4. ESCENARIO DE ESTUDIO Y HERRAMIENTAS

"Comienza haciendo lo que es necesario, después lo que es posible y de repente estarás haciendo lo imposible."

San Francisco de Asís

Este capítulo tiene como objetivo analizar tres de los sistemas honeypots mencionados en el capítulo tres, a través de criterios de evaluación tal como se describirán en el capítulo cinco. Se dará a conocer las herramientas que permitirán realizar un análisis técnico de cada honeypot. Estos criterios y herramientas se describirán más adelante en este capítulo. Los honeypots escogidos se han seleccionado del amplio catálogo disponible de honeypots en función de su popularidad, características y lo servicios que son capaces de emular.

4.1 SELECCIÓN DE HONEYPOT

Una de las primeras decisiones importantes a tomar en cuenta, fue el tipo de honeypots que se debe usar para la implementación. Por una variedad de razones, se decidió que los honeypots de baja interacción deberían ser preferidos a los de alta interacción. La posibilidad de que un honeypot de alta interacción fuera totalmente comprometido y permitiera que un atacante lance nuevos ataques hacia otros sistemas dentro o fuera de la red después de haber tomado bajo control completo el sistema honeypot.

En cambio, los honeypots de baja interacción mediante la simulación de servicios de red que no pueden otorgar el control total de los mismos al intruso, se consideraron una opción más sabia. Finalmente, razones como costo y facilidad de despliegue y mantenimiento también jugaron un papel importante en la decisión.

El Proyecto Honeynet [8] proporciona una lista completa de las herramientas y servicios de software de honeypot disponibles. Además, debido al entorno de red, los honeypots seleccionados deben emular con éxito los servicios más vulnerables y abarcar los potenciales ataques que una red.

Las características de los honeypots elegidos que llevaron a esta selección.

- Honeyd. La herramienta de software de Honeyd con su capacidad de engañar a las herramientas de fingerprinting de la red utilizadas por los atacantes a través de la simulación de la pila de red de una gran gama de sistemas operativos existentes, era una solución bastante atractiva para nuestra implementación.
- Dionaea. Es considerado como uno de los mejores coleccionistas de malware que el software de código abierto tiene para ofrecer. Emula principalmente un servidor SMB en el puerto 445 que se considera uno de los puertos más específicos. Este último puede garantizar un gran número de malware capturado. Además, la oportunidad de descargar y analizar las copias del malware y emular otros servicios de perspectiva de seguridad de la red.
- Kippo. Se basó en el hecho de que queríamos emular a un servidor SSH, común de los ataques de red. Las características de Kippo eran bastante desafiantes, incluyendo la capacidad del software para reproducir un ataque real. Además, la base de datos de Kippo nos puede proporcionar información importante y útil sobre los nombres de usuario y contraseñas más utilizados, así como información sobre las actividades e intenciones de los atacantes.

Al incluir los tres honeypots de múltiples funciones mencionados anteriormente dentro de la implementación, se podrá monitorear la mayoría de los posibles tipos de ataques y obtener estadísticas importantes sobre ellos.

En la tabla siguiente, se presenta un resumen de las características de cada herramienta de honeypot de baja interacción para la selección:

Tabla 4.1 Características de los honeypots seleccionados.

Honeyd	Kippo	Dionaea
• Open source. • Sistema de bajo riesgo. • Facilidad en la implementación. • Capacidad de simular y monitorizar diferente host virtual en múltiples dirección IP simultáneamente. • Mecanismo de direccionamiento de huellas dactilares que simulan los sistemas operaritos de la pila TCP/UDP. • Capacidad de engañar a las herramientas de huellas dactilares de la red como Nmap. • Configuración de scripts que imitan servicios de red.	• Open source. • Sistema de bajo riesgo. • Facilidad de implementar y mantener. • Emulación del servidor SSH. • Mantiene registro del nombre de usuario y contraseñas. • Capacidad de reproducir un ataque real.	• Open source. • Sistema de bajo riesgo • Facilidad de implementar y mantener. • Capacidad para analizar actividades de programas maliciosos. • Emulación del servidor SMB en el puerto 445. • Oportunidad de descargar y analizar copias de malware.

4.2 ANÁLISIS DE DATOS

Se ha establecido un conjunto de elementos a evaluar en cada honeypot. Para cada elemento a evaluar se realizará un breve análisis, documentando las características de la información obtenida mediante el uso de herramientas de testing y de la documentación del honeypot. De este modo la arquitectura completa que define honeypot, es la formada por todas las partes anteriormente mencionadas, la cual se puede observar en la siguiente figura.

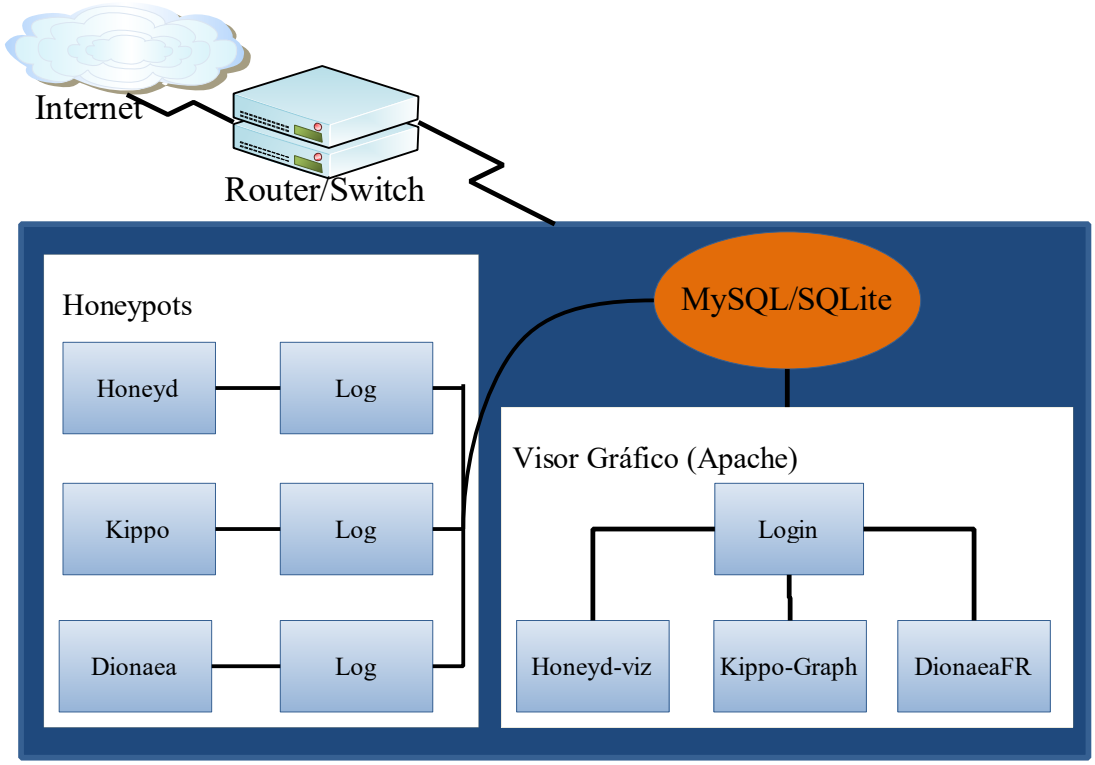


Figura 4.1 Componentes de arquitectura honeypot.

4.2.1 Instalación y facilidad de uso

Se evaluará la dificultad de la instalación del honeypot para el sistema. También se analizará el grado de dificultad para configurar el honeypot de forma básica, además, de la creación y personalización de los servicios ofrecidos.

4.2.2 Servicios ofrecidos

Cada honeypot ofrece un conjunto de servicios que serán enumerados junto con un abreve descripción y propósito de cada uno de ellos. Este estudio permitirá crear una idea del uso que se le puede dar a un honeypot en concreto.

4.2.3 Realismo de los servicios emulados

Este criterio intentara realizar una comparación de las similitudes o diferencias entre un servicio emulado por un honeypot y un servicio real. Este estudio tiene como fin exponer la afinidad y realismo de los servicios emulados, que cuanto mayor sea la semejanza a un servicio real, mejores resultados se obtendrán y más difícil será detectar al honeypot. Una comparación entre los mismos servicios de diferentes honeypots permitirá tener elementos de valor para seleccionar un honeypot determinado.

4.2.4 Gestión de logs, alertas e informes

Un punto de los más importantes es la gestión de la información recopilada por el honeypot. Se valorarán características como la gestión de logs, el uso de bases de datos, almacenamiento remoto de logs, el formato de la información almacenada y el mecanismo necesario para acceder a toda esta información, es decir, realizar consultas a la base de datos, examinar logs en modo texto, opciones de visualización web o generación de gráficas, etc. La gestión de alertas también es importante, para poder analizar continuamente los logs para encontrar indicios de actividad sospechosa.

4.2.5 Calidad de los datos recopilados

Hay que considerar si el tipo de información que se obtiene del honeypot es útil desde un punto de vista técnico y si esta tiene el nivel de detalle suficiente para poder obtener una taza de las acciones realizadas. Los datos que interesan pueden ser direcciones de origen del ataque, comandos ejecutados, modificaciones realizadas en el sistema, credenciales capturadas, binarios o malware cargado en el honeypot, etc.

4.3 HERRAMIENTAS DE VISUALIZACIÓN

El sistema de visualización implementado en un sistema honeypot, se basa de dos partes principales, por un lado, la parte del sistema de login y por otro lado la parte de los visualizadores gráficos, los cuales dan todo tipo de información sobre los honeypot instalados en el sistema, estos sirven de gran ayuda a la hora de realizar un análisis forense previo a la clasificación de los ataques recibidos.

Por otro lado, se han empleado tres visualizadores de base de datos, los cuales permiten obtener una información de estas, sin importar donde se encuentre en el momento del análisis.

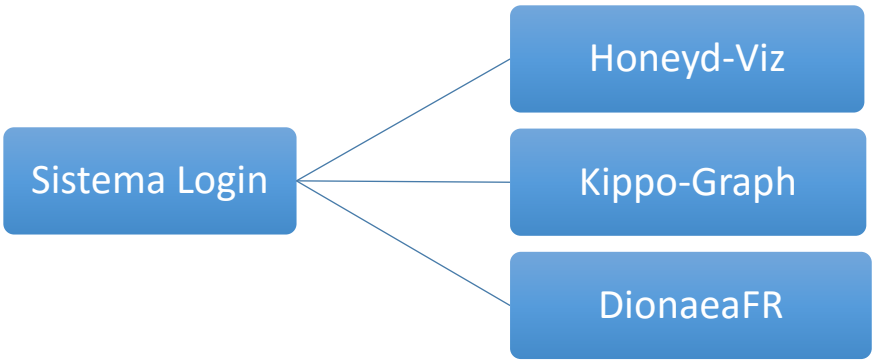


Figura 4.2 Sistema de Visualización

4.3.1 Honeyd-Viz

Para la instalación de este entorno gráfico, se ejecutan los comandos que se describen en el script de instalación de este honeypot. Para realizar su instalación, seguimos los mismos pasos que para el honeypot Kippo, en un primer lugar se descarga el sistema desde el mismo repositorio que para Kippo-Graph y posteriormente proceder a descomprimir dicho archivo y finalizar su instalación dando permisos de ejecución.

Una vez instalado todo el sistema, se modificará el archivo de configuración incluido en el directorio actual de este, el cual se llama `config.php`. En este archivo, se realizará la configuración personalizada para que este se conecte con la base de datos, por lo tanto, se modificarán los datos de acceso a la base de datos e ingresar datos propios.

En este sistema de visualización grafica al igual que para Kippo-Graph existen varias páginas que muestran todo tipo de gráfico y tablas que hacen referencia a los datos extraídos por dicho honeypot, estas son:

- **Index:** Una página de inicio en donde se observará un enlace importante, a través del cual se pueden generar los gráficos correspondientes a la última modificación de la base de datos.
- **Honey-Viz Graphs:** En esta página, se puede observar todos los gráficos generados por la herramienta, en los cuales se encontrará lo siguiente:
    - **Connections by Protocol:** En estos dos gráficos, se puede observar el número de conexiones por protocolos en este honeypot.
    - **Connections by IP:** En estos gráficos, se puede observar las conexiones por IP de destino que ha detectado en el sistema honeypot.
    - **Connections per Day/week:** En estos gráficos, se puede observar las conexiones realizadas por día y por semana al sistema honeypot.
    - **Connections per IP:** En estos gráficos, se puede ver las conexiones por IP de origen que ha detectado Honeyd.
    - **TCP Connections per IP:** En estos gráficos, se puede ver las conexiones por IP de origen que ha detectado Honeyd.
    - **UDP Connections per IP:** En estos gráficos, se puede observar las conexiones en el protocolo UDP realizadas en el sistema honeyd.
    - **ICMP Connections per IP:** En estos gráficos, se puede observar las conexiones en el protocolo ICMP realizadas en el sistema honeyd.
    - **Connections by Destination Port:** En estos gráficos, se puede ver las conexiones realizadas por puerto de destino.
- **Honeyd-Geo:** En esta página se puede ver todo lo referente a la geo localización de los atacantes en el sistema, esta cuenta con los siguientes gráficos y tablas:
    - **Tabla Top 10:** en esta primera tabla, se puede observar el top diez de los atacantes por número de pruebas realizadas, en donde se puede observar tanto la religión como el país de procedencia.
    - **Número de Conexiones:** En estos gráficos se puede observar el número de conexiones en el sistema honeyd en modo gráfico.

- **Google-Maps:** Se puede observar que se ha añadido un complemento adicional basado en Google-Maps, a través del cual se puede ver la posición de los ataques en un mapa de este estilo.
- **Maps:** En estos últimos dos gráficos se puede ver también la localización por países de los ataques.

- **Honeyd-Gallery:** En este apartado de la herramienta, se puede ver todos los gráficos generados en modo repositorio de imágenes, a través del cual se pueden descargar dichos gráficos a través del computador.

4.3.2 Kippo-Ghaph

El primer sistema de visualización grafica web es Kippo-Graph, en este hace referencia a los datos obtenidos de honeypot Kippo. Este sistema se basa en una serie de scripts programados en Python, los cuales generan gráficas y estadísticas obtenidas de la base de datos de dicho honeypot y las muestran en formato web. Se realizará una explicación sobre su funcionamiento, así como los pasos para su instalación y configuración.

Una vez instalado Kippo-Graph, se configurará el archivo `config.php` para que se extraiga todos los datos de la base datos, para ello es necesario modificar el sistema de acceso a la base de datos incluyendo datos de acceso a la base de datos Kippo.

Kippo-Graph posee un gran número de herramientas o archivos PHP donde se pueden ver las distintas funcionalidades de este, estas son:

- **Kippo-índex:** Esta página corresponde al índex de Kippo-Graph, en donde se puede observar la versión instalada y las funcionalidades de cada una de las versiones de este sistema.
- **Kippo-Graph:** Esta página posee una gran cantidad de gráficos y datos sobre todas las estadísticas obtenida de dicho honeypot, están son:
    - **Actividad del honeypot:** En esta primera sección, se puede observar una tabla donde se detallan el número de IP que han atacado al sistema, número de intentos, fecha del primer ataque y fecha del último ataque.
    - **Top 10 passwords:** En este gráfico, se puede observar el top diez de las password ejecutadas para realizar los ataques.

- **Top 10 Usernames:** En este gráfico, se puede observar el top diez de los nombres de usuario ejecutados para realizar los ataques.
- **Top 10 User-pass:** en esto gráfico, se puede observar el top diez del combo usuario-contraseña que han sido utilizados para realizar los ataques. Adicionalmente existe un gráfico circular donde se observará dichos combos de usuario-contraseña en tantos por ciento.
- **Success Ratio:** En este grafico se puede ver la ratio de aciertos que se ha tenido durante toda la ejecución del sistema.
- **Success per day/week:** En estos tres gráficos, se muestrean los aciertos obtenidos por día y por semana durante la ejecución del sistema.
- **Conections per IP:** en estos dos gráficos, se puede ver el número de conexiones por IP durante la ejecución del sistema.
- **Successful Logins:** en este gráfico, se puede observar el nivel de acierto obtenido por cada una de las IP que han realizado un determinado ataque.
- **Probes per day/week:** En estos gráficos, se puede observar el número de pruebas o ataques llevados a cabo por día o por semana.
- **Top 10 SSH clients:** En este graficó, se puede observar el top diez de los clientes SSH utilizados para realizar los ataques.

Como se puede ver en esta página, se posee un gran número de gráficos que ayudan a obtener una mejora a la hora de obtener estadísticas o datos de la base de datos de kippo.

- **Kippo-input:** Esta página está especializada para el análisis de aquellos ataques que han tenido éxito y, además han conseguido entrar en el sistema y ejecutar un comando. Se pueden distinguir los siguientes campos:
  - **Overall:** En esta tabla se puede ver el número de comandos ejecutados y el número de comandos de descarga ejecutados en el sistema.
  - **Human activity inside:** en estos gráficos, se pude observar la actividad que se ha llevado a cabo en el honeypot una vez que este ha sido atacado, por lo tanto, han conseguido entrar dentro del sistema.
  - **Top 10 input:** esta tabla muestra el top diez de los comandos ejecutados en el sistema, así como el número de veces que estos han sido repetidos.

