UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

ANÁLISIS COMPARATIVO DE HONEYPOT DE BAJA INTERACCIÓN HONEYD Y KFSENSOR IMPLEMENTADOS VIRTUALMENTE

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor(a) (es):

Bach. Chapoñan Santisteban David

ORCID: https://orcid.org/0000-0002-5215-0874

Asesor(a):

Mg. Bravo Ruiz Jaime Arturo

ORCID: https://orcid.org/0000-0003-1929-3969

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2021

APROBACIÓN DEL JURADO

ANÁLISIS COMPARATIVO DE HONEYPOT DE BAJA INTERACCIÓN HONEYD Y KFSENSOR IMPLEMENTADOS VIRTUALMENTE

_____

Bach, Chapoñan Santisteban David
Autor

_____

Mg, Bravo Ruiz Jaime Arturo
Asesor

_____ _____

Dr, Vasquez Leiva Oliver
Presidente de Jurado

Mg, Sialer Rivera Maria Noelia
Secretario de Jurado

_____

Mg, Bances Saavedra David
Vocal de Jurado

Dedicatorias

Esta tesis se la dedico a mi Dios quién me guió por el buen camino, por darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban, enseñándome a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento.

A mi familia quienes por ellos soy lo que soy. Para mis padres Cruz y Victoria por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles. Me han dado todo lo que soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos.

A mis hermanos Zacarías, Amelia, Dulia, Julia, María Santos y Elizabeth por estar siempre presentes, acompañándome para poderme realizar, a mi hijo David Emanuel ya que es mi fortaleza para seguir adelante y a mi esposa Marianela por su apoyo incondicional.

Agradecimientos

El presente trabajo de tesis primeramente me gustaría agradecerle a Dios por bendecirme para llegar hasta donde he llegado, porque hiciste realidad este sueño anhelado.

A la Universidad Señor de Sipán por darme la oportunidad de estudiar y ser un profesional. De igual manera agradecer a mi profesor de Investigación y de Tesis de Grado al Mg. Ing. Jaime Arturo Bravo Ruiz, por su visión crítica de muchos aspectos cotidianos de la vida, por su rectitud en su profesión como docente, por sus consejos, que ayudan a formarte como persona e investigador.

Resumen

Los honeypot son herramientas informáticas muy útiles para aprender de los ataques virtuales y sus atacantes, esto permite retroalimentar y mejorar las configuraciones de seguridad de red de una organización. Algunas organizaciones no cuentan con los recursos de hardware para implementar todos los tipos de honeypot, además de ello no cuentan con el personal debidamente capacitado para implementar y dar mantenimiento a honeypot que tienen una alta interacción con los atacantes. Esto conlleva a necesitar el uso de los honeypot que tienen una baja interacción con los atacantes y que necesitan menos requerimientos de hardware para su implementación ya que inclusive se pueden implementar fácilmente de manera virtual. Entre los honeypot de baja interacción más comunes tenemos a kfsensor y honeyd. Por todo lo expuesto, surge la necesidad de comparar a kfsensor y honeyd implementados virtualmente.

En la presente investigación se propuso cuatro etapas para el desarrollo. Dentro de la primera etapa se consideró listar los honeypot más comunes; luego teniendo en consideración las características de licencia, puertos monitoreados y virtualización se eligió solo a dos de ellos para ser comparados, en donde se eligió a honeyd y kfsensor. Dentro de la segunda etapa se implementó virtualmente un entorno donde se instaló y configuró los honeypot elegidos en la primera etapa. Dentro de la tercera etapa se midió el rendimiento teniendo en consideración dos indicadores, los cuales son el consumo de memoria RAM, consumo de CPU (Procesador).

En los resultados se encontró que honeyd consume menos CPU con 0.0897 Ghz frente a 0.31785 Ghz y menos memoria RAM con 403.1 Megabytes frente a 981.5 Megabytes en comparación con KFSensor, además dentro de la cuarta etapa se analizó comparativamente los resultados con lo cual se puede concluir que honeyd presenta mas ventajas en comparación al honeypot kfsensor.

Palabras Clave:
Metodología, Honeypot, Tecnología, Hardware, Interacción, Virtualización

Abstract

Honeypots are very useful IT tools to learn from virtual attacks and their attackers, this allows to feed back and improve the network security configurations of an organization. Some organizations do not have the hardware resources to implement all types of honeypots, and also do not have the personnel properly trained to implement and maintain honeypots that have a high interaction with attackers. This leads to the need to use honeypots that have a low interaction with attackers and that need less hardware requirements for their implementation since they can even be easily implemented virtually. Among the most common low interaction honeypots are kfsensor and honeyd. For all these reasons, the need arises to compare kfsensor and honeyd implemented virtually.

In the present research, four stages were proposed for the development. In the first stage we considered listing the most common honeypots; then, taking into consideration the characteristics of licensing, monitored ports and virtualization, we chose only two of them to be compared, where honeyd and kfsensor were chosen. In the second stage, an environment was virtually implemented where the honeypots chosen in the first stage were installed and configured. In the third stage, the performance was measured taking into consideration two indicators, which are RAM memory consumption and CPU (Processor) consumption.

In the results it was found that honeyd consumes less CPU with 0.0897 Ghz compared to 0.31785 Ghz and less RAM memory with 403.1 Megabytes compared to 981.5 Megabytes compared to KFSensor, also in the fourth stage the results were comparatively analyzed with which it can be concluded that honeyd has more advantages compared to the honeypot kfsensor.

Keywords:
Methodology, Honeypot, Technology, Hardware, Interaction, Virtualization

Índice

I. INTRODUCCIÓN

Las instituciones privadas, requieren manejar grandes cantidades de información como por ejemplo datos de su personal, datos financieros, información de sus proveedores de bienes o servicios, también de sus procedimientos internos como por ejemplo contratos, pagos a proveedores, compra de mercadería, stock de bienes, entre otros datos. Toda esta información permite cumplir la demanda de sus clientes y al mismo tiempo cumplir con sus obligaciones frente al estado. Además de las instituciones privadas, las instituciones públicas, manejan información personal, económica y sobre la salud de sus ciudadanos, con la llegada del Gobierno Digital al Perú, las nuevas herramientas de tecnologías de información han cambiado varios procedimientos de servicios que brinda el estado con el fin de superar las barreras físicas y permitir acceder a los ciudadanos de manera virtual y en el menor tiempo posible a estos servicios.