También se puede ver los mismos datos en formato grafico justo debajo de esta.

- **Top 10 Successful input:** Representa una tabla y un gráfico con el top diez de comandos ejecutados correctamente.
- **Top 10 failed input:** representa una tabla y un gráfico donde se puede observar el top diez de comando ejecutados incorrectamente.
- **Wget Commands:** Esta tabla representan el tipo de comando wget ejecutados en el sistema.
- **Executed scripts:** Esta tabla representa los comandos ejecutados para la ejecución de scripts en el sistema.
- **Interesting Commands:** En esta tabla se representa los comandos más interesantes ejecutados en el sistema.
- **Apt-get Commands:** Esta tabla representa los comandos de tipo Apt-get realizados en el sistema.

- **Kippo-Playlog:** esta página es una funcionalidad nueva de esta versión, en la cual se puede ver en modo video cada uno de los ataques realizados desde dentro del sistema.
- **Kippo-IP:** Esta página representa una tabla con las direcciones IP de todos los atacantes al sistema, donde una de las grandes ventajas de esta es la posibilidad de exportar dicha tabla a formato csv.
- **Kippo-Geo:** Esta página es una de las más importantes, ya que a través de ella se pueden localizar el top diez de ataques recibidos, así como realizar su posterior análisis mediante una serie de herramientas, estas son:
  - **Geolocalization Top 10:** En esta tabla se puede observar de que país y ciudad proviene el ataque, así como el número de ejecuciones llevadas a cabo. En la última columna de esta tabla se pueden observar una serie de enlaces a páginas para realizar su posterior análisis forense.
  - **Connections per unique IP:** En estos gráficos se pueden observar el top diez por países de los ataques realizados al sistema.
  - **Google maps:** Se ha añadido un visor de google-maps donde se puede observar mediante un icono de localización desde donde provienen el top diez de ataques.

    - **Gráficos de intensidad:** en estos dos últimos gráficos, se puede observar un mapa donde ver la intensidad del top diez de ataques recibidos, así como un gráfico circular.
- **Graph Gallery:** En esta última página se puede ver un repositorio de todos los gráficos anteriormente mencionado, donde se dispone de un link para guardar dichas imágenes.

4.3.3 DionaeaFR

Dionaea crea una base de datos SQL desde el archivo de registro que contiene toda la valiosa información. Desde esta perspectiva, es muy fácil leer y analizar todos los datos capturados. Aparte de los scripts SQLite realmente útiles que se explicarán más adelante, para el análisis también se utiliza una herramienta de visualización para Dionaea llamada DionaeaFR. Esta útil herramienta está escrita en Python puede dar una gran visión general de la operación de Dionaea. Para poder concentrarse en estadísticas específicas de ataque como el número de ataques por puerto, ataques por servicio, etc.

En la página principal del software se ve un resumen de la operación del honeypot con estadísticas sobre el número total de conexiones, el número de malware descargado y así sucesivamente. Por otra parte, se puede tener una mirada más profunda sobre una conexión específica y toda la información relativa a IP atacante y país, sistema operativo utilizado, el puerto atacado y la actividad del atacante posible. DionaeaFR proporciona estadísticas y gráficos sobre el número total de conexiones por puerto y por servicio. Lamentablemente, los gráficos DionaeaFR incluyen datos sólo para los últimos 7 días de la operación de honeypot y no para el período de ejecución total.

4.3.4 Sistema PhpMyAdmin

Uno de los sistemas de visualización de bases de datos empleado ha sido phpmyadmin, a través de este se puede acceder a cualquier base de datos estemos en cualquier lugar tan solo con la ayuda de cualquier navegador web.

4.4 BITÁCORAS SQL

Se ha implementado una serie de bases de datos y herramientas adicionales que hace que el sistema honeypot use el sistema MYSQL para el almacenamiento de todos los ataques detectados. Estas herramientas adicionales son:

4.4.1 Honeyd SQL

Honeyd2MySQL esta herramienta se ha considerado adicional, aunque es muy recomendable su uso, ya que Honeyd no dispone de una base de datos por defecto, y por esta herramienta sirve para visualizar los datos más cómodamente. Se basa de una herramienta donde se almacena información en una base de datos, obtenida de los archivos log generados por el honeypot. Hay que decir que no es necesario la creación de tablas en dicha base de datos, ya que la ejecución de honeyd2mysql, este creará todas las tablas indicadas en dicho script. [16]

Posteriormente antes de realizar la ejecución de este script, habrá que modificar los parámetros de base de datos indicados en dicho script, es decir, se modifica el script para indicarle la base de datos, nombre de usuario y contraseña para el acceso a la base de datos anteriormente creada, por lo tanto, quedará vinculada a dicho honeypot.

4.4.2 Kippo SQL

Kippo2MySQL esta herramienta es un script programado en perl, el cual extrae estadísticas básicas de los archivos log de kippo, siendo de gran ayuda a la hora de leer estos archivos, ya que son bastantes densos y tediosos de ver.

Su funcionamiento es básicamente la inserción de estas estadísticas extraídas del archivo log en una base de datos previamente creada, por lo tanto, su uso es esencial para la creación de una base de datos, la cual aparece en el anexo anterior y en donde se puede apreciar que se sigue el mismo procedimiento que para la base de datos Kippo, donde en este caso se va a crear la base de datos Kippo2MySQL, con el usuario root y dando permisos de ejecución al usuario kippo, posteriormente se accede a MySQL mediante el usuario kippo y se procede a la creación de las tablas correspondientes al script

Kippo2MySQL, aquí habrá tres tablas determinadas: auth, clients y hosts, donde se registraran los autores de los ataques direcciones IP, los clientes SSH para su acceso y los hosts detectados respectivamente. [17]

4.4.3 Dionaea SQL

En el caso de Dionaea, la base de datos SQL puede dar una vista detallada de la operación del honeypot. Dionaea crea 26 tablas diferentes, incluyendo la tabla de conexión con información sobre el tipo de conexión, el puerto conectado, la dirección IP de origen y así sucesivamente. Otras tablas interesantes incluyen la tabla de descarga, con detalles sobre los archivos descargados y la tabla de conexiones que contiene las credenciales de inicio de sesión utilizadas y registradas por Dionaea. Se puede ver información detallada sobre los comandos MS-SQL o MySQL que se están utilizando y las estadísticas sobre el protocolo SIP. La importancia de la base de datos SQL es que crea scripts de consulta sencillos para exportar toda la información deseada. [16]

4.5 DESCRIPCIÓN DEL ESCENARIO DE CAMA DE PRUEBAS

Para realizar el análisis de los honeypots es necesario un entorno configurado adecuadamente que permita una conexión entre el atacante y el honeypot. Dicho entorno será implementado mediante Virtual Box, proporcionando un dominio virtualizado de pruebas. Los honeypots se instalarán en máquinas virtuales con el sistema operativo recomendado para cada uno de ellos y que será comentado en la sección correspondiente de cada honeypot. Para llevar a cabo las baterías de pruebas sobre los honeypots se ha optado por utilizar una distribución de Linux orientada a realizar test de penetración. Esta distribución recopila una gran cantidad de herramientas de seguridad y hacking instaladas en el sistema, facilitando el trabajo del analista que en esta ocasión toma el papel del atacante como se ve en la figura 4.1. [18]

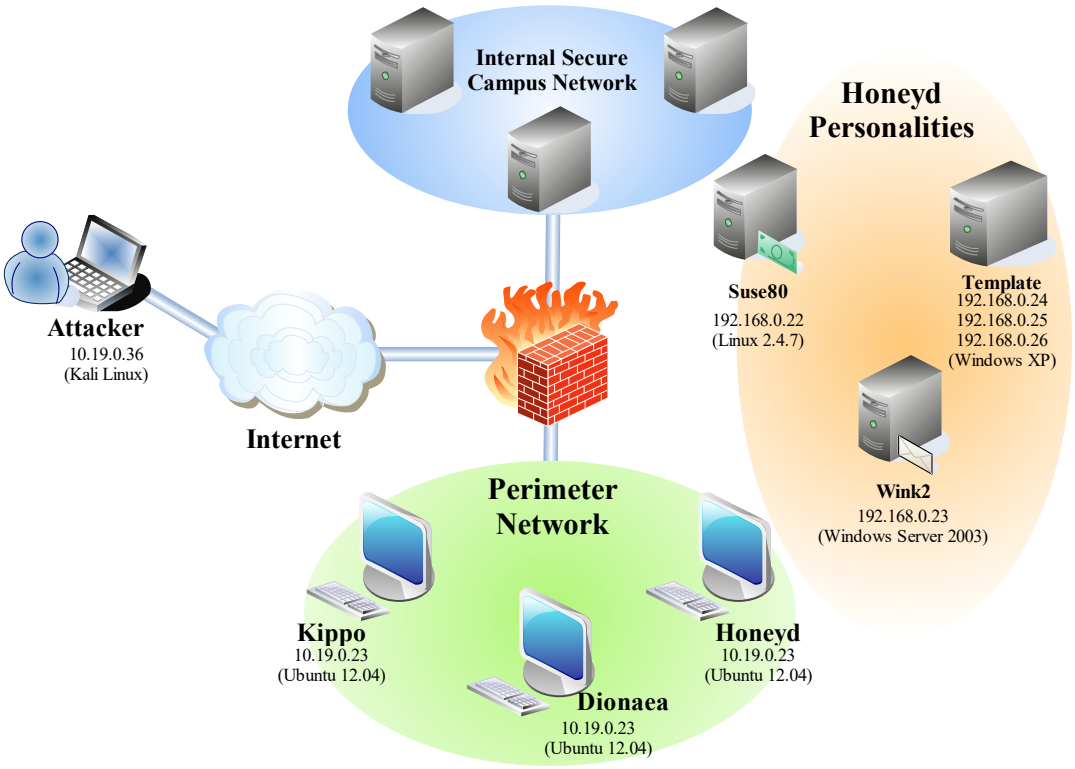


Figura 4.3 Arquitectura Honeypot implementada.

4.5.1 Fase de diseño

De este modo, se han definido los componentes necesarios para el sistema honeypot a implementar. En este punto, se sabe que el entorno consistirá en los siguientes componentes:

- Una máquina física.
- Un programa de virtualización.
- Sistemas honeypots mencionados en el capítulo tres de este documento.
- Sistema para visualizar los contenidos.
- Scripts y programa para el análisis.

A continuación, se va a realizar una breve introducción sobre los sistemas empleados en la fase de diseño, más adelante se realizará una explicación más densa sobre dichos sistemas y herramientas.

- **Máquina física**: La máquina física se trata de un PC con las siguientes características:
    - Sistema operativo: XUbuntu
    - Versión: 12.04
    - Fabricante del sistema: Acer
    - Modelo del sistema: Aspire 52
    - Procesador: Intel Core i3-380M
    - Gráfica: AMD Raedon HD 6370M
    - RAM: 4GB DDR3
    - Disco: 500GB HDD
    - Configuración regional: España
- **Programa de virtualización**: Como software de virtualización, se tomó Virtual Box con la versión 5.1.16. Este programa es de una gran versatilidad permite el arranque de varias máquinas virtuales con funcionalidad completa o no.

  Además, incluye una gran variedad de herramientas de administración de cada una de las máquinas virtuales disponibles en el sistema, las cuales facilitan las principales configuraciones deseadas para desplegar el honeypot.

CAPÍTULO 5. IMPLEMENTACIÓN DE LOS SISTEMAS HONEYPOTS

"Solo porque algo no haga lo que se quiere, no significa que sea inútil el esfuerzo."

Thomas Alva Edison

5.1 FASE DE IMPLEMENTACIÓN

En este capítulo, se ofrece una descripción detallada de la implementación honeypot. En concreto, se demuestran todos los pasos de instalación del sistema honeypot y se describe todas las configuraciones y recursos necesarios. Para el alcance de esta implementación, se integraron las tres herramientas de software de honeypot mencionadas anteriormente en la arquitectura de honeypot. Las herramientas Honeyd, Kippo y Dionaea constituyen el núcleo funcional de la implementación. Una visión general de la topología de honeypot implementada se muestra en la Figura 4.1 del capítulo cuatro.

Éstos incluyeron el emulador de código abierto PuTTY que puede ser utilizado como un cliente para conexiones SSH, Telnet o incluso conexiones TCP, Nmap que es una herramienta de escaneo de seguridad que detecta hosts y servicios y crea mapas de redes

Después de realizar una variedad de varias pruebas incluyendo ping, tracertrouting, exploración y tentativas de conexión, se verificó que los hosts virtuales eran percibidos como sistemas reales por esas herramientas y correspondía con ellos como habrían hecho verdaderos anfitriones.

Y posteriormente se realizaron pruebas para poner en funcionamiento el sistema honeypot, y verificar su desempeño dentro de la red con herramientas como nmap y metasploit.

5.2 IMPLEMENTACIÓN HONEYD + HONEYD-VIZ

A continuación, se enmarca de forma detallada la instalación del sistema Honeyd y la herramienta de visualización Honey-viz y finalmente su instalación.

1. Instalación y configuración en Xubuntu

En primer lugar, abrir la terminal y actualizar el sistema:

```
root@honeyd:~# apt-get update & upgrade
root@honeyd:~# apt-get install honeyd honeyd-common
root@honeyd:~# apt-get install linux-generic Ubuntu-minimal
```

Para configurar honeyd existe un archivo de vital importancia en el cual residen los datos y parámetros que definen el comportamiento del honeypot, es el archivo `honeyd.conf` en el que se delimitan las características que tienen los equipos y servidores simulados para fungir como distractores para los posibles intrusos.

Al instalar honeyd en el sistema se crea automáticamente un archivo `honeyd.conf` que contiene configuración predeterminada, este archivo se modifica de acuerdo con las características de los equipos que se desean simular, a continuación, se presenta como ejemplo una posible configuración de un host:

```
create default
set default personality "FreeBSD 2.2.1-STABLE"
set default default tcp action reset
add default tcp port 80 open
add default tcp port 22 "ssh scripts/web.sh"
add default tcp port 113 reset
add default tcp port 1 reset
```

A continuación, se enmarcan los puntos importantes de la configuración de este ejemplo:

- Create: Parámetro que se utiliza para definir el nombre que tendrá el host a simular.
- Set: Parámetro que se utiliza para asignar el sistema operativo que simula el host creado, los sistemas operativos y las versiones disponibles de asignar se encuentran en el archivo ubicado en la ruta `/etc/honeypot/nmap.assoc`.

- Default: Parámetro que indica el protocolo a utilizar, en este caso es TCP, sin embargo, también es posibles asignar los protocolos UDP e ICMP.
- Action: Parámetro que determina el comportamiento del puerto, el parámetro *reset* significa que este puerto responde con un RST, que de acuerdo con la especificación TCP RFC793 se devuelve un paquete RST cuando se intenta conectar a un puerto sin servicio; en la tabla 3.1 se especifican los posibles comportamientos de los puertos.

Tabla 5.1 Protocolos, comportamiento y especificacion.

Protocolo	Comportamiento	Especificación
TCP	Open	Responde con SYN/ACK para establecer conexión. • El indicador SYN de TCP representa un pedido para establecer una conexión. • El indicador ACK indica que el paquete es un acuse de recibo.
TCP	Block	El paquete se pierde y no hay respuesta.
TCP	Reset	Significa que este puerto responde con un RST, que de acuerdo con la especificación TCP RFC793 se devuelve un paquete RST cuando se intenta conectar a un puerto sin servicio.
UDP	Open	No hay respuesta.
UDP	Block	El paquete se pierde y no hay respuesta.
UDP	Reset	Responde con un mensaje ICMP de error de puerto.
ICMP	Open	Responde con paquete ICMP.
ICMP	Block	El paquete se pierde y no hay respuesta.

- add default tcp port 80 open. Esta línea indica el número de puerto y el estado en el que se encuentra dicho puerto.
- `sh scripts/web.sh`. Esta línea indica que se utiliza un script para simular determinada acción, en este caso el script *web.sh* simulara servicios web, los

scripts se encuentran ubicados en la ruta `/usr/share/honeyd/script;` cabe mencionar que para estos scripts es necesario asignarle privilegios al directorio esto se logra con la siguiente línea de comandos:

```
root@honeyd:~# chmod 777 cd /usr/share/honeyd/scripts/
```

Una vez instalada las paqueterías, acceder a la carpeta donde está alojado honeyd:

```
root@honeyd:~# cd /etc/honeypot
```

A continuación, se presenta el archivo de configuración de un servidor real en el cual se configuran tres máquinas virtuales, Windows 2003, Windows XP y Linux suse8.0. Así mismo un Servidor web con sus respectivas direcciones IP y protocolos activos en cada uno de ellos.

El presente archivo se encuentra alojado en la siguiente dirección `cd /etc/honeypot` y para editar este archivo, teclear la siguiente instrucción y observar la configuración:

```
root@honeyd:~# nano /etc/honeypot/honeydconf3
create win2k
set win2k personality "Microsoft Windows Server 2003 Standard Edition"
set win2k default tcp action reset
set win2k default udp action reset
set win2k default icmp action reset
set win2k uptime 3867
set win2k droprate in 13
add win2k tcp port 80 "/usr/share/honeyd/scripts/win32/win2k/iis.sh $ipsrc $sport $ipdst $dport"
add win2k tcp port 110 "/usr/share/honeyd/scripts/win32/win2k/exchange-pop3.sh $ipsrc $sport $ipdst $dport"
add win2k tcp port 143 "/usr/share/honeyd/scripts/win32/win2k/exchange-imap.sh $ipsrc $sport $ipdst $dport"

create template
set template personality "Microsoft Windows XP Professional SP1"
set template uptime 1728650
set template maxfds 35
add template tcp port 80 "/usr/share/honeyd/scripts/web.sh"
add template tcp port 22 "/usr/share/honeyd/scripts/test.sh $ipsrc $dport"
```

```
set template uid 32767 gid 32767

create suse80
set suse80 personality "Linux 2.4.7 (X86)"
set suse80 default tcp action reset
set suse80 default udp action block
set suse80 default icmp action open
set suse80 uptime 79239
set suse80 droprate in 4
add suse80 tcp port 21 "/usr/share/honeyd/scripts/unix/linux/suse8.0/proftpd.sh $ipsrc $sport $ipdst $dport"
add suse80 tcp port 22 "sh /usr/share/honeyd/scripts/unix/linux/suse8.0/ssh.sh $ipsrc $sport $ipdst $dport"
add suse80 tcp port 25 "/usr/share/honeyd/scripts/unix/linux/suse8.0/sendmail.sh $ipsrc $sport $ipdst $dport"
add suse80 tcp port 79 "sh /usr/share/honeyd/scripts/unix/linux/suse8.0/fingerd.sh $ipsrc $sport $ipdst $dport"
add suse80 tcp port 80 "/usr/share/honeyd/scripts/unix/linux/suse8.0/apache.sh $ipsrc $sport $ipdst $dport"
add suse80 tcp port 3128 "sh /usr/share/honeyd/scripts/unix/linux/suse8.0/squid.sh $ipsrc $sport $ipdst $dport"
add suse80 tcp port 8080 "sh /usr/share/honeyd/scripts/unix/linux/suse8.0/squid.sh $ipsrc $sport $ipdst $dport"
add suse80 udp port 514 "/usr/share/honeyd/scripts/unix/linux/suse8.0/syslogd.sh $ipsrc $sport $ipdst $dport"



create default
set default default tcp action block
set default default udp action block
set default default icmp action block



bind 192.168.0.22 suse80
bind 192.168.0.23 win2k
bind 192.168.0.24 template
bind 192.168.0.25 template
bind 192.168.0.26 template
```

Tras haber configurado el archivo `honeyd.conf3` se instalaran dos de las herramientas que complementan el trabajo de la aplicación *honeyd*, las aplicaciones son:

- Farpd. Farpd rellenará el espacio de direcciones IP que no han sido asignadas en la red, dando al intruso información falsa respecto a las direcciones asignadas. Para instalarlo se agrega la siguiente línea de comando en una terminal:

```
root@honeyd:~# apt-get install farpd
```

- Nmap. Hay otro archivo importante en el honeyd que es el `nmap.prints`, este archivo guarda las huellas. Nmap usa este archivo para validar el sistema operativo de un equipo remoto y honeyd lo usa para emular la pila de protocolos IP para el sistema operativo, de esta manera es posible ir actualizando la lista con nuevos sistemas operativos. Para instalarlo se agrega la siguiente línea de comando en una terminal:

```
root@honeyd:~# apt-get install nmap
```

Una vez instaladas las aplicaciones necesarias se procede a su aplicación y finalmente el inicio de `honeyd` en el sistema. Se aplica `farpd` a todo el segmento de red en el que actúa `honeyd`:

```
root@honeyd:~# farpd 192.168.0.0/24
```

Para poder registrar todas las alarmas que se presentan es necesario tener un archivo de bitácoras –log el cual es de suma importancia, dado que de este archivo se obtendrá la información que se analizará y permitirá llegar a conclusiones importantes en este proyecto, dicho archivo se crea en la ruta `/var/log/honeypot/` con el siguiente comando:

```
root@honeyd:~# nano /var/log/honeypot/honeyd.log
```

Honeyd2mysql se encuentra en la página oficial http://bruteforcelab.com/honeyd2mysql, descargar honeyd2mysql bajo el siguiente comando para la parte SQL, extraer el archivo y guardar en la siguiente ruta:

```
root@honeyd:~# cd /opt
root@honeyd:~# wget http://bruteforcelab.com/wp-content/uploads/honeyd2mysql-0.3.tar
root@honeyd:~# tar xzf honeyd2myql-0.3.tar
root@honeyd:~# cd /opt/honeyd2mysql/
```

A continuación, se instala MySQL-server para tener un mejor control de los logs, al momento de su instalación pedirá asignar una contraseña para el server, insertar una contraseña, confirmar y proceder con la instalación:

```
root@honeyd:~# apt-get install mysql-server
```

Una vez instalado MySQL server, se crea una base de datos bajo el siguiente comando, así mismo pedirá la contraseña que se asignó al momento de la instalación y se puede acceder a la creación de la base de datos:

```
root@honeyd:~# mysql –u root –p
Enter password:
mysql> create database honeyd2mysql;
mysql> GRANT ALL ON honeyd2mysql.* TO root@localhost IDENTIFIED
BY 'honeyd'
mysql> exit
```

Para enlazar la base de datos con el archivo honeyd2mysql configurar el siguiente archivo y agregar los valores creados para el sistema honeyd mediante los siguientes parámetros, después de esto, guardar la configuración para el sistema:

```
root@honeyd:~# nano /opt/honeyd2mysql/honeyd2mysql.pl
     línea 26 #MySQL servers values – change accordingly!
     línea 27  my $sql_user = 'root';
     línea 28  my $sql_password = 'honeyd';
     línea 29  my $database = 'honeyd2mysql';
     línea 30  my $hostname = 'localhost';
```

2. Instalación de Honey-Viz

Para este sistema de visualización honey-viz, se necesitan descargar los siguientes programas: PHP versión 5.3.4, así como sus librerías, para eso insertar el siguiente comando:

```
root@honeyd:~# apt-get install libapache2-mod-php5
root@honeyd:~# apt-get install php5-gd
root@honeyd:~# apt-get install php5-mysql
```

Configurar `apache2.config` y agregar la siguiente línea al final del archivo para su compatibilidad con el sistema honeypot, ingresar mediante el comando:

```
root@honeyd:~# nano /etc/apache2/apache2.conf
     -> ServerName nombredelhost
```

Iniciar el servidor apache2 mediante el siguiente comando:

```
root@honeyd:~# /etc/init.d/

root@honeyd:~# apache2 start
```

Descargar el sistema honey-viz mediante el siguiente comando y dar privilegios a la carpeta `generated-graphs/`:

```
root@honeyd:~#          wget          http://bruteforcelab.com/wp-
content/uploads/honeyd-viz-0.2.tar
root@honeyd:~# mv honeyd-viz-0.2.tar /var/www
root@honeyd:~# cd /var/www
root@honeyd:~# tar xvf honeyd-viz-0.2-tar --no-same-permissions
root@honeyd:~# cd honeyd-viz
root@honeyd:~# chmod 777 generated-graphs/
```

Abrir el siguiente archivo de configuración y modificar las líneas para enlazar con la base de datos antes creada, seguidamente guardar la configuración y proceder a la ejecución del sistema honeyd:

```
root@honeyd:~# nano config.php
     línea 12 define('DB_HOST', 'localhost');
     línea 13 define('DB_USER', 'root');
     línea 14 define('DB_PASS', 'honeyd');
     línea 15 define('DB_NAME', 'honeyd2mysql);
```

Abrir un navegador y teclear la dirección IP para generar los gráficos de kippo-graphs

```
http://your-ip-address/honeyd-viz
```

3. Ejecución

Para la ejecución de `honeyd2mysql`, ejecutar el siguiente comando.

```
root@honeyd:~# cd /opt/honeyd2mysql/
root@honeyd:~# ./honeyd2mysql.pl
```

Esta operación puede tardar varios minutos, dependiendo de la actividad en el honeypot, también es recomendable realizarla una vez al día, ya que si no se hace no se observarán las entradas a la base de datos y parecerá como si el honeypot no hubiese estado trabajando.

Como último paso queda iniciar el servicio de *honeyd*, añadiendo la siguiente línea de comandos en una terminal:

```
root@honeyd:~# honeyd -d -f /etc/honeypot/honeyd.conf3 -l
/var/log/honeypot/honeyd.log
```

5.3 IMPLEMENTACIÓN KIPPO + KIPPO-GRAPH

A continuación, se enmarca de forma detallada la instalación del sistema Kippo y la herramienta de visualización Kippo-Graph y finalmente su instalación.

1. Instalación y configuración en Xubuntu

Se descargan los siguientes paquetes de Internet para la estación del sistema kippo:

```
root@kippo:~# apt-get install python-dev openssl python-openssl
python-pyasn1 python-twisted python-mysqldb
```

Kippo escucha en el puerto 2222 de forma predeterminada, lo cual está bien para propósitos de prueba, pero en realidad reduce las posibilidades de registrar cualquier ataque. Por lo tanto, sería bueno hacer que kippo escuche en el puerto 22. Para ello primero se debe cambiar el puerto que utiliza el servidor SSH, con el fin de ser capaz de conectarse de nuevo al sistema correctamente, para esto descargar la paquetería de SSH para el sistema mediante el siguiente comando:

```
root@kippo:~# apt-get install openssh-server openssh-client
```

Cambiar la opción de puerto 22 y reiniciar el servicio SSH con el siguiente comando:

```
root@kippo:~# nano /etc/ssh/sshd_config
```

Se crea un nuevo usuario que no sea root para ejecutar kippo:

```
root@kippo:~# adduser kippo
```

En la siguiente instrucción, modificar el archivo añadiendo el usuario kippo debajo de la instrucción del usuario root, para que este usuario tenga privilegios para hacer funcionar el sistema, guardar la configuración y salir de este archivo:

```
root@kippo:~# visudo

           -> kippo ALL=(ALL:ALL) ALL
```

Para descargar la última versión estable de kippo, se necesita la paquetería de instalación de git, el cual permitirá descargar este sistema para ello ingresar el siguiente comando de instalación:

```
root@kippo:~# apt-get install git
```

Descargar kippo mediante el siguiente comando así mismo entrar a la carpeta en la cual fue descargada la paquetería:

```
root@kippo:~# git clone https://github.com/desaster/kippo.git
root@kippo:~# cd /home/kippo/
```

Realizar una copia del archivo de configuración, mediante el siguiente comando:

```
root@kippo:~# cp kippo.cfg.dist kippo.cfg
```

A continuación, se instala MySQL-server para tener un mejor control de los logs, al momento de su instalación pedirá asignar una contraseña para el server, insertar una contraseña, confirmar y proceder con la instalación:

```
root@kippo:~# apt-get install mysql-server
```

Una vez instalado MySQL server, se crea una base de datos bajo el siguiente comando, así mismo pedirá la contraseña que se asignó al momento de la instalación y se puede acceder a la creación de la base de datos:

```
root@kippo:~# mysql –u root –p
Enter password:
mysql> create database kippo;
mysql> GRANT ALL ON kippo.* TO kippo@localhost IDENTIFIED BY
'root';
mysql> exit

root@kippo:~# mysql –u kippo –p
Enter password:
mysql> use kippo;
mysql> source ./doc/sql/mysql.sql;
mysql> exit
```

En el archivo `kippo.cfg`, configurar los parámetros para el sistema kippo, así mismo descomentar algunas líneas, para obtener y enlazar la base de datos con él, modificar los valores que se señalan, guardar la configuración y cerrar el archivo.

```
root@kippo:~# nano kippo.cfg
     -> ssh_port = 22
     -> hostname = root@webserver
     [database_mysql]
     -> host = localhost
     -> database = kippo
     -> username = kippo
     -> password = root
     -> port = 3306
```

El siguiente comando permitirá re-direccionar el servicio SSH a los puertos antes configurados en el sistema kippo.

```
root@kippo:~# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport
22 -j REDIRECT --to-ports 2222
root@kippo:~# iptables-save
```

2. Instalacion de Kippo-Graph

Para la instalación de kippo-graph se descargan las siguientes librerías, mediante el comando:

```
root@kippo:~# apt-get install php5-gd
root@kippo:~# apt-get install php5-mysql
root@kippo:~# apt-get install libapache2-mod-php5
```

Descargar el sistema kippo-graph mediante el siguiente comando y dar privilegios a la carpeta `generated-graphs/`:

```
root@kippo:~#  wget http://bruteforce.gr/wp-content/uploads/kippo-graph-
0.9.3.tar
root@kippo:~# mv kippo-graph-0.9.3.tar /var/www
root@kippo:~# cd /var/www
root@kippo:~# tar xvf kippo-graph-0.9.3.tar
```

```
root@kippo:~# cd kippo-graph/
root@kippo:~# chmod 777 generate-graphs/
```

Abrir el siguiente archivo de configuración y modificar las líneas para enlazar con la base de datos antes creada, posteriormente guardar la configuración y proceder a la ejecución del sistema kippo:

```
root@kippo:~# nano config.php
#Modificar los siguientes parámetros.
     define('DB_HOST','localhost');
     define('DB_USER','kippo');
     define('DB_PASS','root');
     define('DB_NAME','kippo');
     define('DB_PORT','3306');
```

Correr los servicios de apache y SSH mediante el siguiente comando:

```
root@kippo:~# service apache2 restart
root@kippo:~# service ssh start
```

Abrir un navegador y teclear la dirección IP para generar los gráficos de kippo-graphs:

```
http://your-ip-address/kippo-graph
```

3. Ejecución

Para la ejecución de `honeyd2mysql`, ejecutar el siguiente comando:

```
root@kippo:~# cd /opt/kippo2mysql/
root@kippo:~# ./kippo2mysql.pl
```

Esta operación puede tardar varios minutos, dependiendo de la actividad en el honeypot, también es recomendable realizarla una vez al día, ya que si no se hace no se observarán las entradas a la base de datos y parecerá como si el honeypot no hubiese estado trabajando.

Acceder como usuario kippo e ir al directorio `/home` iniciar el servicio de kippo, añadiendo la siguiente línea de comandos en una terminal:

```
root@kippo:~# su kippo
root@kippo:~# cd /home/kippo
root@kippo:~# ./start.sh
```

5.4 IMPLEMENTACIÓN DIONAEA + DIONAEAFR

A continuación, se enmarca de forma detallada la instalación del sistema Dionaea y la herramienta de visualización DionaeaFR y finalmente su instalación.