Por consiguiente, la seguridad de las redes de datos es fundamental dentro de una organización. Ya que de esto depende mantener la privacidad de la información resguardada y evitar el uso malintencionado de la información por parte de algún atacante dispuesto a vulnerar redes de datos a cambio de alguna compensación económica o por simple un pasatiempo. Haciendo uso de la tecnología, por ejemplo, tenemos a los Honeypot, que sirven de señuelo y trampa para atacantes informáticos; con estos se puede aprender de los ciberataques y prevenirlos. Los honeypot pueden dividirse según el grado de interactividad con los atacantes en Honeypot de Baja y Alta interacción, siendo los de Baja interacción los más fáciles de implementar, entre estos tenemos a HoneyD y KFSensor.

Durante el proceso de esta investigación se estableció los pasos necesarios para realizar el análisis comparativo de honeypot de baja interacción honeyd y kfsensor implementados virtualmente. Luego así determinar entre ellos sus ventajas y desventajas.

1.1. Realidad Problemática.

Actualmente el mundo está pasando a una era digital en donde las nuevas tecnologías de información cambian la manera de hacer las cosas, tanto en las

organizaciones públicas y privadas; el ahorro de tiempo es solo uno de los beneficios de dichos cambios, pero al mismo tiempo nos expone a nuevos peligros. La plataforma Securelist es una librería de virus no comercial recopilada y actualizada por los analistas de la empresa Kaspersky; esta plataforma en el año 2019 publicó un artículo donde se habla de los ciberataques spam y phishing y sus efectos durante el primer trimestre del mismo año. Spam son mensajes de correo no deseados que de alguna manera perjudican al receptor y phishing es un ciberataque que consiste en establecer una aparente comunicación oficinal con sus víctimas suplantando la identidad de una entidad de confianza. En este artículo se indica que la proporción de ataques phishing realizados en contra de organizaciones de crédito aumentó en 5,23 % en comparación al cuarto trimestre del año 2018 hasta el 27,78 %; la proporción detectada se basa en las detecciones realizadas por la empresa Kaspersky por su componente Anti-Phishing incluidos en algunos de sus productos, en la siguiente figura se observa la distribución de organizaciones sujetas a ataques de phishing por categoría en el primer trimestre del 2019:

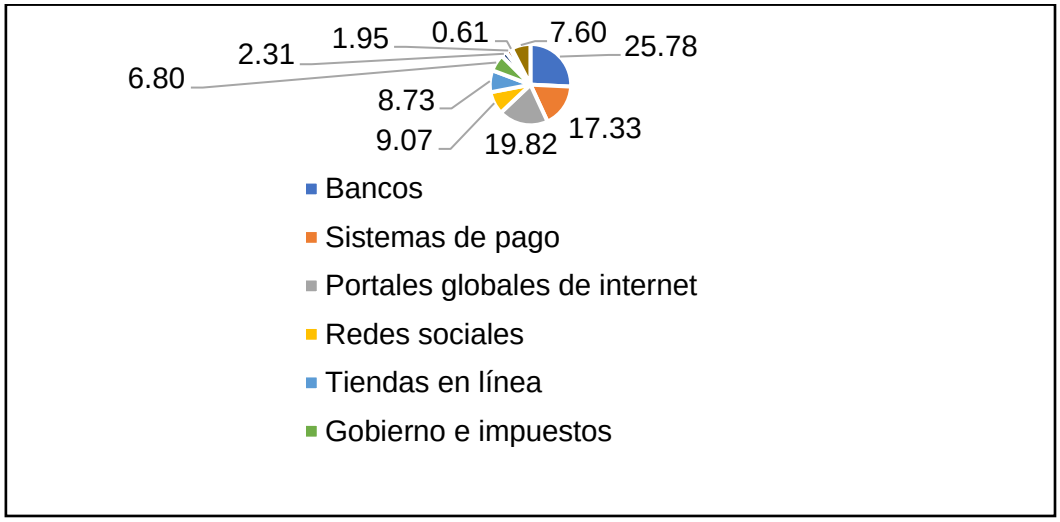


Figura 1. Proporción de Ataques a Organizaciones (%). Recuperado de Securelist.

El Perú no es ajeno a los cambios causados por la era digital, como una muestra de ello, en el sector público, el 13 de setiembre de 2018 se aprobó del Decreto Legislativo N° 1412 que aprueba la Ley del Gobierno Digital; este decreto tiene como fin establecer el marco de gobernanza del gobierno digital para la adecuada gestión en el uso de dispositivos tecnológicos de comunicación y el

régimen jurídico aplicable en el uso de las tecnologías en la prestación de los servicios a los ciudadanos. En el sector privado también existen muestras de estos cambios, los cuales están desde la manera en que se realizan las ventas (el uso del comercio electrónico) hasta los métodos de pago (el uso del monedero digital). Las nuevas tecnologías de información que son parte de la era digital permiten a las organizaciones el ahorro de costos y horas de trabajo por parte del personal, además rompe las barreras físicas acercándolas de manera virtual con sus usuarios finales. A pesar de los grandes beneficios que la tecnología pueda contribuir en la sociedad, también expone a las personas a nuevos peligros. El portal Agencia Peruana de Noticias (ANDINA) publicó un artículo virtual el 8 de agosto de 2020, en el cual la División de Investigación de Delitos de Alta Tecnología (DIVINDAT) de la Policía indicó que las denuncias por fraude informático son superiores en el año 2019 en comparación con el año 2018, en las siguientes figuras se indican el número de denuncias de delitos informáticos en el año 2018 y 2019:

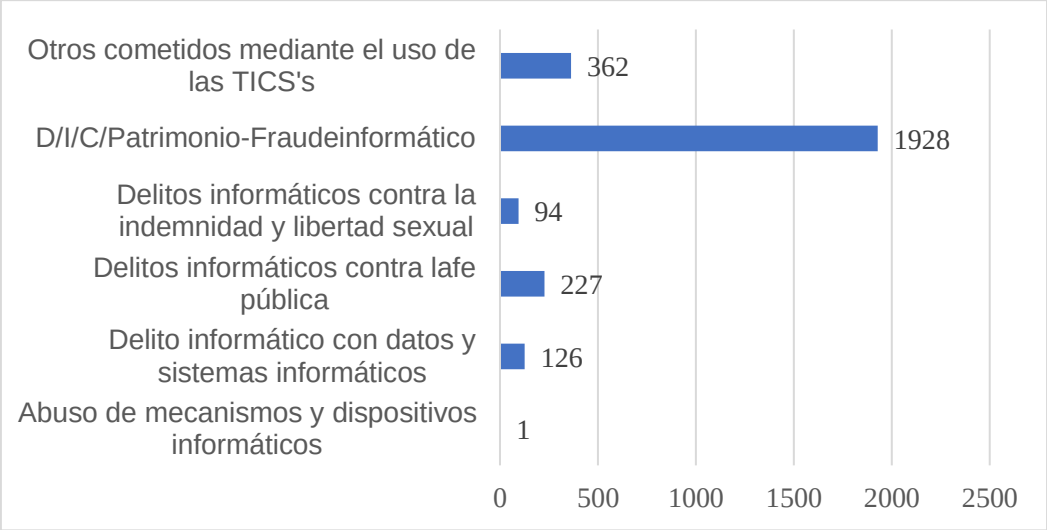


Figura 2. Cantidad de denuncias de Delitos Informáticos (año de 2018). Recuperado de DIVINDAT - PNP