1. Instalación y configuración en Xubuntu

Abrir la terminal del sistema para obtener las últimas actualizaciones de seguridad e instalar la siguiente paquetería y dependencias de construcción requeridos para dionaea con el siguiente comando:

```
root@dionaea:~# apt-get install libtool libudns-dev libglib2.0-dev libssl-dev libcurl4-openssl-dev libreadline-dev libsqlite3-dev python-dev automake autoconf build-essential subversion git-core flex bison pkg-config libgc-dev libloudmouth1-dev libnl-dev libnetfilter-queue-dev
```

Instalar sistemas para la detección de librerías SQL:

```
root@dionaea:~# apt-get install sqlite3
root@dionaea:~# apt-get install libxml2-dev
root@dionaea:~# apt-get install libxslt1-dev
```

Se creará el directorio donde se instalarán de forma manual las siguientes librerías `liblcfg, libemu, libev, Python3, Cython libpcap y yaml:`

```
root@dionaea:~# mkdir /opt/dionaea
root@dionaea:~# mkdir dionaea
root@dionaea:~# cd dionaea
```

Instalación `liblcfg`:

```
root@dionaea:~# git clone
https://github.com/ThomasAdam/liblcfg.git
root@dionaea:~# cd liblcfg/code
root@dionaea:~# autoreconf -vi
root@dionaea:~# ./configure --prefix=/opt/dionaea/
root@dionaea:~# make install
root@dionaea:~# cd ..
```

Instalación `libemu`:

```
root@dionaea:~# git clone https://github.com/buffer/libemu.git
root@dionaea:~# cd libemu
root@dionaea:~# autoreconf -vi
root@dionaea:~# ./configure --prefix=/opt/dionaea/
root@dionaea:~# make
root@dionaea:~# make install
root@dionaea:~# cd ..
```

Instalación `libev`:

```
root@dionaea:~# git clone https://github.com/enki/libev.git
root@dionaea:~# cd libev
root@dionaea:~# ./configure --prefix=/opt/dionaea/
root@dionaea:~# make
root@dionaea:~# make install
root@dionaea:~# cd ..
```

Instalación `Python3.5`:

```
root@dionaea:~# wget http://python.org/ftp/python/3.5.0/Python-
3.5.0.tgz
root@dionaea:~# tar xfz Python-3.5.0.tgz
root@dionaea:~# cd Python-3.5.0
root@dionaea:~# ./configure --enable-shared --
prefix=/opt/dionaea/ --with-computed-gotos --enable-ipv6 LDFLAGS="-
Wl,-rpath=/opt/dionaea/lib/"
root@dionaea:~# make
root@dionaea:~# make install
root@dionaea:~# cd ..
```

Instalación `Cython`:

```
root@dionaea:~# git clone https://github.com/cython/cython.git
root@dionaea:~# cd cython
root@dionaea:~# /opt/dionaea/bin/python3.5 setup.py build
root@dionaea:~# /opt/dionaea/bin/python3.5 setup.py install
root@dionaea:~# cd ..
```

Instalación `curl`:

```
root@dionaea:~# git clone https://github.com/curl/curl.git
root@dionaea:~# cd curl
root@dionaea:~# autoreconf -vi
root@dionaea:~# ./configure --prefix=/opt/dionaea/
root@dionaea:~# make
root@dionaea:~# make install
root@dionaea:~# cd ..
```

Instalación `libpcap`:

```
root@dionaea:~# git clone https://github.com/the-tcpdump-
group/libpcap.git
root@dionaea:~# cd libpcap
root@dionaea:~# ./configure --prefix=/opt/dionaea/
root@dionaea:~# make
root@dionaea:~# make install
root@dionaea:~# cd ..
```

Instalación `yaml`:

```
root@dionaea:~# wget http://pyyaml.org/download/pyyaml/PyYAML-
3.12.tar.gz
root@dionaea:~# tar xfz PyYAML-3.12.tar.gz
root@dionaea:~# cd PyYAML-3.12
root@dionaea:~# /opt/dionaea/bin/python3.5 setup.py build
root@dionaea:~# /opt/dionaea/bin/python3.5 setup.py install
root@dionaea:~# cd ..
```

Instalación `Dionaea`:

```
root@dionaea:~# git clone https://github.com/rep/dionaea.git
root@dionaea:~# cd dionaea
root@dionaea:~# autoreconf -vi
root@dionaea:~# ./configure --with-lcfg-
include=/opt/dionaea/include/
--with-lcfg-lib=/opt/dionaea/lib/
--with-python=/opt/dionaea/bin/python3.5
--with-cython-dir=/usr/local/bin
--with-udns-include=/opt/dionaea/include/
--with-udns-lib=/opt/dionaea/lib/
--with-emu-include=/opt/dionaea/include/
--with-emu-lib=/opt/dionaea/lib/
--with-gc-include=/usr/include/gc
--with-ev-include=/opt/dionaea/include
--with-ev-lib=/opt/dionaea/lib
--with-nl-include=/opt/dionaea/include
--with-nl-lib=/opt/dionaea/lib/
--with-curl-config=/opt/dionaea/bin/
--with-pcap-include=/opt/dionaea/include
--with-pcap-lib=/opt/dionaea/lib/
--with-glib=/opt/dionaea
root@dionaea:~# make
root@dionaea:~# make install
```

Asignar permisos a los siguientes directorios:

```
root@dionaea:~# chown -R nobody:nogroup /opt/dionaea/var/dionaea
root@dionaea:~# chown -R nobody:nogroup /opt/dionaea/var/log
```

2. Instalación de DionaeaFR

Abrir la terminal del sistema e instalar PIP *python package manager* y el paquete `python-netaddr` con el siguiente comando:

```
root@dionaea:~# apt-get install python-netaddr
root@dionaea:~# apt-get install python-pip
```

Continuar con los requisitos previos utilizando PIP para la instalación automatizada bajo el siguiente comando:

```
root@dionaea:~# pip install --upgrade pip setuptools
root@dionaea:~# pip2.7 install django==1.8.18
root@dionaea:~# pip2.7 install pygeoip
root@dionaea:~# pip2.7 install django-pagination
root@dionaea:~# pip2.7 install django-tables2
root@dionaea:~# pip2.7 install django-htmlmin
root@dionaea:~# pip2.7 install django-compressor
root@dionaea:~# pip2.7 install django-filter
root@dionaea:~# pip2.7 install django-htmlmin
```

Descargar e instalar `PySubnetTree`:

```
root@dionaea:~# cd /opt/
root@dionaea:~# git clone https://github.com/bro/pysubnettree.git
root@dionaea:~# cd pysubnettree/
root@dionaea:~# /opt/dionaea/bin/python3.5 setup.py install
```

Descargar e instalar `DionaeaFR`:

```
root@dionaea:~# cd /opt/
root@dionaea:~# wget
https://github.com/RootingPuntoEs/DionaeaFR/archive/master.zip -
O DionaeaFR.zip
root@dionaea:~# unzip DionaeaFR.zip
root@dionaea:~# mv DionaeaFR-master/ DionaeaFR
```

Copiar y editar el archivo de configuración de ejemplo:

```
root@dionaea:~# cp /opt/DionaeaFR/DionaeaFR/settings.py.dist
/opt/DionaeaFR/DionaeaFR/settings.py
root@dionaea:~# mkdir /var/lib/dionaea
root@dionaea:~# nano /opt/DionaeaFR/DionaeaFR/settings.py
     -> ALLOWED_HOSTS = ['YOUR_IP', 'localhost', '127.0.0.1']
```

Obtener e instalar `Django-table2-simplefilter` manualmente:

```
root@dionaea:~# cd /opt/
root@dionaea:~#  wget https://github.com/benjiec/django-tables2-
simplefilter/archive/master.zip -O django-tables2-
simplefilter.zip
root@dionaea:~# unzip django-tables2-simplefilter.zip
root@dionaea:~# mv django-tables2-simplefilter-master/ django-tables2-
simplefilter/
root@dionaea:~# cd django-tables2-simplefilter/
root@dionaea:~# /opt/dionaea/bin/python3.5 setup.py install
```

Obtener las bases de datos GeoIP y Geolite para DionaeaFR:

```
root@dionaea:~# cd /opt/
root@dionaea:~# wget
http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
root@dionaea:~# wget
http://geolite.maxmind.com/download/geoip/database/GeoLiteCountry/GeoIP.dat.gz
root@dionaea:~# gunzip GeoLiteCity.dat.gz
root@dionaea:~# gunzip GeoIP.dat.gz
root@dionaea:~# mv GeoIP.dat DionaeaFR/DionaeaFR/static
root@dionaea:~# mv GeoLiteCity.dat DionaeaFR/DionaeaFR/static
```

Compilar e instalar `Node.js` desde las fuentes:

```
root@dionaea:~# cd /opt/
root@dionaea:~#    wget    http://nodejs.org/dist/v0.8.16/node-
v0.8.16.tar.gz
root@dionaea:~# tar xzvf node-v0.8.16.tar.gz
root@dionaea:~# cd node-v0.8.16
```

```
root@dionaea:~# ./configure
root@dionaea:~# make
root@dionaea:~# make install
```

Instalar LESS para npm *Gestor de paquetes node.js:*

```
root@dionaea:~# npm install -g less
root@dionaea:~# npm install -g promise
```

Iniciar el servidor web:

```
root@dionaea:~# mkdir /var/run/dionaeafr
root@dionaea:~# cd /opt/DionaeaFR/
root@dionaea:~#       /opt/dionaea/bin/python3.5       manage.py
collectstatic
root@dionaea:~# python manage.py migrate
root@dionaea:~# python manage.py runserver 0.0.0.0:8000
```

3. Ejecución

Como último paso queda iniciar el servicio de dionaea, añadiendo la siguiente línea de comandos en una terminal:

```
root@dionaea:~# cd /opt/dionaea/bin/
root@dionaea:~#./dionaea –L '*' –u dionaea –g users
```

Página intencionalmente en blanco

CAPÍTULO 6. RESULTADOS Y ANÁLISIS

""El bien que hemos hecho nos da una satisfacción interior, que es la más dulce de todas las pasiones."

René Descartes

Este capítulo presenta el análisis y discusión de los resultados obtenidos de este estudio. Los datos derivados de las tres herramientas de software de honeypot desplegadas se presentan y analizan en base a herramientas discutidas en el capítulo cuatro. La información estadística obtenida durante el proceso de evaluación de los dispositivos es presentada a través de las interfaces graficas de soporte discutidas en el capítulo anterior.

6.1 ANÁLISIS DE RESULTADOS HONEYD

En la figura siguiente, se muestra parte de la interfaz web de la herramienta de visualización Honeyd-Viz. Esta herramienta fue utilizada para generar estadísticas gráficas útiles que ayudaron a analizar los resultados obtenidos de la implementación de honeypot.

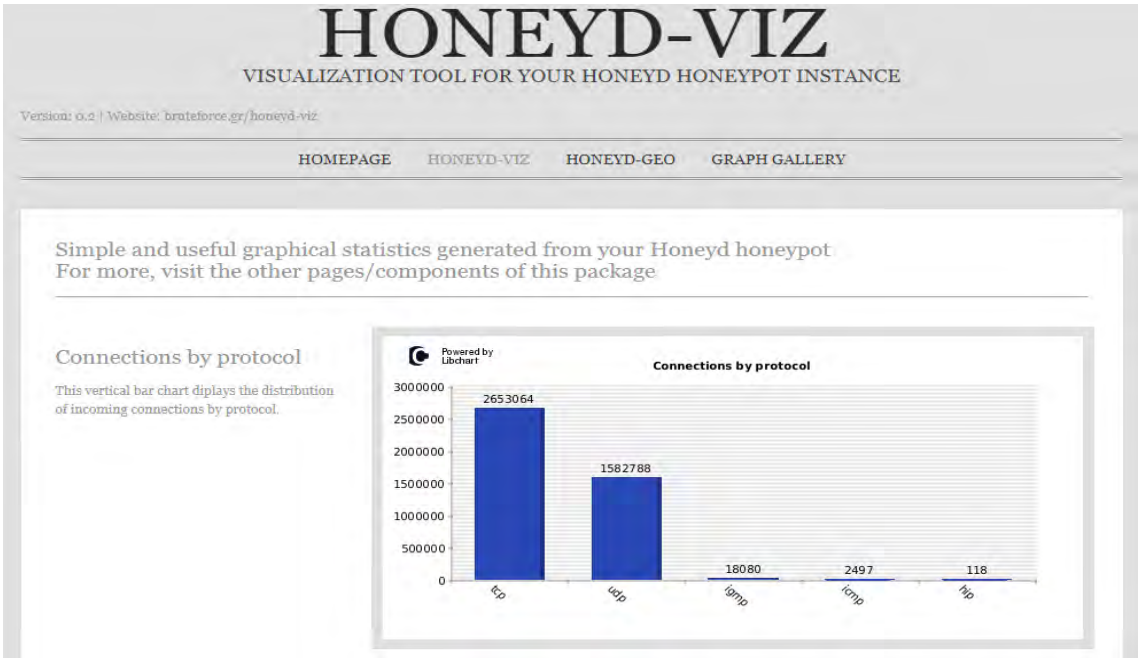


Figura 6.1 Interfaz Web de la herramienta de visualización Honey-Viz

Como se muestra en las siguientes figuras y como se esperaba, el mayor porcentaje de conexiones están relacionadas con el protocolo TCP. Esto se debe a dos razones:

En primer lugar, los puertos más frecuentemente hackeados son los TCP, donde se encuentran vulnerables los servicios populares a los que se accede a través de Internet, como FTP, SSH, TELNET, SMTP, etc.

En segundo lugar, debido a ello, la proporción de los puertos TCP dejados deliberadamente abiertos en la configuración de los honeypots virtuales es mucho mayor que la de los UDP, lo que lleva a una mayor posibilidad de conexiones TCP. El porcentaje más bajo pertenece a las conexiones IGMP, ICMP. Desde el punto de vista técnico, no sería preciso referirse a ellos como conexiones reales, ya que los paquetes ICMP no deben establecer conexiones, sino básicamente controlarlas a través de mensajes de estado y de error.

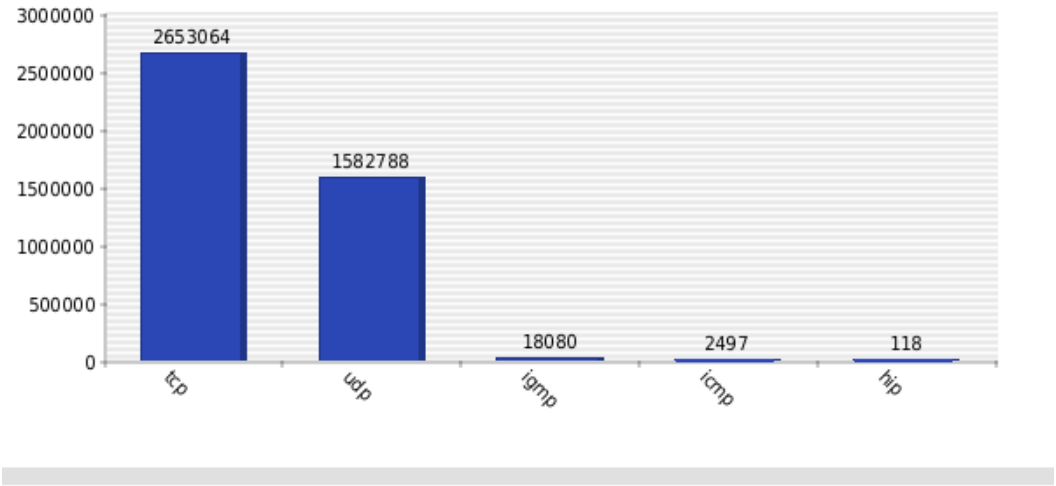


Figura 6.2 Grafico que muestra la distribución de las conexiones entrantes.

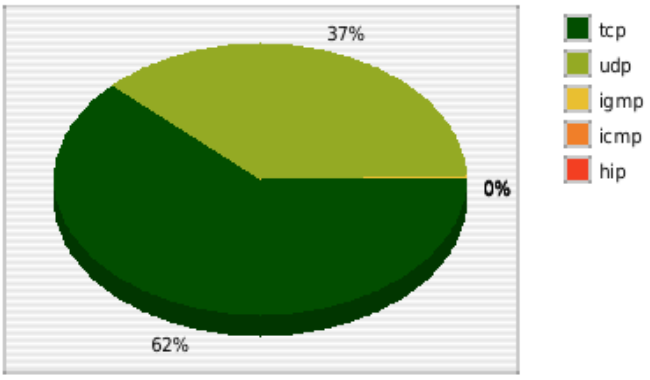


Figura 6.3 Grafico que muestra el porcentaje de conexiones entrantes.

Analizando más de cerca a las conexiones mencionadas anteriormente en las figuras 6.2 y 6.3 se van examinando cada uno de los tres tipos de protocolo. A continuación en la figura siguiente, se muestra un gráfico de barras verticales con las diez direcciones IP principales para establecer el mayor número de conexiones TCP con el sistema honeypot.

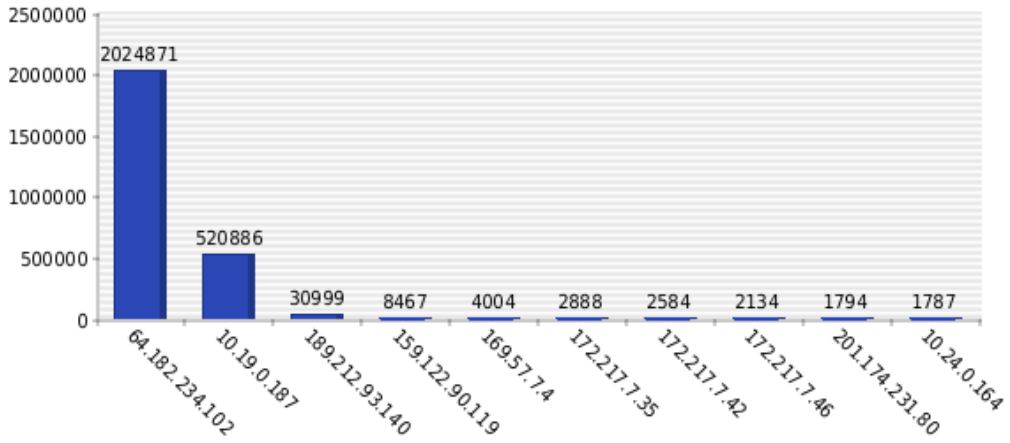


Figura 6.4 Grafico que muestra el número de diez conexiones TCP por IP.

En la figura siguiente, el gráfico de barras vertical muestra el número de conexiones de UDP por IP única. Como se hace evidente también de esta figura, las conexiones UDP son mucho menos en comparación con los TCP. Desde el gráfico, se puede observar que el porcentaje más alto de conexiones UDP proviene de una dirección IP que pertenece a la red local de honeypot. Específicamente, esta es la dirección IP del sistema en el que se está ejecutando la herramienta de software Kippo.