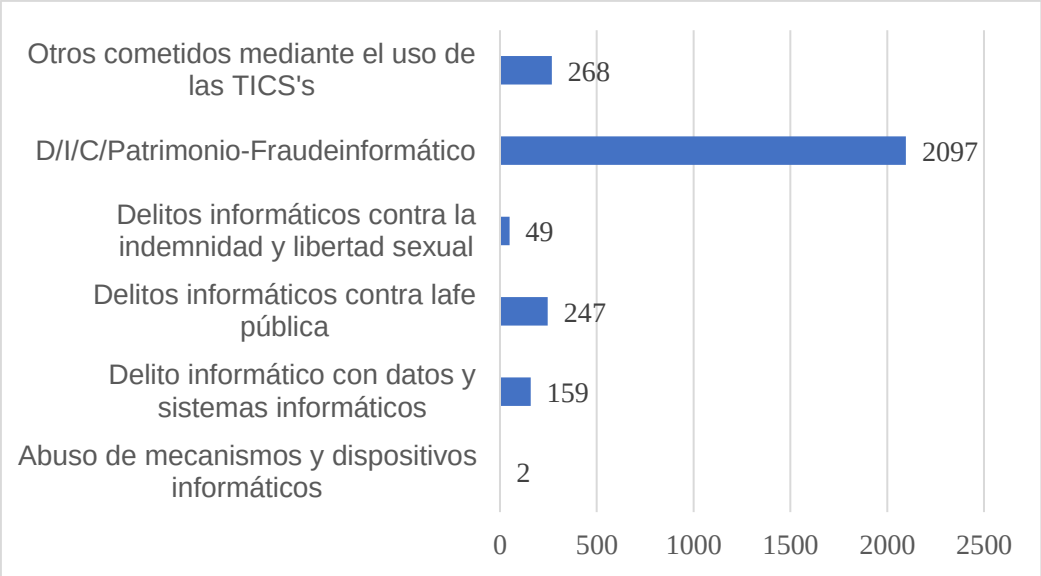


Figura 3. Cantidad de denuncias de delitos informáticos en el año 2019. Recuperado de DIVINDAT - PNP

Los peligros virtuales de la era digital pueden ser prevenidos o mitigados gracias al uso de técnicas y herramientas dentro de la seguridad informática, una de estas herramientas es los Honeypot, estos actúan como señuelos o trampas ante ataques virtuales, los honeypot permiten aprender de los ciberataques y en base a la información recolectada se puede retroalimentar las medidas de protección consideradas en la seguridad informática de las redes de datos. No existe un honeypot en específico que se tenga que utilizar en todos los casos; el uso de un honeypot depende de las capacidades de implementación de esta herramienta o la necesidad de que tipo de ataque se necesite aprender. Se puede dividir los tipos de honeypot en base al grado de la interactividad virtual con el atacante tenemos los de Baja y Alta interacción. Los de Baja interacción son más fáciles de implementar, además permiten el ahorro de costos extras en hardware ya que pueden ser implementados fácilmente tanto en entornos virtuales. Para determinar de manera general cuál de esos se requiere implementar para mejorar la seguridad en las redes de datos dependerá de las ventajas y desventajas que presenten entre ellos, entre los honeypot más destacados de este tipo tenemos a honeyd y kfsensor. Bajo los argumentos señalados nace la necesidad de realizar un análisis comparativo de honeypot de baja interacción honeyd y kfsensor implementados virtualmente.

.

1.2. Trabajos previos.

A continuación, se presentan diferentes investigaciones desde el campo internacional, nacional y local relacionadas con el problema de investigación.

1.2.1. Internacional

La investigación: *Implementation of Honeypot to Detect and Prevent Distributed Denial of Service Attack* (en español quiere decir *Implementación de Honeypot para detectar y prevenir ataques distribuidos de denegación de servicio*) del autor Sembiring, I. (2016) surgió con el fin de mejorar la seguridad de las redes de datos ante ataques de Denegación de Servicio Distribuido (DDOS). A través de su investigación se describe el uso de Honeypot de Baja interacción y otros programas informáticos de apoyo como *apache2* y *bind9* para dar información falsa a los atacantes potenciales. El desarrollo en la investigación se basa en establecer los pasos necesarios para realizar la implementación simulada del Honeypot y del software de apoyo. Luego de esto los resultados obtenidos por el Honeypot se procesan en gráficos a través del software Honeyd-Viz a través de la interfaz de red para que al administrador de la red de datos le sea más fácil realizar el análisis de la información de los ataques y autores de los mismos. En base a los resultados obtenidos de la investigación, pudo concluir que los Honeypot son capaces de detectar ataques de Denegación de Servicio Distribuido proporcionando información sobre los ataques y autores de los mismos en tiempo real. Además, al obtener información de los atacantes, se puede obtener direcciones de red y con esto crear políticas de seguridad que bloqueen la comunicación de la red de datos con las direcciones IP encontradas por ser consideradas potencialmente.

También tenemos la investigación *Intrusion Detection Using Honeypots* (en español quiere decir *Detección de intrusiones mediante Honeypots*) de los autores N., Bhagat; B., Arora (2018), surgió de la necesidad de

analizar el tráfico que viaja entre en el honeypot y los atacantes; en base a los resultados se concluyó que los honeypot proporcionan datos relevantes en una pequeña cantidad para que los investigadores de seguridad puedan entender fácilmente y que el análisis de los datos sea posible.

Además tenemos la investigación *Network Monitoring & Analysis along with Comparative Study of honeypots* (en español quiere decir *Monitoreo y análisis de red junto con estudio comparativo de honeypots*) de los autores Kumar & Girdhar (2017), la cual surgió de la necesidad de analizar y comparar el funcionamiento de honeypot como herramienta de seguridad informática necesaria de implementarse en redes con gran potencial de crecimiento; en base a los resultados se concluyó, que toda la información recolectada por los honeypot ayudan a convertir a la red de datos muy segura.