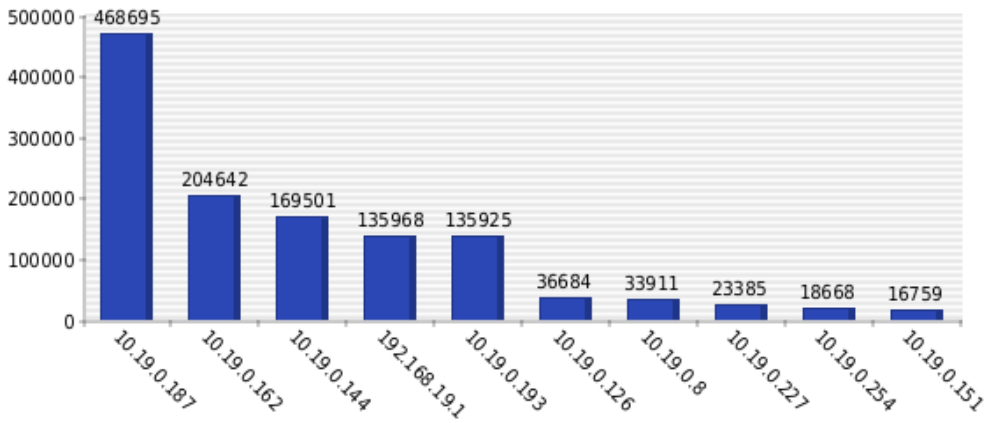


Figura 6.5 Grafico que muestra el número de diez conexiones UDP por IP.

El último tipo de conexiones examinadas es el protocolo relacionado con ICMP. Como se explicó anteriormente, el protocolo ICMP es utilizado por sistemas conectados a redes para enviar mensajes de error y de estado con respecto a las conexiones en lugar de intercambiar datos reales entre hosts. La siguiente figura muestra las 10 principales direcciones IP con la mayoría de las conexiones ICMP a los honeypots.

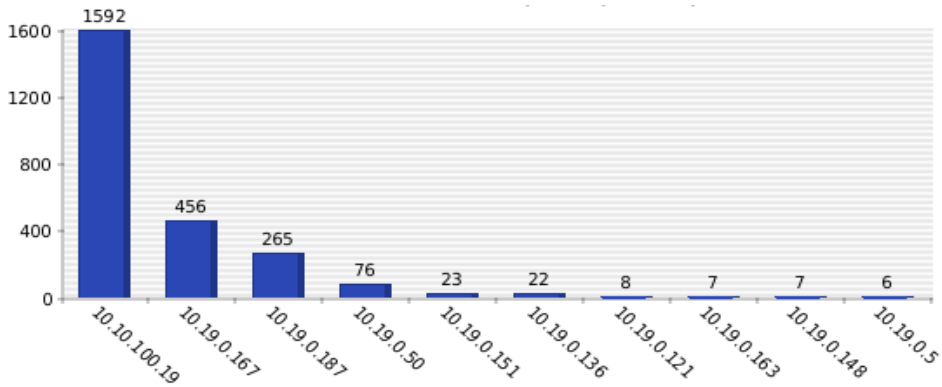


Figura 6.6 Grafico que muestra el número de diez conexiones ICMP por IP.

Después de examinar las estadísticas derivadas del tipo de conexiones por protocolo que se intentaron a los honeypots virtuales, sería bastante interesante examinar en total las conexiones recibidas por Honeyd con respecto a las direcciones IP de origen, el número de intentos de conexión por dirección IP, Países o regiones de origen, número de conexiones por país y las conclusiones que se extraer de estos datos. Las estadísticas incluidas en las siguientes cifras resumen la información antes mencionada tanto en barras verticales como en gráficos estadísticos de forma fácilmente percibida.

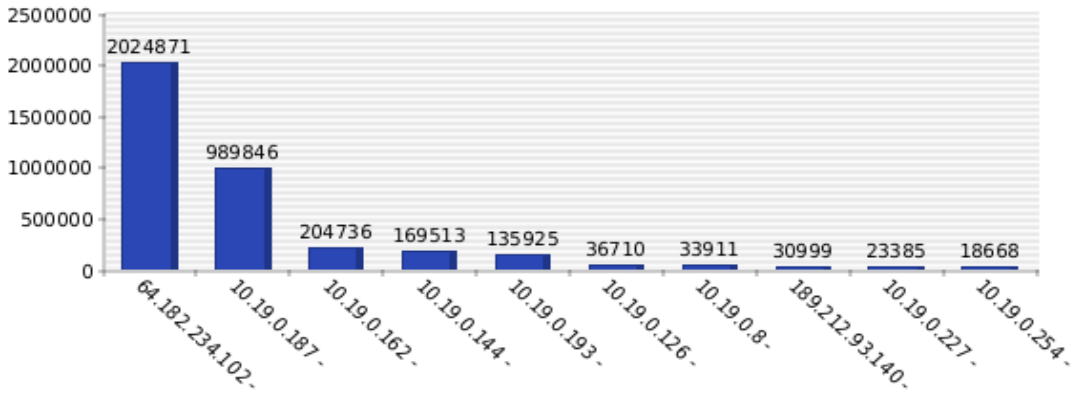


Figura 6.7 Grafico que muestra el número de conexiones IP a los principales países.

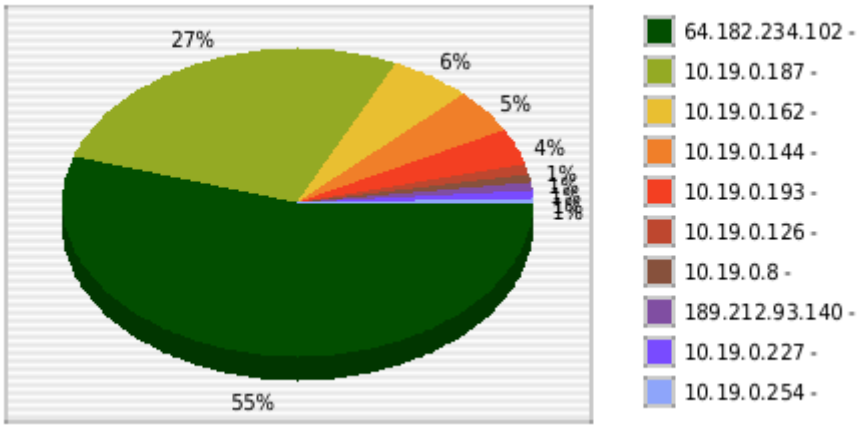


Figura 6.8 Grafico que muestra el porcentaje de conexiones IP de los principales países.

La siguiente tabla muestra las 10 principales direcciones IP conectadas a los honeypots virtuales del sistema Honeyd ordenados por volumen de conexiones y acompañados por geolocalización y otra información útil.

ID	IP Address	Probes	City	Region	Country Name	Code	Latitude	Longitude	Hostname	Lookup
1	64.182.234.102	2024871							64.182.234.102	
2	10.19.0.187	989846							10.19.0.187	
3	10.19.0.162	204736							10.19.0.162	
4	10.19.0.144	169513							10.19.0.144	
5	10.19.0.193	135925							10.19.0.193	
6	10.19.0.126	36710							10.19.0.126	
7	10.19.0.8	33911							10.19.0.8	
8	189.212.93.140	30999							189.212.93.140	
9	10.19.0.227	23385							10.19.0.227	
10	10.19.0.254	18668							10.19.0.254	

Figura 6.9 Tabla que muestra las 10 principales direcciones IP conectadas al Honeyd.

Un aspecto muy interesante del experimento es la frecuencia con la que los puertos de red son probados por hosts externos. Más específicamente, estarían extremadamente interesados en reconocer cuáles son estos puertos que aceptan la mayoría de los intentos de conexión. Saber cuáles son estos puertos es crucial ya que puede ayudar a los administradores de red a aplicar reglas sobre firewalls y otros sistemas relacionados con la seguridad de la red, restringiendo el acceso a los servicios que residen en estos puertos. Los piratas informáticos son muy conscientes de las vulnerabilidades de los puertos específicos y estos se convierten en sus primeros objetivos en su intento de entrar en un sistema. La siguiente figura muestra el número de conexiones por puerto de destino, prácticamente demostrando cuáles fueron los puertos más significativos durante el período de implementación de Honeyd. Es importante mencionar que las siguientes estadísticas no significan necesariamente que ha habido un acceso exitoso real, sino que se basan principalmente en intentos no autorizados.

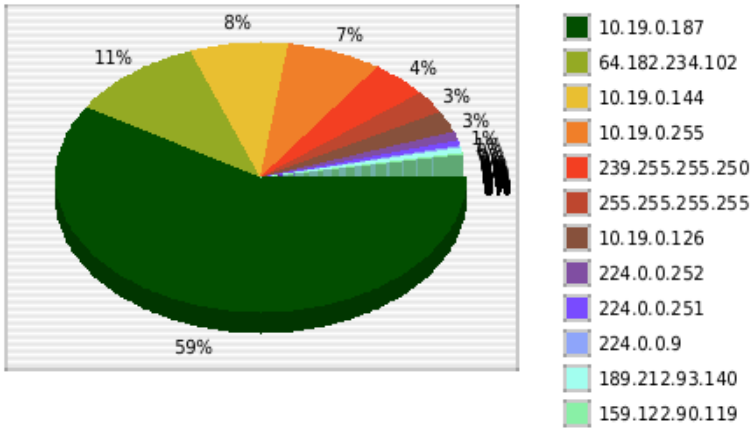


Figura 6.10 Grafico que muestra el porcentaje de conexiones por puerto de destino.

A pesar del tiempo de despliegue de honeypot, se puede ver que las estadísticas con respecto a los puertos atacados son bastante similares a las esperadas según las investigaciones que se han llevado a cabo. Ya que las tres primeras posiciones pertenecen a los mismos puertos y las restantes son, con algunas excepciones, casi idénticas. Finalmente, las últimas estadísticas dadas son las relativas al número de conexiones por IP de destino.

6.2 ANÁLISIS DE RESULTADOS KIPPO

A continuación, parte de la interfaz web de Kippo-Graph de la implementación de Kippo. En esta figura se puede tener una visión general de los eventos que han tenido lugar en este periodo de pruebas. Los intentos de inicio de sesión totales contra el servidor SSH honeypot fueron 15 de 7 IPs diferentes. Enfrento el primer ataque pocas horas después de la instalación del honeypot y cómo podemos ver incluso el último día del experimento, los ataques todavía estaban ocurriendo. Es realmente útil examinar el número de inicios de sesión con respecto a los diferentes IPs, ya que debido a ataques de fuerza bruta el número total de intentos de inicio de sesión es difícil. Como se verá más adelante en el análisis de los resultados de Kippo, Kippo-Graph proporciona gráficos útiles para detectar este tipo de ataques.

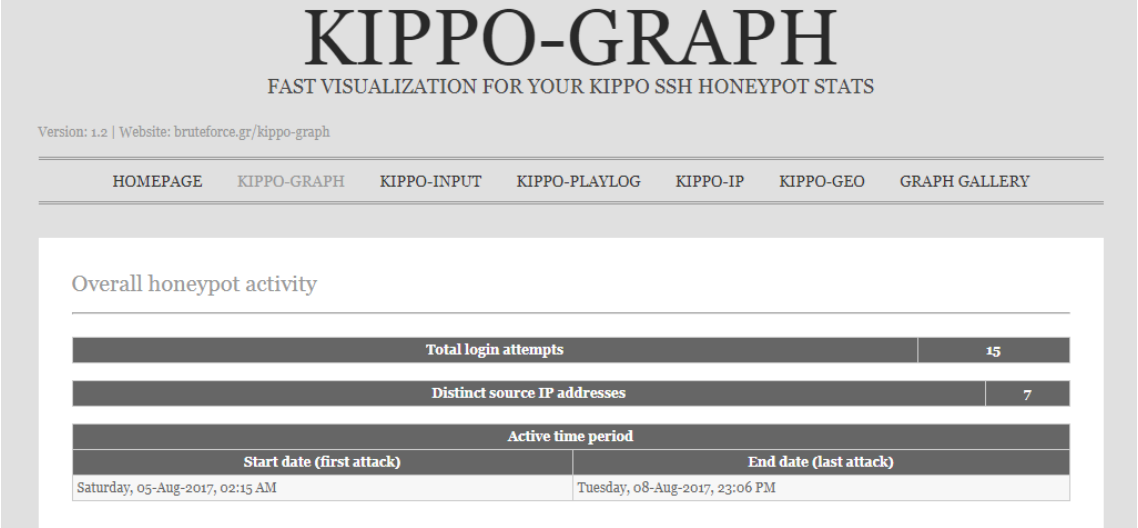


Figura 6.11 Visión general de los resultaos de Kippo.

El ratio de inicio de sesión de éxito general se muestra en la siguiente figura. Del total de 45 intentos de inicio de sesión, sólo 6 tuvieron éxito, lo que significa que el atacante encontró el nombre de usuario y la contraseña correctos. La mayoría - 39 intentos de inicio de sesión - no tuvieron éxito. Hasta ahora, al ver estas dos cifras, una posible pregunta podría ser la razón por la cual, aunque ponemos una vulnerabilidad a los ataques de diccionario cuenta de inicio de sesión, los intentos de éxito fueron inferiores al 0,5%.

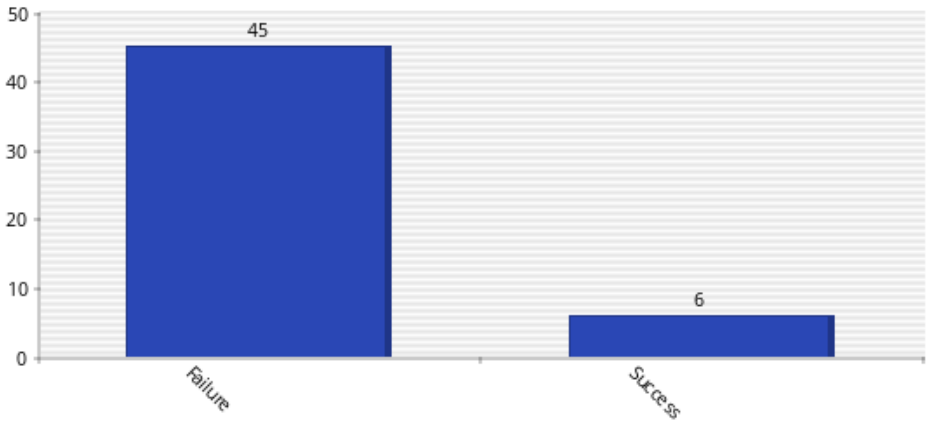
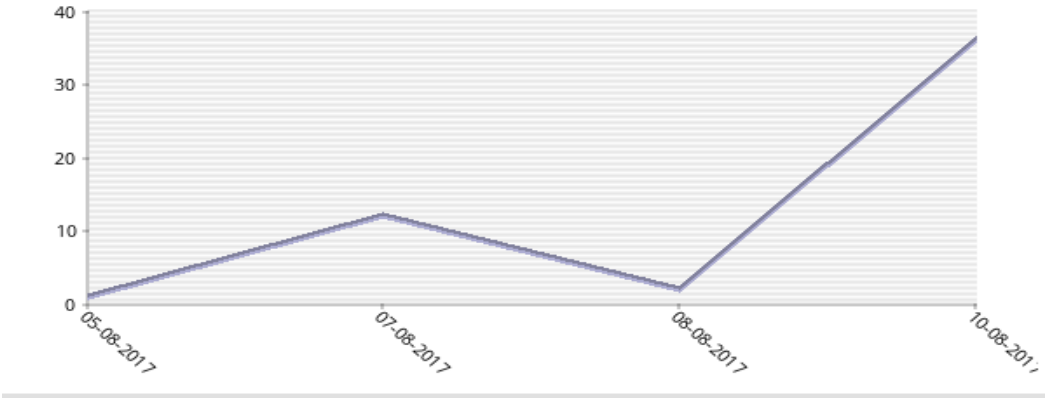


Figura 6.12 Índice de éxito global.

En las siguientes figuras se presentan el número total de sondas y los intentos exitosos por día respectivamente contra el servidor SSH. Además, Kippo-Graph nos proporciona la misma información por semana. Este último podría ser útil para experimentos a largo

plazo. Se puede ver específicamente la rápida fluctuación del gráfico. Por otra parte, los inicios de sesión exitosos aumentaron como se esperaba debido al gran número de sondas que condujeron probablemente a encontrar los detalles de inicio de sesión. Es interesante en ese punto debido al análisis hecho ya que permite ver las IPs de estos ataques que se originan. Así podemos tener una visión más completa de lo que sucedió exactamente.



30Figura 6.13 Sondas por día.