Luego podemos encontrar en la investigación *Optimized Virtual Honeynet with Implementation of Host Machine as Honeywall* (en español quiere decir *Honeynet virtual optimizada con implementación de máquina host como Honeywall*) de los autores Gautam, R.; Kumar, S.; Bhattacharya, J. (2015) donde propone la implementación de Honeywall y varios tipos de honeypot de manera virtual con el fin de reducir el consumo de memoria RAM y consumo de CPU. En base a los resultados de la investigación se concluyó que la implementación virtual contribuyó a reducir el consumo de CPU y memoria RAM en comparación a la implementación sin un software virtualizador.

Por otro lado Matías, J., & Gabriel, M. (2015) realizaron la investigación *Análisis y desarrollo de mejoras a un sistema Honeypot para mitigar ataques en servicios de VoIP* con la finalidad de analizar el rendimiento y la operatividad de un sistema Honeypot, luego en base análisis realizado implementar las mejoras en la seguridad informática de la red de datos para mitigar ataques a los servicios VOIP. El desarrollo de la investigación consta de tres fases. En la primera se realiza la

implementación del sistema Honeypot Artemisa dentro de la red de datos del gobierno de Córdoba. La segunda fase consta en analizar los datos recolectados del Honeypot acerca de los ataques a los servicios VOIP. En la tercera y última fase se agregó al Honeypot Artemisa la funcionabilidad de interactuar con el firewall perimetral del gobierno de Córdova. En base a los resultados obtenidos de la investigación se concluyó que en base a la información recolectada por el Honeypot se pudo diseñar e implementar mejoras fundamentales en la defensa de las redes de datos, para este caso, en especial para mitigar los ataques de los servicios VOIP.

Además, en la investigación *Dynamic Virtual Network Honeypot* (en español quiere decir *Red Virtual Dinámica Honeypot*) de los autores B., Sa Pham D., S., J., & M. (2019) surgió de la necesidad de implementar honeypot en el menor tiempo posible y con la menor cantidad de recursos necesarios para su construcción. Los resultados mostraron que al implementar honeypot con nuevas tecnologías que permitan que el software responda de manera dinámica ante los ataques se optimizó el consumo de recursos de hardware necesarios para su funcionamiento.

También los autores Sekar, Gayathri, Anisha, Ravichandran, & Manikandan (2018) de la investigación *Dynamic Honeypot Configuration for Intrusion Detection* (en español quiere decir *Configuración dinámica de Honeypot para la detección de intrusiones*) surgió de la necesidad de establecer el mecanismo para implementar honeypot en redes de datos con alto potencial de crecimiento con el fin de detectar intrusos. En base a los resultados obtenidos se concluyó que los honeypot son muy útiles para detectar y confundir a los piratas informáticos, además de eso que los honeypot brindan información valiosa para realizar rastreos a nivel de red de los atacantes.

Además, en la investigación *A Flexible Laboratory Environment Supporting Honeypot Deployment for Teaching Real-World Cybersecurity Skills* (en español quiere decir *Un entorno de laboratorio*

flexible que respalda Implementación de Honeypot para la enseñanza Habilidades de ciberseguridad del mundo real) de los autores Eliot, N.; Kendall, D.; Brockway, M. (2018) surgió de la necesidad de implementar un entorno controlado de red de datos para ofrecer alumnos que estudian temas acerca de seguridad informática. En base a los resultados se obtuvo un caso de éxito al ofrecer a sus estudiantes equipos y configuraciones en redes de datos de manera controlada con total libertad de acceso para mejorar el aprendizaje, lo cual permitir en una institución en producción sería un alto riesgo.

1.2.2. Nacional

El Registro Nacional de Trabajo de Investigación (RENATI) pertenece a la Superintendencia Nacional de Educación Superior Universitaria (SUNEDU). El RENATI ofrece de manera gratuita un buscador virtual para buscar trabajos de invetigación realizados a nivel nacional. En los resultados arrojados que guardan relación con las variables de estudio de este trabajo de investigación, que estén realizados fuera del ámbito local y sean de acceso abierto solo se encontró una tesis de pregrado, la cual es *Desarrollo de una Red Honeypot para la Detección de Intrusiones en la Municipalidad Distrital de Víctor Larco Herrera – Trujillo* del autor Valdiviezo, J. (2020). Esta investigación surgió por la necesidad de mejorar la seguridad informática de la red de datos de la Municipalidad Distrital de Víctor Larco Herrera, con respecto a la detección de intrusos, mediante el desarrollo de una red Honeypot. El desarrollo de la investigación se basa en cuatro fases. Durante la primera se analizó la situación actual de la red de datos, además se intentó vulnerar su seguridad mediante técnicas informáticas avanzadas. En la segunda fase se implementó la red de Honeypot en base a Snort y Kippo, ambos simulan servicios de red. Luego en la tercera fase, se realiza lo mismo que en la primera fase, pero esta vez es con la finalidad de determinar las diferencias entre la información de esta fase con la primera. En la última fase, en base a la información analizada en fases

anteriores, se aplicó correcciones en la seguridad informática de la red de datos. Según los resultados del trabajo de investigación se concluyó que la red Honeypot logró mejorar la detección de intrusos.

1.2.3. Local

En los resultados arrojados, en el buscador en línea de RENATI, que guardan relación con las variables de estudio de este trabajo de investigación, que estén realizados dentro del ámbito local y sean de acceso abierto solo se encontró una tesis de pregrado, la cual es la investigación de *Implementación de honeypot para la corrección de vulnerabilidades en la red de datos de la Municipalidad Distrital de Huambos* del autor Harlyn, A. (2018) surgió a partir de la necesidad de mejorar la seguridad informática de la red de datos de la Municipalidad Distrital de Huambos mediante el uso de Honeypot. La metodología de investigación que se utilizó es cuasi experimental, ya que la muestra no aleatoria es el personal administrativo de la institución de la Municipalidad Distrital de Huambos que manejan un software informático en específico, el Sistema Integrado de Administración Financiera (SIAF). En la investigación describió la implementación virtual de Honeypot dentro de la red de datos de la institución señalada, luego se realizó las pruebas necesarias para encontrar las vulnerabilidades de la red; posteriormente se realizó un análisis en base a la información recolectada a través del Honeypot para establecer las mejoras en las políticas de seguridad informática. Con los resultados obtenidos en la investigación se concluyó que Honeypot es un importante recurso informático para simular servicios de red, para despistar y aprender de los ciberataques y ayuda a mejorar la seguridad de red de datos.

Con base al análisis realizado a los antecedentes de estudio, entonces Honeypot puede ser utilizado como una herramienta muy útil para prevenir los ciberataques y también para reforzar las políticas de seguridad informática dentro de una red de datos en base a la

información recolectada. Lo señalado es útil para esta investigación porque nos muestra la gran importancia de la información recolectada por el Honeypot, ya sea de ataques controlados o ataques reales: esto muestra que su uso es de gran importancia para complementar la seguridad informática, así identificar las vulnerabilidades de la red de datos y en base a eso realizar las mejoras posteriores.

1.3. Teorías relacionadas al tema.

En este apartado se muestran las principales teorías recolectadas que guardan relación con las variables de la presente investigación.

1.3.1. Concepto de Honeypot

Un honeypot en español quiere decir *Trampa de miel,* esto hace referencia a que es un sistema informático que se "sacrifica" para atraer ciberataques, como un señuelo una trampa. Además, simula ser un objetivo para los hackers y utiliza sus intentos de vulnerar redes de datos para obtener información sobre los ciberataques y la forma en que operan sus autores o para distraerlos de otros objetivos reales.

Un honeypot es un sistema diseñado para analizar cómo los atacantes virtuales realizan sus ataques y los programas informáticos que utilizan para intentar entrar en un sistema y modificar, duplicar o eliminar sus datos o la totalidad de éstos (por ejemplo, copiar información privada de usuarios de un sistema). Por medio del aprendizaje de sus técnicas de intrusión, programas informáticos y métodos informáticos se puede, entonces, mejorar la seguridad informática.

Internamente un honeypot puede estar conformado por distintos programas informáticos, uno estos pueden servir para detectar un intruso o capturar las accionan los intrusos dentro de la red.

En escenarios más avanzados, varios honeypots forman una Honeynet, dando origen así una herramienta que proporciona funciones extendidas en capacidades en comparación a un solo honeypot, esto proporciona al administrador de la red de datos la mayor información posible para su

estudio. Esto favorece a presentar muchas más trampas o señuelos ante posibles.

1.3.2. Funcionamiento de un Honeypot

El honeypot es un sistema informático real, con servicios de red simulados, además posee aplicaciones y datos internos. Esto hace creer los ciberdelincuentes que es un objetivo real.

Honeypot posee vulnerabilidades de seguridad informática intencionales para ser más atractivo ante los atacantes virtuales y ser víctima de sus ciberataques.

Cuando un honeypot es atacado registra la mayor cantidad de información posible del autor, las herramientas utilizadas y cada uno de los pasos que abarcan el ciberataque.

En la siguiente figura podemos observar las etapas de su funcionamiento:

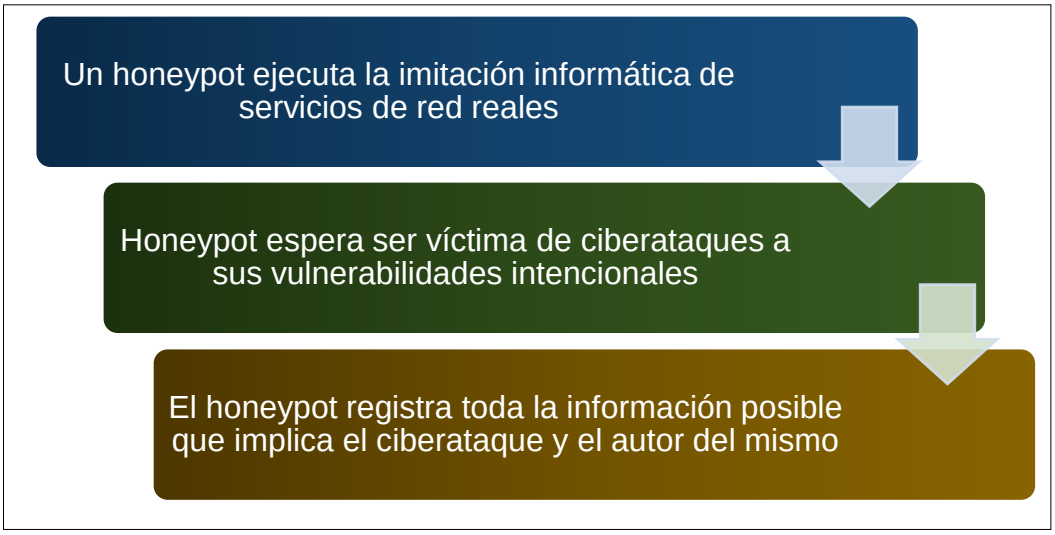


Figura 4. Funcionamiento de un Honeypot. Esta figura es una elaboración propia.

1.3.3. Datos obtenidos

En este apartado se listan los datos principales datos de obtener de la información registrada de un Honeypot durante los ciberataques:

a. El origen de los ciberataques.
b. El nivel de la amenaza potencial a la red de datos.
c. Los procedimientos ejecutados en los ciberataques.
d. Las herramientas informáticas utilizadas en los ciberataques.
e. Las vulnerabilidades en las medidas de seguridad informática existentes.

1.3.4. Funciones de Honeypot

En esta parte se lista las funciones más representativas de Honeypot:

a. Evadir la atención de los atacantes virtuales de la red de datos, sin comprometer el rendimiento y operatividad normal de los servicios de red.
b. Registrar toda la información posible de los ataques y sus autores para su análisis posterior.
c. Ordenar la información registrada para implementar normas específicas que ayuden a prevenir los mismos ataques.
d. Aprender sobre las nuevas vulnerabilidades y riesgos a los que está expuesta la red de datos.

1.3.5. Clasificación de Honeypot

Para clasificar a los Honeypot se debe tener en cuenta que sus funciones principales están estrechamente ligadas con la capacidad de información que puedan registrar. Es por ello que mientras más datos sean capaces de registrar se tendrá un mejor análisis de la información recolectada. Por lo tanto, a mayor cantidad de información registrada será mayor el grado de interacción con el atacante.

Teniendo en cuenta lo expuesto uno de los datos más importantes en la operatividad de un honeypot es el grado de interactividad con los intrusos. En base a ello podemos encontrar los siguientes tipos de honeypot:

a. Honeypot de Baja Interacción

Este tipo de honeypot tiene una interacción baja con atacante, y su funcionamiento consiste en imitar a programas informáticos u otros sistemas o hardware dentro de la red de datos. Se debe tener en claro que solo serán simuladas para hacer que el atacante intruso realice su ciberataque, y así, recolectar todos los datos que posibles.