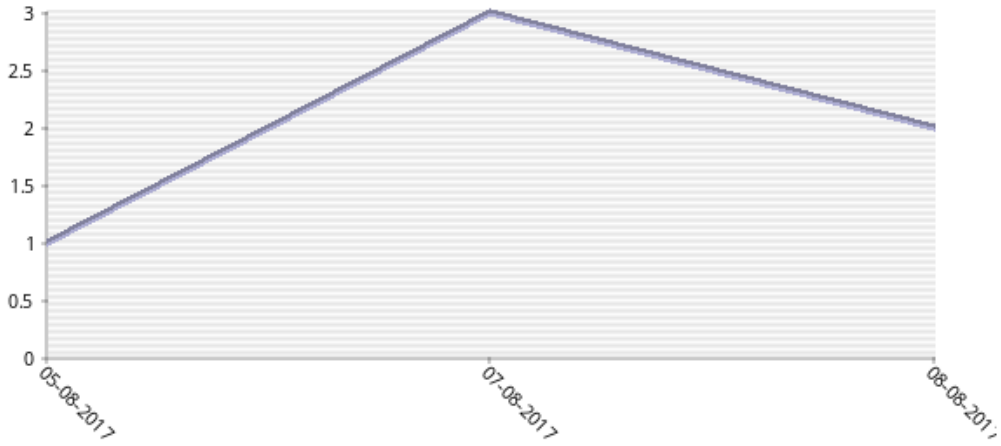


Figura 6.14 Éxito por día.

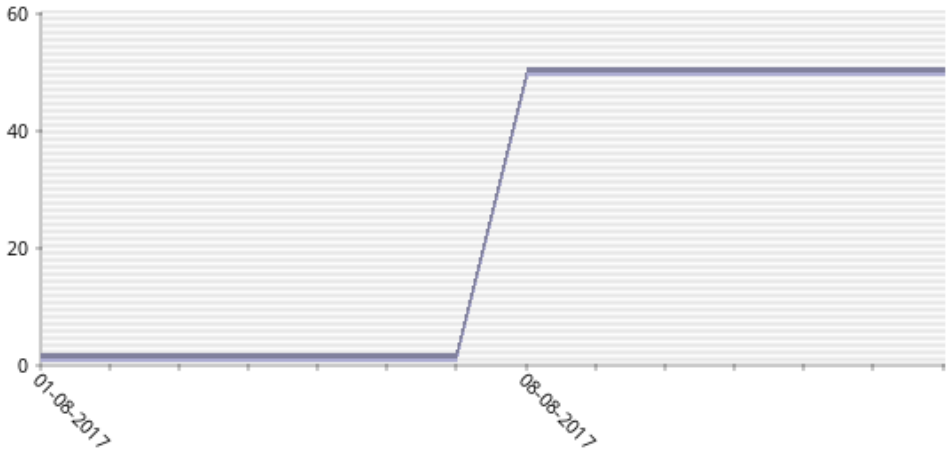


Figura 6.15 Sondas por semana.

La Figura 6.16 muestra la IP de los 10 atacantes más activos. Como podemos ver, la mitad de los intentos totales de inicio de sesión fueron generados por el mismo atacante. La figura 6.17 representa las mismas estadísticas en un gráfico circular y con los códigos de país para las IP correspondientes. Además, en la Figura 6.18 las IPs que se registraron con éxito más de una vez en el servidor SSH.

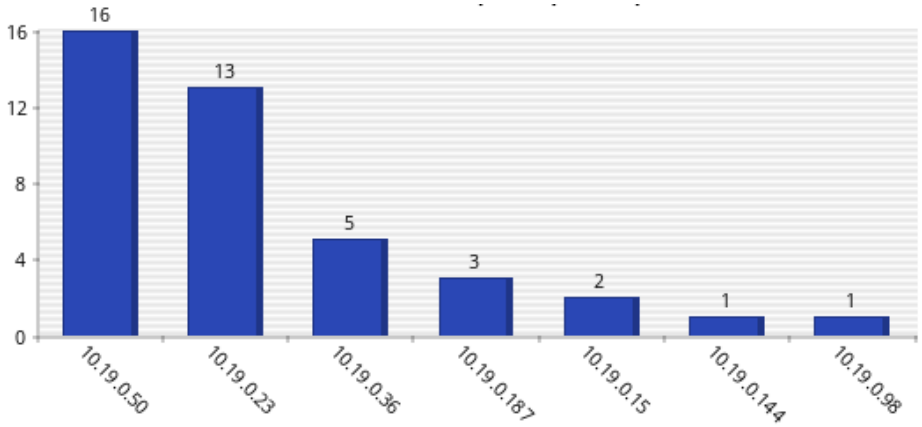


Figura 6.16 Sondas por semana.

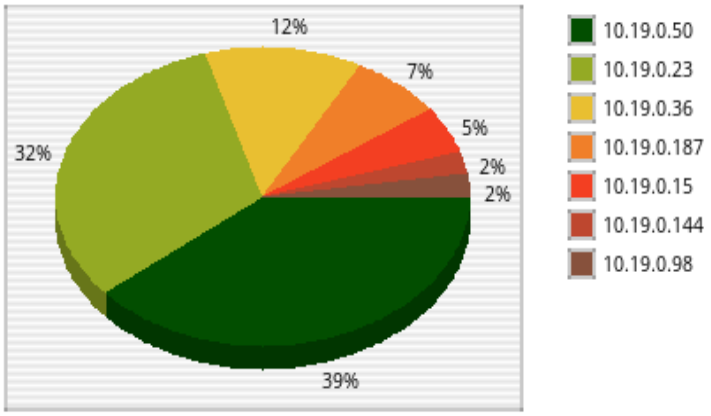


Figura 6.17 Sondas por semana.

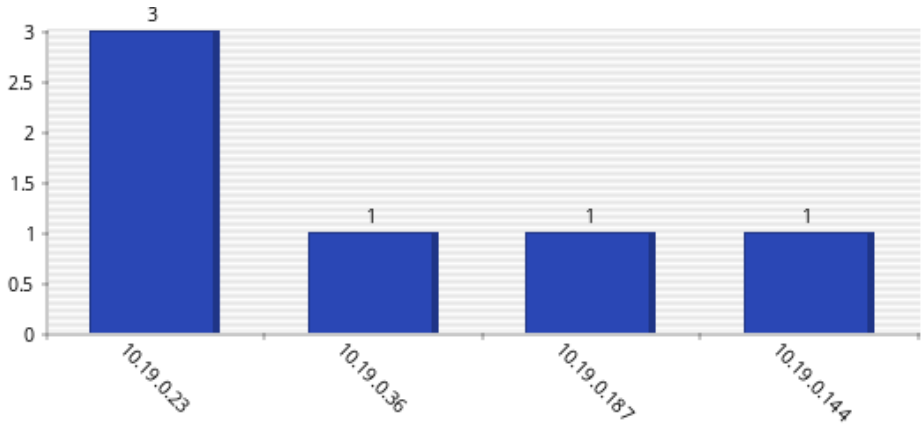


Figura 6.18 Top 20 de logging con éxito Kippo.

En esta sección analizaremos las credenciales de inicio de sesión que se han utilizado en los ataques contra el honeypot. En las siguientes figuras 6.19 y 6.20, se demuestran los nombres de usuario y contraseñas más usados. Como se esperaba del análisis teórico sobre los ataques de SSH, la raíz de nombre de usuario fue la más utilizada por los atacantes, mientras que el resto de los nombres de usuario son ligeramente diferentes.

En la Figura 6.20, las contraseñas utilizadas son casi las mismas que se esperaban, es decir, variaciones de 'admin' y '123456' entre la parte superior utilizada. Kippo-Graph también proporciona estadísticas sobre las 10 combinaciones de nombre de usuario y contraseña que Kippo grabó. Dos gráficos en las Figuras 6.21 y 6.22, donde en la primera

se puede ver los números de las combinaciones utilizadas y también los porcentajes en forma de gráfico circular.

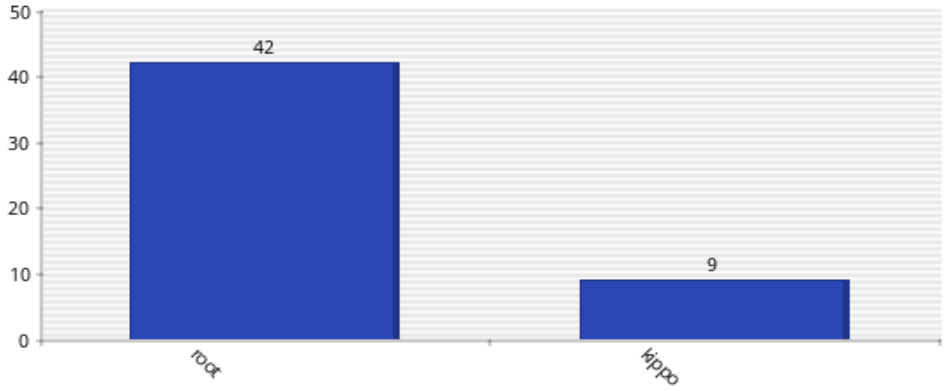


Figura 6.19 Top 10 de usenames intentados con Kippo.

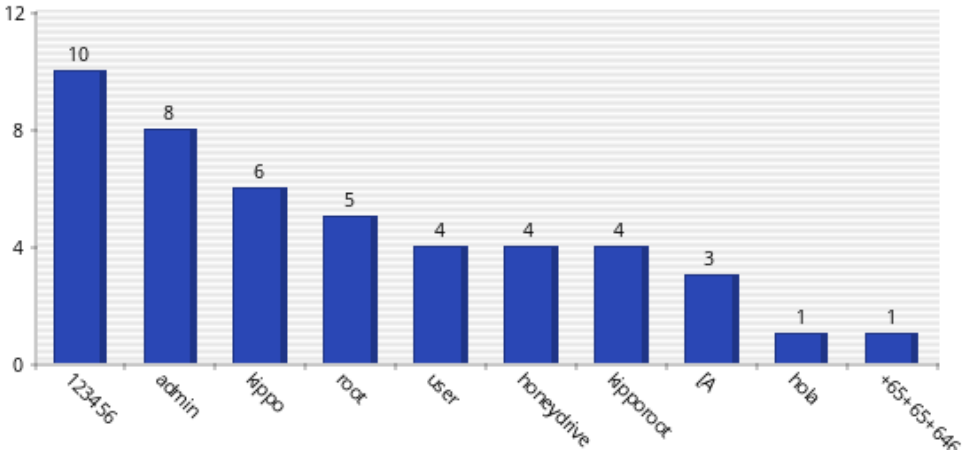


Figura 6.20 Top 10 de password intentados con Kippo.

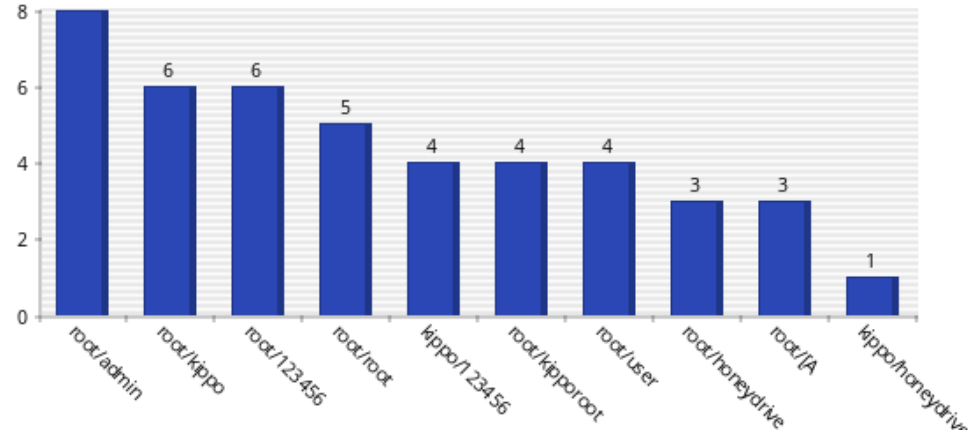


Figura 6.21 Top 10 de combinaciones username/password para Kippo.

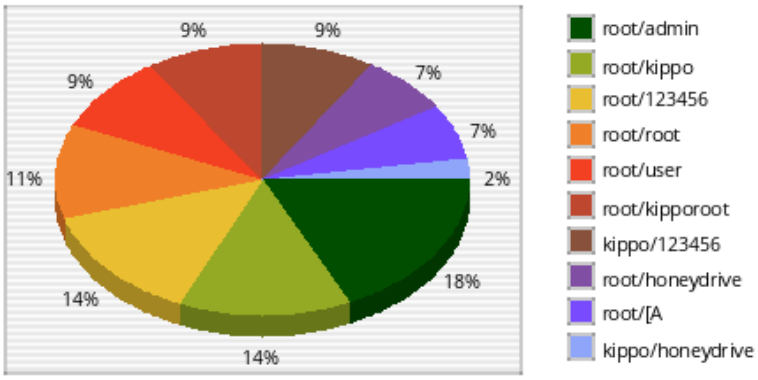


Figura 6.22 Top 10 de combinaciones username/password para Kippo.

Como se ha mencionado anteriormente, Kippo nos da la oportunidad de reproducir a través de archivos de registro TTY todos los ataques que se han producido. Es fascinante ver cómo ocurre un ataque real, pero no sería útil si no se tiene una visión general de los comandos utilizados y las actividades de todos los atacantes en general. La segunda sección de Kippo-Graph da esta información en particular.

Input presentation and statistics gathered from the honeypot system

Overall post-compromise activity

Post-compromise human activity	
Total number of commands	**Distinct number of commands**
27	15

Downloaded files	
Total number of downloads	**Distinct number of downloads**
0	0

Figura 6.23 Resumen de actividad de interacción.

Dos cartas más interesantes se muestran en las figuras 6.24 y 6.25. La primera figura muestra el número de comandos por día, mientras que la última muestra los comandos más utilizados por los atacantes. Comando - w, en la parte superior de esta lista, muestra quién está conectado al sistema y su actividad. Además, el comando `uname` da información sobre el sistema operativo, la versión del kernel, etc. En este caso, uname con `flag -a`, da toda la información posible del sistema.

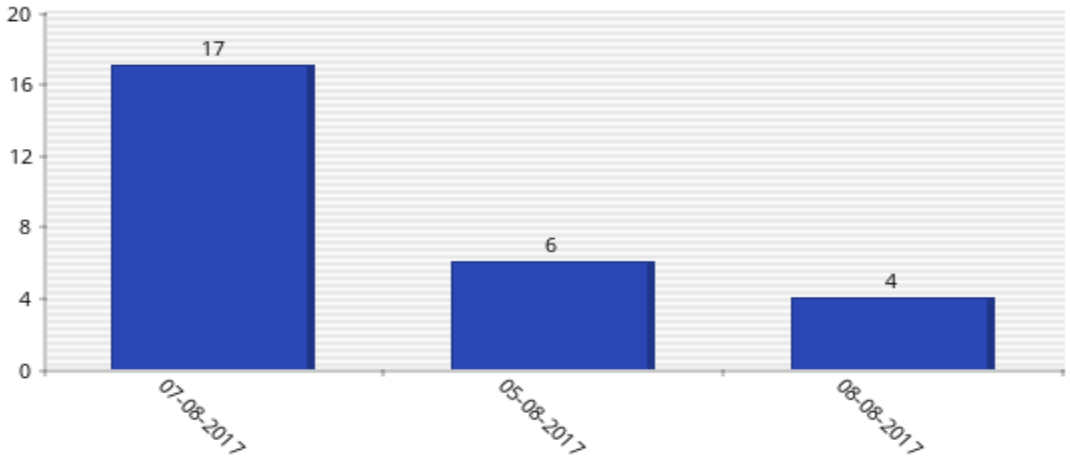


Figura 6.24 Días de mayor actividad humana.

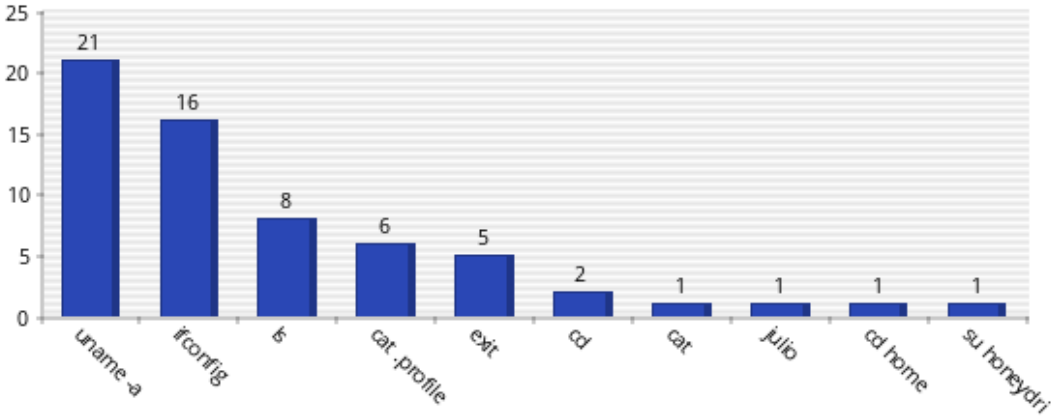


Figura 6.25 Top de los 10 principales comandos de entrada.