En la investigación "*Implementación de honeypot para la corrección de vulnerabilidades en la red de datos de la Municipalidad Distrital de Huambos*" del autor Harlyn, A. (2018) se indica a Honeyd, KFSensor y Specter como honeypot de baja interacción más comunes.

b. Honeypot de Alta Interacción

Estos tipos de honeypot normalmente son hardware con sistemas reales que poseen los mismos servicios al igual como lo tuviera un sistema real dentro de la red. Es decir, son equipos que operan dentro de una red real, de manera tan similar como puede ser cualquier otro servidor físico. Por este motivo, este tipo de honeypot, tiene que estar debidamente protegido ya que, si no, el riesgo de intrusión se eleva considerablemente.

En la investigación "*Implementación de honeypot para la corrección de vulnerabilidades en la red de datos de la Municipalidad Distrital de Huambos*" del autor Harlyn, A. (2018) se indica a HoneyNet, ManTrap y HoneyWall como honeypots de alta interacción más comunes.

1.4. Formulación del Problema.

¿Qué honeypot de baja interacción tiene mejor eficiencia de rendimiento?

1.5. Justificación e importancia del estudio.

Esta investigación surge como la necesidad de comparar las ventajas y desventajas que ofrecen los honeypot de baja interacción honeyd y

kfsensor implementados virtualmente al reforzar la seguridad informática de las redes de datos.

Kaspersky es una compañía internacional dedicada a la seguridad informática; en mayo de 2019 realizó un informé de Spam y Phishing en el primer trimestre del mismo año señalado; en el cual se señala que el sistema Anti-Phishing de Kaspersky evitó más de 111,832,308 re direccionamientos a sitios de phishing, con respecto a periodos anteriores la cifra es superior en 35,220,650 de re direccionamientos.

Asimismo, la seguridad informática de las redes de datos implica un conjunto de técnicas que deben de ser aplicadas con el fin de mantener la privacidad e integridad de los datos gestionados. Una de aquellas técnicas es el uso de Honeypot, los cuales son herramientas que permiten prevenir y aprender de los ciberataques, por lo cual es importante conocer que beneficios nos ofrece su uso.

Es por eso que en la presente investigación se realizó un análisis comparativo de honeypot de baja interacción honeyd y kfsensor implementados virtualmente, de manera que nos ayudó a determinar cuál de ellos tiene mejor eficiencia en su rendimiento.

1.6. Hipótesis.

Hipótesis. - El honeypot de baja interacción honeyd es el más eficiente en su rendimiento.

1.7. Objetivos.

1.7.1. Objetivo general.

Comparar la eficiencia en rendimiento de los honeypot de baja interacción honeyd y kfsensor implementados virtualmente.

1.7.2. Objetivos específicos.

a. Determinar los honeypot de baja interacción a utilizar en la investigación.

b. Implementar la virtualización de los honeypot de baja interacción utilizados en la investigación.

c. Medir el rendimiento de los honeypot virtualizados.

d. Analizar de manera comparativa los resultados obtenidos.

II. MATERIAL Y MÉTODO

2.1. Tipo y Diseño de Investigación.

2.1.1. Tipo de Estudio

El tipo de estudio de la presente investigación es de tipo Cuantitativa, porque las medidas que se utilizaron en las evaluaciones del presente trabajo se basan en cantidades; además es Aplicada, porque se analizó los resultados obtenidos de implementaciones virtuales; con esto se determinó cuál de los honeypot de baja interacción honeyd y kfsensor tiene más ventajas; también es Tecnológica, ya que por su naturaleza se deriva de conocimiento científico apoyando por las ciencias de la computación.

2.1.2. Diseño de la investigación

El diseño para esta Investigación es Cuasi-Experimental, debido que no se utiliza ningún tipo de selección aleatoria, para seleccionar los honeypot de baja interacción utilizados.

2.2. Población y muestra.

2.2.1. Población

Está determinada por seis honeypots de baja interacción. Que se encuentran en la tabla N° 08 ubicado dentro del aporte práctico.

2.2.2. Muestra

La muestra fue determinada por conveniencia, y están conformados por HoneyD y KFSensor. Que se seleccionó de acuerdo a lo señalado en la tabla N° 05 ubicado dentro del aporte práctico.

2.3. Variables, Operacionalización.

2.3.1. Variable Dependiente

La variable es el "*Eficiencia de Rendimiento*"

Tabla 1:

Operacionalización de variable dependiente

Variable	Indicadores	Fórmula	Técnicas e instrumentos de recolección de datos
Eficiencia de rendimiento	Consumo de CPU	$\mathrm{R} = \frac{F_{reloj} \times N_{bits}}{C_{pro}}$	Técnica de observación
	Consumo de Memoria RAM	Cantidad en Megabytes	

Fuente: Elaboración propia

En donde:

R: Rendimiento

F_{Reloj}: Frecuencia del reloj

N_{bits}: Número de bits

C_{pro}: Ciclos de procesamiento

2.3.2. Variable Independiente

La variable es los "*honeypot de baja interacción"*

Tabla 2:

Operacionalización de variable independiente

Variable	Indicadores	Fórmula	Técnicas e instrumentos de recolección de datos
Honeypot de baja interacción	Tipo de Licencia	Costo de Licencia en Soles	Observación

Fuente: Elaboración propia

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

Para obtener la mejor información, se utilizó la herramienta de observación directa, la cual es el registro visual de lo que ocurre en una situación real. Para el presente trabajo se utilizó para consignar los datos obtenidos durante el desarrollo del segundo objetivo específico.

2.5. Procedimiento de análisis de datos.

Para el análisis de los datos se procedió a realizar lo siguiente:

a) Para medir el consumo de CPU: Es importante conocer que el uso de la CPU es un indicador del nivel actual de (sobre-) carga del procesador o CPU y de la capacidad que aún está libre. Esta información se obtiene del servidor que se utilizó en la ejecución de la implementación virtual de los honeypot de baja interacción honeyd y kfsensor según corresponda. El consumo del CPU se obtiene del siguiente cálculo:

$$\mathrm{R} = \frac{F_{reloj} \text{ x } N_{bits}}{C_{pro}}$$

En donde F_{reloj} es la frecuencia del Reloj del CPU del servidor. Además, el N_{bits} es el Número de Bits y el C_{pro} es los Ciclos de Procesamiento.

Para esta investigación se puede obtener el consumo del CPU directamente mediante uso del comando “htop” en la terminal del sistema operativo del servidor.

b) Para medir el consumo de Memoria RAM: Se debe de tener en claro que se trata de la cantidad de Memoria RAM que necesita el servidor para ejecutar implementación virtual de los honeypot de baja interacción honeyd y kfsensor según corresponda. Se puede obtener mediante el uso del comando “free -m -h” en la terminal del sistema operativo del servidor.

Los resultados obtenidos servirán para realizar tablas y figuras, con lo cual se pueda comparar la ejecución implementación virtual de los honeypot de baja interacción honeyd y kfsensor. Para esto se usa el programa informático Microsoft Excel.

2.6. Criterios éticos.

Veracidad

Dentro de la información contenida en la presente investigación tendrá señalado todos los procesos realizados durante el desarrollo en archivos digitales, así también disponible para su revisión y confirmación de su autenticidad.

Objetividad

La presente investigación contiene información que tiene por finalidad mostrar hechos reales desde diferentes perspectivas y la evaluación de trabajos de investigación que comprueben sus conclusiones, al mismo tiempo, el reporte de resultados que el mismo demuestre.

2.7. Criterios de Rigor Científico.

Fiabilidad

Tendrá la suficiente capacidad de reducir la imprecisión. Está relacionada con la reducción en lo posible del error aleatorio y necesita de un tamaño de muestra suficiente.

Validez

El correcto uso de las preguntas de investigación, de forma que las variables que se utilicen sean relevantes y que engloben todas las dimensiones que contienen las preguntas del presente trabajo de investigación.

III. RESULTADOS.

3.1. Resultados en Tablas y Figuras.

En este apartado se muestran los resultados obtenidos durante la presente investigación. Todas las pruebas virtuales se realizaron en un hardware con el procesador AMD A10-7870K Radeon R7. 12 Compute Cores 4C+8G 3.90 GHz, con memoria RAM de 16 GB y con el espacio de almacenamiento de 1TB. Se utilizó el software parar virtualizar VMWare, se realizó 20 simulaciones de ataques a los honeypot honeyd y kfsensor (ver anexo VI y VII) en funcionamiento a cada uno, durante cada intento se recolecto la información sobre el consumo de CPU (Unidad Central de Procesamiento o Procesador) y memoria RAM, para luego realizar un promedio de los resultados.

Cuando se realizó la implementación virtual de los honeypot honeyd y kfsensor, se pudo comprobar que la diferencia del uso del procesador es más del cinco por ciento de la capacidad total del uso del procesador del servidor o también conocido como Unidad Central de Procesamiento (CPU), teniendo a kfsensor como el honeypot que demanda mayor consumo; el honeypot kfsensor hace uso de interfaces gráficas que

interactúan con el usuario para ser instalado y configurado. La interfaz gráfica de un programa a mayor cantidad de efectos visuales y mayor interactividad con el usuario va a demandar más uso de recursos del computador virtual donde se ejecuta; esto puede ser la razón principal de que kfsensor consumió mayor cantidad de CPU que honeyd, ya que este último se instaló y configuró mediante el uso de una consola del sistema operativo. A continuación, tenemos una tabla y una figura donde se indican los resultados del indicador *Consumo de CPU*; se usó como unidad de medida el porcentaje (Ghz).

Tabla 3:

Resultados de Consumo de CPU (Ghz)

N°	Honeypot	Consumo CPU (Ghz)	Porcentaje (%)
1	HoneyD	0.0897	0.22
2	KFSensor	0.31785	0.78
Total		0.40755	1

Fuente: Elaboración propia

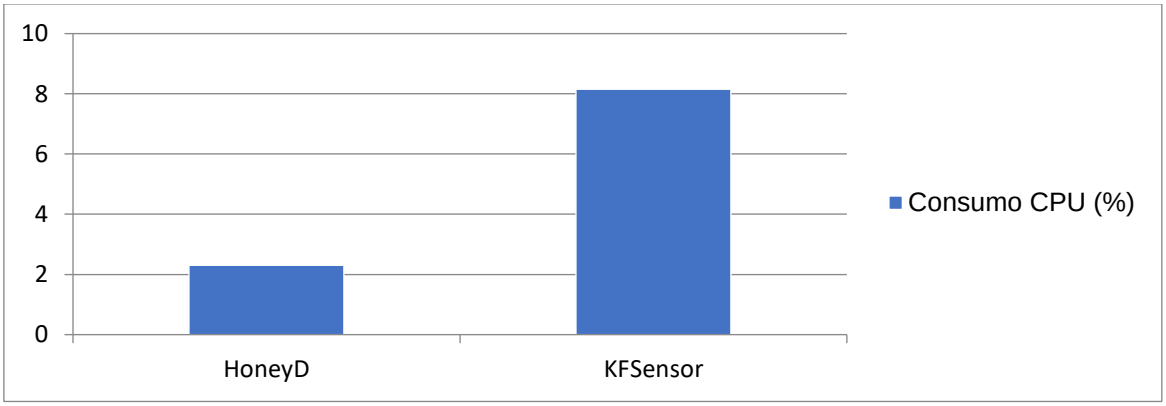


Figura 5. Resultados de Consumo de CPU (%). Esta figura es elaboración propia.

Luego, al comparar la implementación virtual de honeyd y kfsensor con respecto al consumo de memoria RAM se observó una diferencia de más de 500 Megabytes, donde kfsensor exige una mayor cantidad de megas al computador virtual donde se ejecutó en comparación a honeyd; para este último solo fue suficiente el uso de una terminal. A continuación, tenemos una tabla y una figura donde se indican los resultados del indicador *Consumo de Memoria RAM*; se usó como unidad de medida el Megabytes:

Tabla 4:

Resultados de Consumo de Memoria RAM (Megabytes)

N°	Honeypot	Consumo Memoria RAM (Megabytes)	Porcentaje
1	HoneyD	403.1	0.29
2	KFSensor	981.85	0.71
Total		1384.95	1

Fuente: Elaboración propia

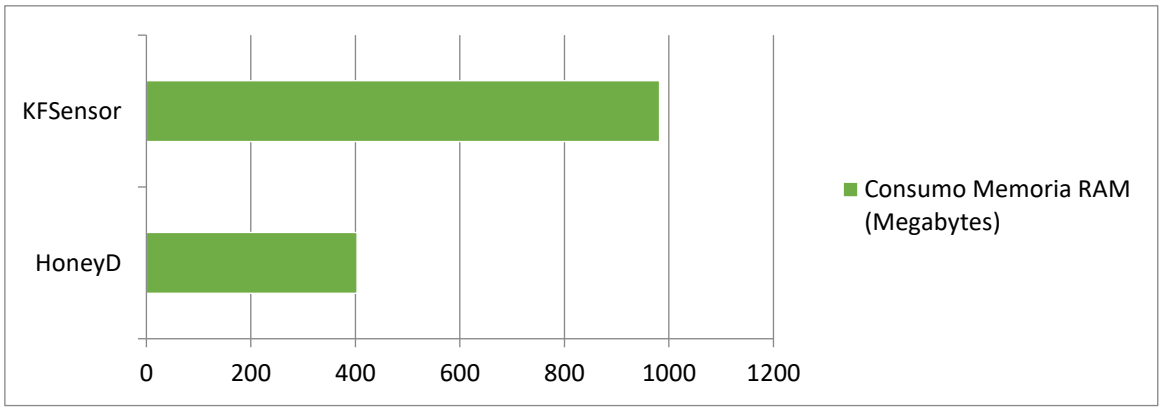


Figura 6. Resultados de Consumo de Memoria RAM (Megabytes). Esta figura es elaboración propia.