A continuación, describiremos un interesante ataque que Kippo grabó y reprodujimos usando los archivos de registro. Inicialmente, el atacante quería explotar nuestro sistema usando el comando `cat /etc/issue`. Después de eso, descargó e instaló la aplicación de pantalla y trató de crear un nuevo usuario con el comando adduser. El atacante no pudo crear con éxito un usuario, por lo que intentó editar el archivo `/etc/passwd` que contiene información sobre los usuarios. Por esa razón, instaló las aplicaciones `vim` y `nano`, ya que no estaban instaladas. Kippo muestra que las aplicaciones se han instalado correctamente. El atacante probó diferentes métodos como actualizar los paquetes, ya que Kippo respondió con errores a comandos nano o vi. Eventualmente, después de todos estos comandos sin éxito, el atacante escribió `rm -rf /*` para eliminar todos los

archivos y salir del sistema. Por último, Kippo-Graph nos proporciona estadísticas de las herramientas de software cliente SSH que se han utilizado y también una útil tabla con las 10 principales direcciones IP conectadas al honeypot. La Figura muestra esta tabla donde se proporciona información detallada sobre la posición geográfica de los atacantes y además comprobar directamente si la IP está en la lista negra o no, haciendo clic en los enlaces en el área de búsqueda.

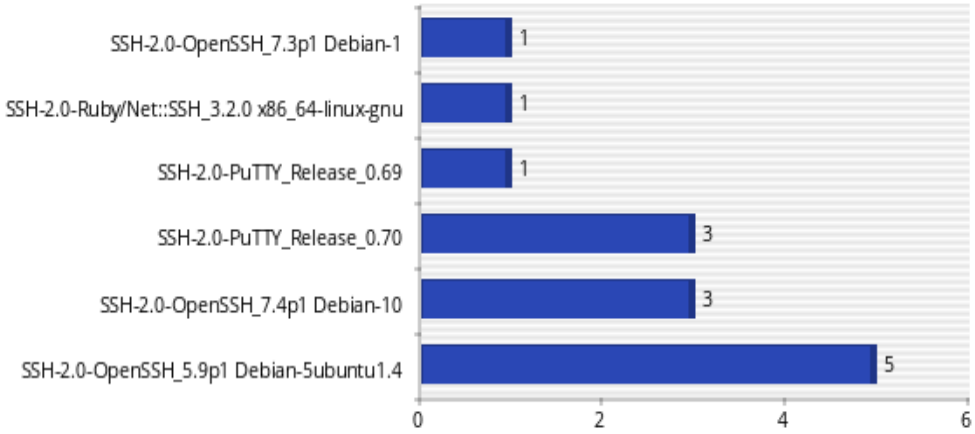


Figura 6.26 Clientes SSH utilizados.

Geolocation information gathered from the top 10 IP addresses probing the system

The following table displays the top 10 IP addresses connected to the system (ordered by volume of connections).

ID	IP Address	Probes	City	Region	Country Name	Code	Latitude	Longitude	Hostname	Lookup
1	10.19.0.50	16							10.19.0.50	
2	10.19.0.23	13							10.19.0.23	
3	10.19.0.36	5							10.19.0.36	
4	10.19.0.187	3							10.19.0.187	
5	10.19.0.15	2							10.19.0.15	
6	10.19.0.98	1							10.19.0.98	
7	10.19.0.199	1							10.19.0.199	
8	10.19.0.144	1							10.19.0.144	

Figura 6.27 Direcciones IP e información gráfica de los atacantes.

6.2 ANÁLISIS DE RESULTADOS DIONAEA

En la siguiente captura de pantalla, se representa la página principal de DionaeaFR, con el resumen de la operación Dionaea. Para la implementación, el número total de conexiones fue de 13.595 y de 431 IP diferentes. Dionaea capturó 8 malware de 5 URLs diferentes, pero como no elegimos enviar estos malware a terceros, estos malware no fueron analizados y por lo tanto no se conocen, como se muestra a continuación. Además, la página de inicio nos proporciona información sobre las conexiones e IPs por país.

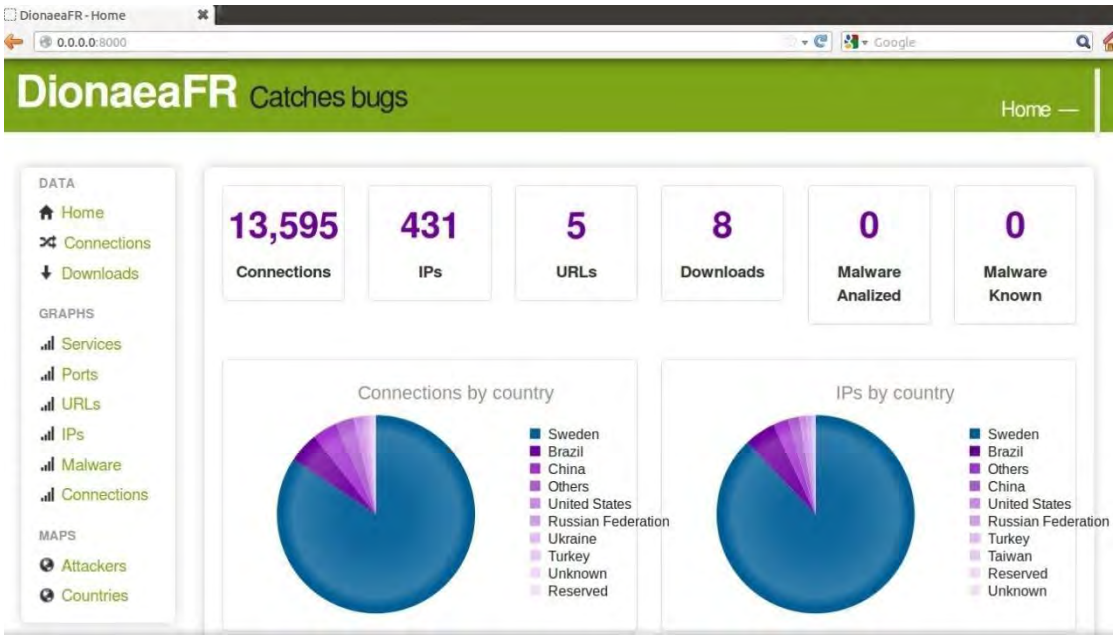


Figura 6.28 Página principal de DionaeaFR.

DionaeaFR proporciona información importante sobre las conexiones. Una captura de pantalla de los datos visualizados. Podemos ver en detalle todas las conexiones que se han aceptado. Además, al hacer clic en un identificador de conexión específico, se ve más detalladamente la conexión. Por ejemplo, se puede ver la conexión con id 13573. En ese ejemplo, Dionaea aceptó una conexión TCP en el puerto 1433 en el servicio MS-SQL. El atacante intentó conectarse con el nombre de usuario 'sa' y la contraseña en blanco tal y como se muestra en el área de inicio de sesión. DionaeaFR da una descripción detallada de la dirección IP del atacante, la distancia y también la huella digital MS-SQL. Se puede tener la misma información detallada sobre cada conexión.

ID	State	Protocol	Service	Root	Parent	Sensor	DST Port	Attacker	Hostname	SRC Port
13565	reject	tcp	pcap	13565	—	194.47.18.195	5900	117.79.91.203	—	40634
13564	reject	tcp	pcap	13564	—	194.47.18.195	3389	198.100.99.136	—	6000
13563	reject	tcp	pcap	13563	—	194.47.18.195	3389	198.100.99.136	—	6000
13562	reject	tcp	pcap	13562	—	194.47.18.195	23	83.109.211.184	—	2091
13561	reject	tcp	pcap	13561	—	194.47.18.195	23	219.70.42.61	—	48861
13560	connect	udp	SipSession	13560	—	194.47.18.195	5060	192.157.196.102	—	5273
13559	connect	udp	SipSession	13559	—	194.47.18.195	5060	192.157.196.102	—	5097
13558	accept	tcp	smbd	13558	—	194.47.18.195	445	188.55.91.109	—	1167
13557	reject	tcp	pcap	13557	—	194.47.18.195	22	210.152.137.101	—	54245
13556	accept	tcp	httpd	13556	—	194.47.18.195	80	37.8.27.209	—	25083
13555	accept	tcp	httpd	13555	—	194.47.18.195	80	37.8.27.209	—	26230
13554	connect	udp	SipSession	13554	—	194.47.18.195	5060	173.242.117.176	—	5127
13553	connect	udp	SipSession	13553	—	194.47.18.195	5060	199.180.116.158	—	5083
13552	connect	udp	SipSession	13552	—	194.47.18.195	5060	202.155.233.190	—	5061
13551	accept	tcp	epmapper	13551	—	194.47.18.195	135	4.234.252.241	—	3535

Previous Next

Figura 6.29 Lista de conexiones.

Las cifras anteriores proporcionan información sobre todos los datos registrados por Dionaea, lo que significa que se calcularon los tres tipos de conexión (aceptar, rechazar, conectar). Es interesante para este análisis echar un vistazo a las conexiones de "aceptar" o "conectar" solo que por simplicidad se denominan conexiones válidas. Para ello se elimina de la base de datos SQLite las conexiones 'rechazar' antes de visualizar los datos con DionaeaFR. En la siguiente figura se ven los resultados.

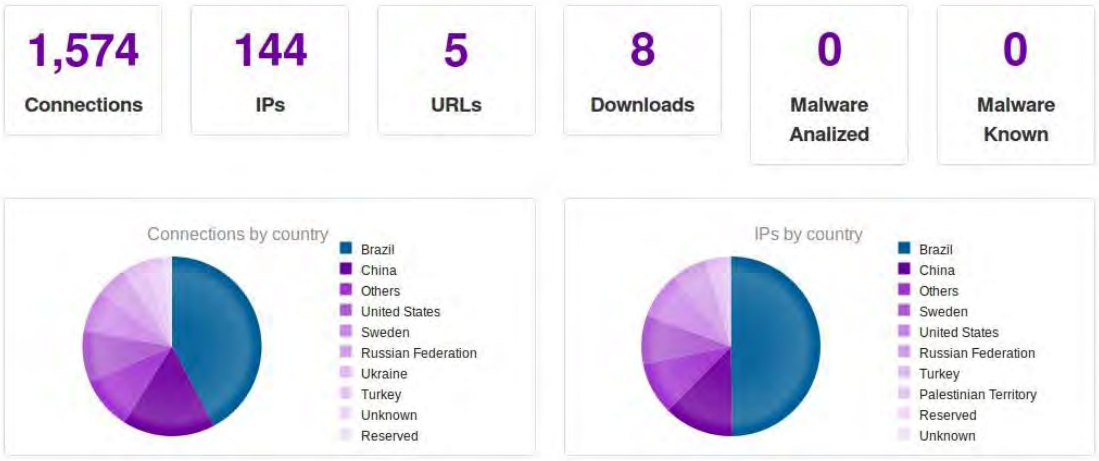


Figura 6.30 Resumen de conexiones Dionaea.

Hasta el momento, los datos recogidos por Dionaea eran bastante satisfactorios, en contraste con el breve período de experimento: casi 1.600 conexiones válidas de 144 IPs únicas y además 8 malware descargables, que se analizará más adelante en la siguiente sección.

En ese momento en un análisis más profundo a los gráficos proporcionados por DionaeaFR para detectar exactamente lo que ha ocurrido durante este período.

Para una mejor y más clara visión de los resultados, en las siguientes figuras mostramos las gráficas de DionaeaFR, ambas con todos los tipos de conexiones y con conexiones exclusivamente válidas (aceptar, conectar).

Connection	URL	MD5
13260	http://146.185.246.86/aa.exe	0092eae69b458ccf4c641debc8ba23b0
13159	tftp://194.27.76.22/ssms.exe	1f8a826b2ae94daa78f6542ad4ef173b
12276	http://146.185.246.86/aa.exe	30edf62eec7dcbc4688a7fe618077dad
11998	tftp://194.38.117.57/upds.exe	c5306102e449f83fcdc12a025eb59333
11691	smb://110.36.230.34	4d56562a6019c05c592b9681e9ca2737
11691	smb://110.36.230.34	ad7a6105590ce45b53520cfc1e4a0b65
11613	tftp://194.38.117.172/upds.exe	eb9905189a73ffd0cdadc76cd75fe9ac
11570	tftp://194.27.76.22/ssms.exe	1f8a826b2ae94daa78f6542ad4ef173b

Figura 6.31 Descarga MD5.

Específicamente en la figura 6.32, el número de conexiones por servicio se representa en el gráfico superior y el número de conexiones por puerto en el segundo gráfico durante los últimos siete días. La figura 6.32 es un resumen de todas las conexiones Dionaea. Como se puede ver, MS-SQL y el puerto 1433 estaban en la parte superior de los intentos conectados, seguidos por el servicio MySQL en el puerto 3306. Los puertos 135 y 445 atrajeron aproximadamente 200 conexiones. El servicio pcap mantiene información sobre las conexiones rechazadas. Esa es la razón por la que se enumera en la parte superior de los servicios conectados.

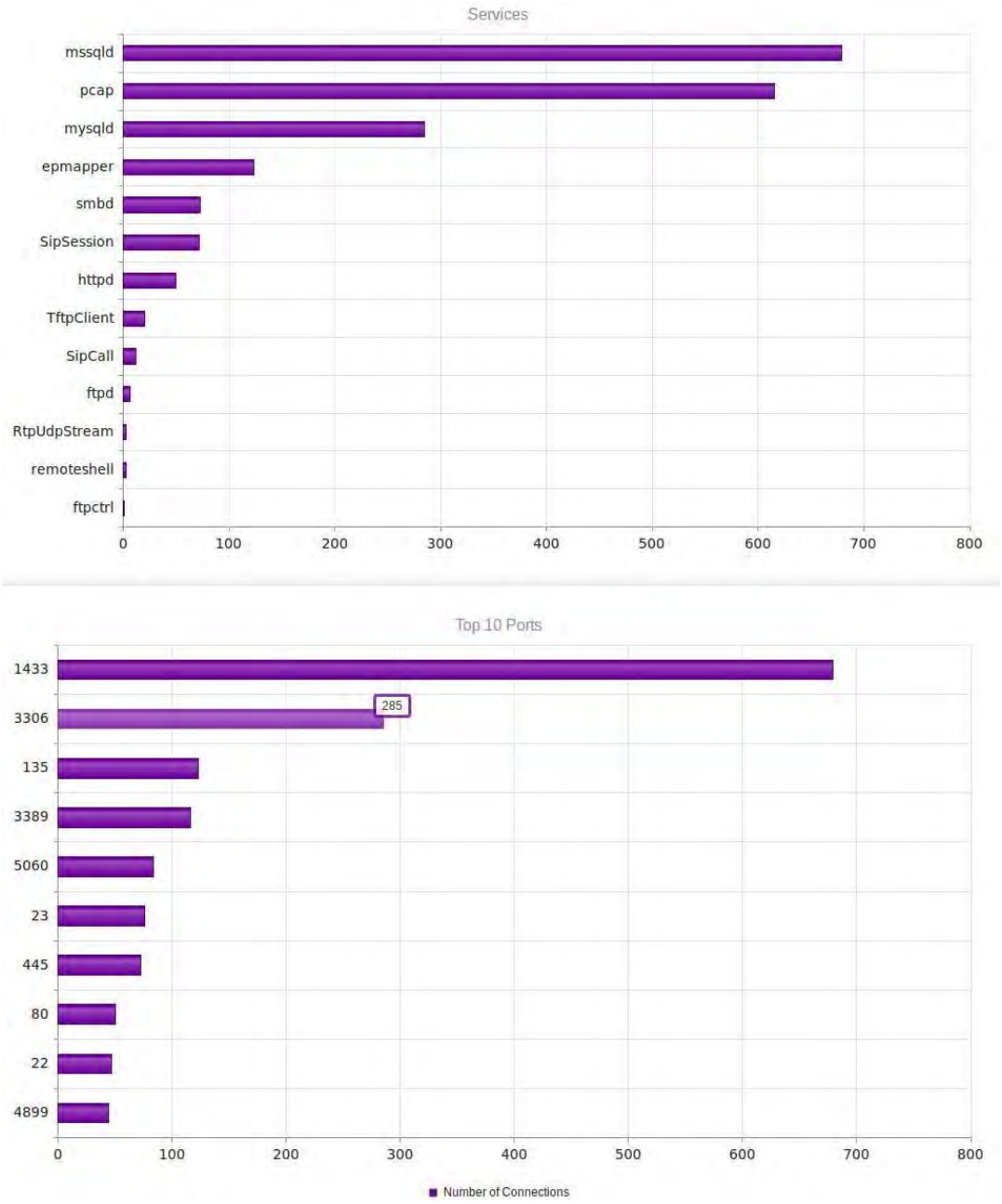


Figura 6.32 Conexiones por servicios y puertos.

La figura 6.33 representa las conexiones por servicios y puertos para las conexiones válidas, durante los últimos días. En la figura, al eliminar las conexiones rechazadas podemos ver todos los puertos en los que Dionaea escucha.

Los resultados hasta ahora estaban en contraste con los esperados del análisis teórico. Como las investigaciones han demostrado, el puerto 445 se espera que esté en la parte superior de los puertos atacados. Además, DionaeaFR muestra estadísticas de los últimos siete días, lo que reduce aún más la duración de este estudio.

En la siguiente sección donde se realiza consultas a la base de datos SQLite, podemos ver un resumen total de lo que sucedió con cada uno de los protocolos y su actividad reciente.

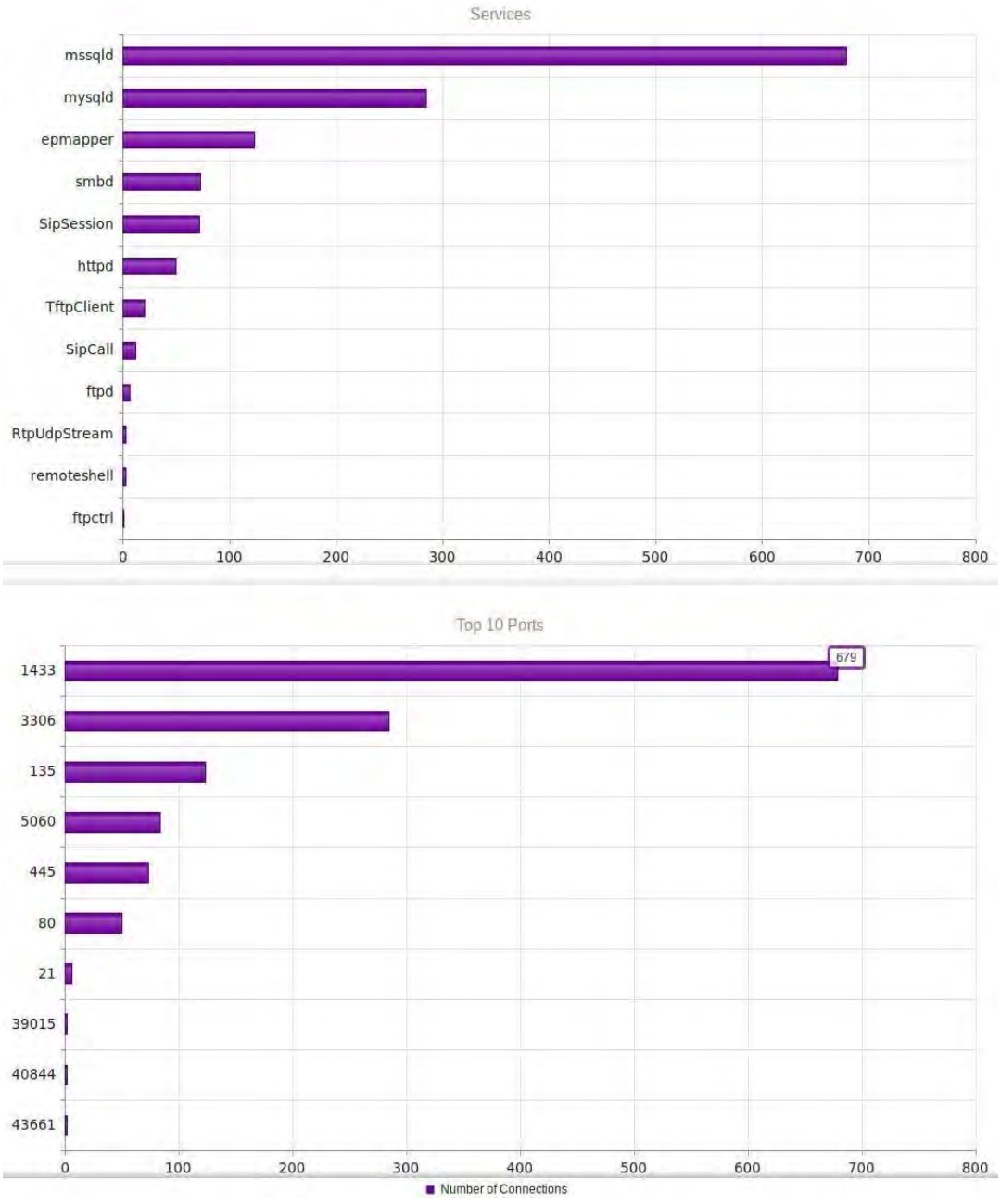


Figura 6.33 Conexiones por servicios y puertos (‘accept’,’connect’only).

DionaeaFR proporciona también información sobre el país del atacante y la dirección IP. Esto se hace a través de la sección de mapas y se muestra en las figuras 6.34 y 6.35, donde se ve la posición del atacante presentada por punto y el país del atacante, respectivamente.

Además, en la figura 6.35, los países más activos se presentan con un color púrpura más oscuro.

Figura 6.34 Mapa con la posición de atacantes.

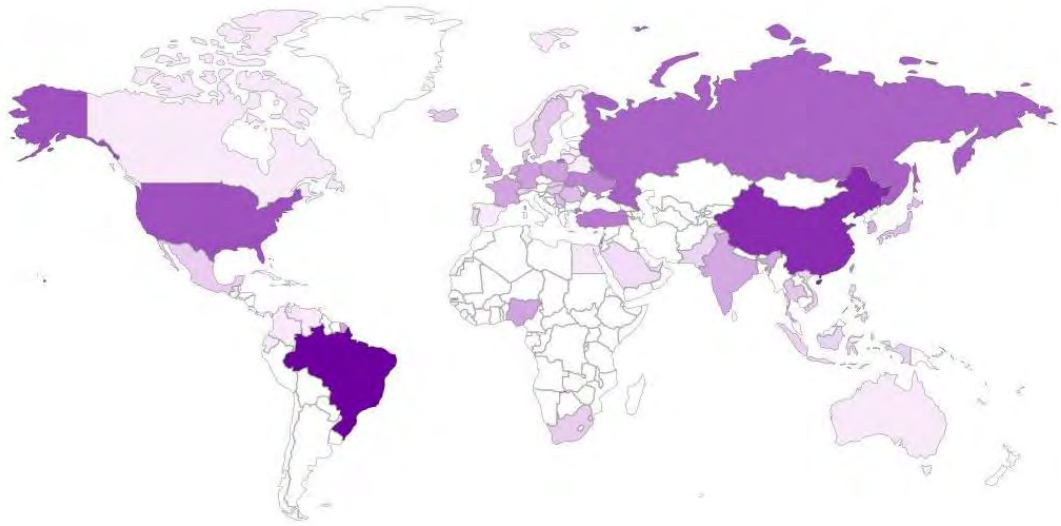


Figura 6.35 Países mayormente activos.

En esta sección, se recupera información útil directamente del archivo de registro de Dionaea, consultando la base de datos SQL. Para el análisis se usan scripts proporcionados por la página de inicio de Dionaea. La herramienta de visualización de la sección anterior, no podía darnos un panorama total de los puertos atacados, ya que sólo

podíamos ver los resultados de los últimos días. Para esa información hemos consultado la base de datos SQL y los resultados se muestran en la figura 6.36. La mayoría del puerto atacado ha sido el puerto 1433 con un hitcount de 686 conexiones aceptadas, seguido por el puerto 3306 (servicio MySQL) y el puerto 135 con 169 conexiones aceptadas.

	hitcount	port
1	11	21
2	23	42
3	97	80
4	169	135
5	85	445
6	686	1433
7	319	3306

Figura 6.36 Puertos más atacados.

Con esto se ve la forma en la que actuó el sistema honeypot Dionaea ante pruebas de testeo e intrusión de tráfico por medio de herramientas mencionadas en capítulos anteriores.

CAPÍTULO 7. CONCLUSIONES Y TRABAJO A FUTURO

"Para empezar un gran proyecto, hace falta valentía, Para terminar un gran proyecto, hace falta perseverancia."

Anónimo

7.1 CONCLUSIONES

La implementación de sistemas de seguridad en redes informáticas es de gran importancia en todo tipo de instituciones u organizaciones, todo con el objetivo de mantener la integridad, confidencialidad, y disponibilidad de la información.

En este proyecto se analizaron diferentes honeypots instalados en un entorno virtualizado. Con el fin de capturar diferentes tipos de ataques contra esta red y monitorear eficientemente las actividades de los atacantes, se implementaron tres diferentes herramientas de software de código abierto de honeypot.

El principal objetivo ha sido detectar y registrar las actividades maliciosas dirigidas a la red, para obtener la mayor información posible sobre los atacantes, los métodos que utilizan para entrar en los sistemas y las vulnerabilidades de la red.

La primera opción ha sido Honeyd, una potente herramienta de honeypot que permitió implementar una plataforma de sistemas operativos virtuales de la manera más efectiva y de bajo costo. Los diferentes sistemas virtualizados desplegados por Honeyd en combinación con los puertos que quedaron abiertos y los servicios emulados, fueron elegidos cuidadosamente para llamar la atención de los atacantes.

Uno de los tipos más comunes de ataque contra las redes de campus, indicado por la investigación relacionada, tiene que ver con el malware. Por esta razón, la segunda opción ha sido la herramienta de honeypot Dionaea, un coleccionador de malware. La creación de Dionaea en esta implementación aumentó las posibilidades de detectar y capturar ataques además de brindar la oportunidad de descargar copias del malware detectado.

Esta característica es de gran importancia ya que fue capaz de analizar en un entorno seguro el propósito del malware.

Kippo ha sido la tercera elección de honeypot. Kippo demostró ser una gran herramienta estadística con respecto a las credenciales de inicio de sesión intentadas. Además, Kippo nos proporcionó información útil sobre las intenciones de los atacantes a través de los comandos de entrada ingresados. Para el análisis de datos, se centró en los resultados de las herramientas de visualización especializados para cada herramienta de honeypot.

Además, también se utilizaron los archivos de registro generados por los honeypots y las bases de datos correspondientes. Las herramientas de visualización utilizadas han sido Honeyd-Viz para Honeyd y DionaeaFR junto con Kippo-Graph para Dionaea y Kippo respectivamente. Se ha establecido este experimento por un período de dos semanas. Todos los honeypots han sido accesibles desde el exterior, ya que se asignaron direcciones IP públicas a cada uno de ellos y se asignaron en la zona perimetral de la infraestructura del campus. Los resultados han sido interesantes dada la corta duración del proyecto.

Concluyendo, los honeypots demostraron ser un gran mecanismo de seguridad. La variedad de los datos recopilados puede ayudar en gran medida a fortalecer las políticas de seguridad dentro de la red del campus universitario. En pocas palabras, los honeypots virtuales, se pueden configurar y poner en acción, ya que constituyen un medio de primera clase para obtener un profundo conocimiento de la red en un mundo real, pero al mismo tiempo de bajo riesgo.

REFERENCIAS

[1] J. W. a. Z. A. Kissel, Introduction to Network Security: Theory and Practice, Wiley, 2015.

[2] N. Krawetz, Anti-Honeypot Technology, IEEE Security & Privacy, 2004.

[3] L. Spitzner, Honeypots: tracking Hackers, Pearson Education, 2003.

[4] Enisa, Proactive Detection of Security Incidents, CERT Polska, 2012.

[5] K.Kendall, A Database of Computer Attacks for the Evaluation of Instrusion Detection Systems, Massachusetts Institute of Technology Master´s Thesis, 1998.

[6] R. A. Grimes, Honeypots for Windows, Springer-Verlag New York, Inc, 2005.

[7] N. P. a. T. Holz, Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Pearson Education, Inc, 2008.

[8] T. H. Proyect, Know your Enemy: Learning About Security Threats (Second Edition), Pearson Education, Inc, 2004.

[9] S. M. Bellovin, Thinking Security: Stopping Next Year's Hackers, Addison-Wesley Professional Computing Series, 2015.

[10] N. Mansfield, Practical TCP/IP - Designing, using, and troubleshouting TCP/IP networks on Linux and Windows, Pearson Education, 2003.

[11] S. B. a. M. Kabay, Computer Security Handbook, Jhon Wiley and Son, 2002.

[12] J.Krsul, Software Vulnerability Analysis, Purdue University Ph.D, 1998.

[13] B. McCarty, The Honey Files, IEEE Security and Privacy, 2004.

[14] D.Marchette, Computer Intrusión Detection and Network Monitoring, A Stadistical ViewPoint, New York: Springer, 2001.

[15] J. Dykstra, Essential Cybersecurity Science. Build, Test, and Evaluate Secure Systems, O'Reilly, 2016.

[16] J. V. a. A. O. W.G.J. Halfond, A Classification of SQL Injection Attacks and Countermeasures, Georgia Institute of Technology: College of Computing.

[17] "CERT® Advisory CA-1996-21 TCP SYN Flooding and IP Spoofi ng Attacks," 1996. [Online]. Available: http://www.cert.org/advisories/CA-1996-21.html.

[18] J. G. a. M. K. Friedemann Bitsch, Computer Safety, Reliability, and Security(Lecture Notes in Computer Science)., Springer, 2013.

[19] C. D. a. D. N. Serpanos, Network Security. Current Status and Future Directions., IEEE Press, 2007.

[20] N. Weiler, Honeypots for distributed denial of service, in Proceedings of the Eleventh IEEE International Workshops Enabling Technologies: Infrastructure for Collaborative Enterprises, Pitsburgh: PA, 2002.

[21] H.-u. R. Mohssen Mohammed, Honeypots and Routers, Collecting Internet Attacks, 2015.

[22] A. S. R.C. Joshi, Honeypots. A New Paradigm to Information Security, CRC, 2011.

[23] C. Stoll, The Cuckoo´s Egg: Tracking a Spy Throught the Maze of Computer Espionage, Gallery Books, September 13, 2005.

[24] T.Aslam, A Taxonomy of Security Faults in the UNIX Operating System, Purdue University Master´s thesis, 1995.

[25] S.Kumar, Classification and Detection of Computer Intrusión, Computer Science Departament, Purdue University Ph.D, 1995.

[26] J. D. J. a. L. T.Richardson, Developing a Database of Vulnerabilities to Support of Study of Denial Service Attacks, IEEE Symposium on Security and Privacy, May 1999.

[27] S. H. a. S. S. M.B. Salem, A Survey of Insider Attack Detection Research., Columbia University.

[28] A. C. K. a. R. E.Chan, Performing Live Forensics on Insider Attacks, November 2010.

[29] J. S. a. K. Y. K. Ahmad, International Journal: Classification of SQL Injection Attacks, 2010.

[30] N. W. a. J. Zhang, Factor-Analysis Based Anomaly Detection and Clustering, March 2005.

[31] R. C. a. P.M.B, Clustering by Compression, April 2005.

[32] K. Poulsen, Slammer worm crashed Ohio nuke plant network, Security Focus, 2003.

ANEXOS: RECONOCIMIENTOS

QUINTANA ROO
GOBIERNO DEL ESTADO
2016 • 2022

CONACYT

MÁS Y MEJORES OPORTUNIDADES

COQCYT
CONSEJO QUINTANARROENSE DE CIENCIA Y TECNOLOGÍA

RECONOCIMIENTO

A **JULIO CESAR MATUS CHAN**

Por su participación como **Ponente** en el **Quinto Encuentro de Jóvenes Investigadores**, efectuado del 17 al 19 de Octubre 2017.

Chetumal, Quintana Roo, 19 de Octubre 2017.

Ing. Víctor Manuel Alcérreca Sánchez
Director General del COQCYT

QUINTANA ROO
GOBIERNO DEL ESTADO
2016 • 2022

MÁS Y MEJORES OPORTUNIDADES

COQCYT
CONSEJO QUINTANARROENSE DE CIENCIA Y TECNOLOGÍA

CONACYT

RECONOCIMIENTO

A **JULIO CESAR MATUS CHAN**

Por haber obtenido el **PRIMER LUGAR** con el proyecto denominado **"Análisis e Implementación de una Solución Honeypot para un Entorno Experimental"** en el área de **"Robótica, Redes y Tecnologías de la Información"** en el marco del ***Quinto Encuentro de Jóvenes Investigadores***, efectuado del 17 al 19 de Octubre 2017.

Chetumal, Quintana Roo, 19 de Octubre 2017.

Ing. Víctor Manuel Alcerreca Sánchez
Director General del COQCYT


UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO
"ESTUDIO EN LA DUDA: ACCIÓN EN LA FE"
CONACYT
Otorgan la presente
CONSTANCIA
A:
Julio César Matus Chan
Por participar en el 4to. Congreso Interinstitucional de Jóvenes Investigadores (CIJI) 2017, que se llevó a cabo del 8 al 10 de noviembre en la ciudad de Villahermosa, Tabasco, México.
Mtro. Raúl Guzmán León
Secretario de Investigación, Posgrado y Vinculación
Consorcio de Universidades Mexicanas
4to. Congreso Interinstitucional de Jóvenes Investigadores 2017