3.2. Discusión de resultados.

En esta investigación al comparar los honeypot de baja interacción honeyd y kfsensor implementados virtualmente, se encontró que honeyd consume menos CPU y menos memoria RAM que kfsensor, con lo cual se puede decir que honeyd presenta más ventajas que kfsensor en la implementación virtual. Frente a lo señalado se aceptó la hipótesis que establece que el honeypot de baja interacción honeyd presentado es más eficiente para su implementación en comparación al honeypot kfsensor. A continuación, se muestra una tabla donde se comparan los resultados:

Tabla 5:

Comparación de Resultados obtenidos según los indicadores

N°	Honeypot	Consumo CPU (Ghz)	Consumo Memoria RAM (Megabytes)
1	HoneyD	0.0897 Ghz(0.22%)	403.1
2	KFSensor	0.31785 Ghz(0.78%)	981.85

Fuente: Elaboración propia

Se ha buscado en la literatura y no se ha evidenciado otras investigaciones que hayan realizado la recolección de datos sobre el consumo de CPU y memoria RAM.

3.3. Aporte práctico.

Para el desarrollo de esta investigación se consideró cuatro etapas, y en cada una de ellas se establecieron los pasos necesarios para comparar implementación virtual de los honeypot de baja interacción honeyd y kfsensor.

En la siguiente figura se expone los pasos de cada una de las etapas que se utilizó en el presente trabajo de investigación:

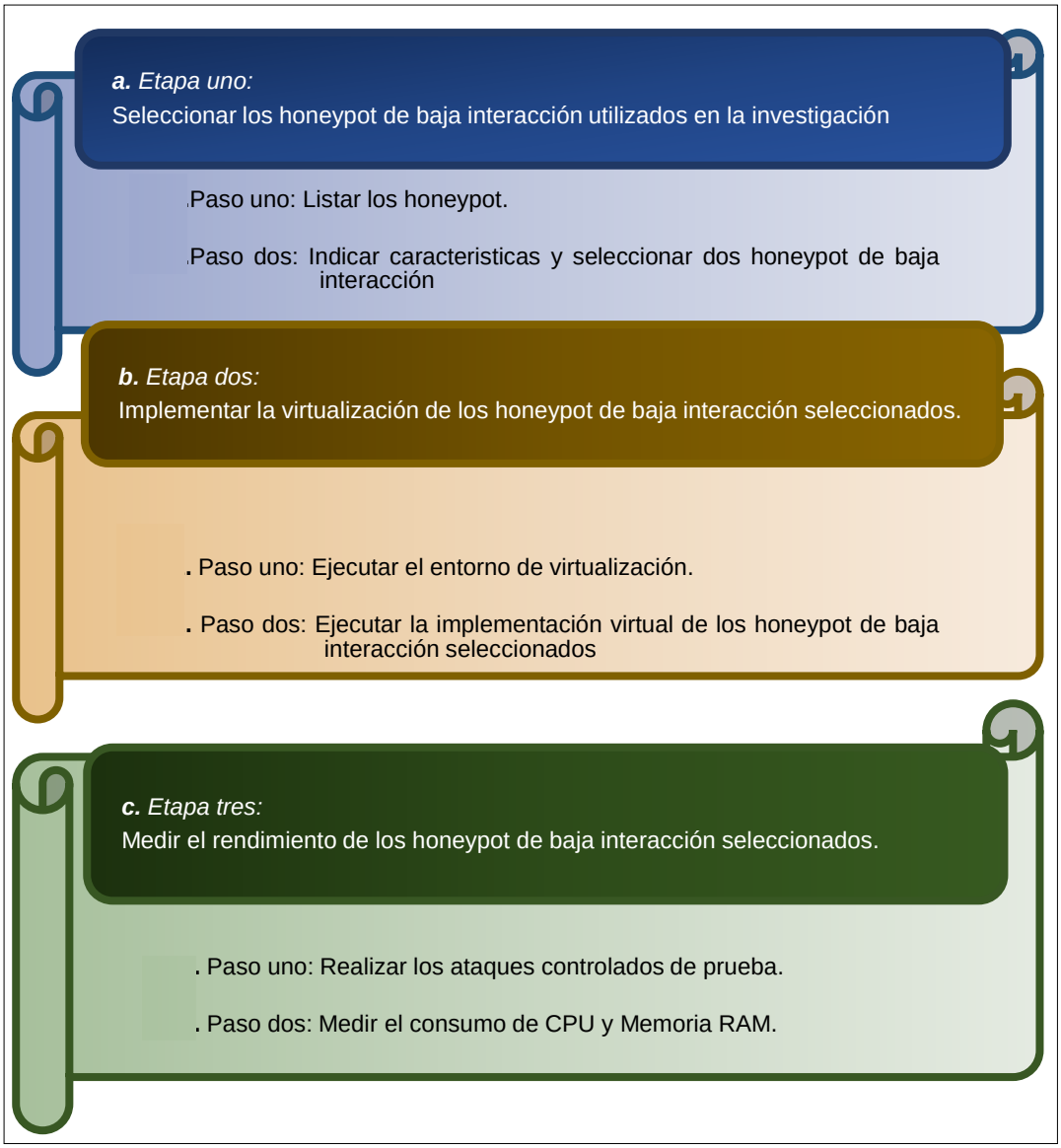


Figura 7. Etapas de desarrollo de la investigación. Esta figura es elaboración propia.

a. Etapa uno

Esta etapa se conformó por dos pasos.

a.1. *Paso uno*:

Se listó los honeypot encontrados durante la recolección de la información para la presente investigación. De acuerdo a la información recolectada en el apartado de teorías relacionadas

al tema. En la siguiente tabla se observa cada uno de ellos con una breve descripción:

Tabla 6:

Lista de honeypot

N°	Nombre	Descripción
1	Honeyd	Honeyd es un programa de código libre que permite a un usuario configurar y ejecutar varios hosts virtuales en una red informática.
2	KFSensor	Es un honeypot de baja interacción que se ejecuta bajo Windows y permite simular servicios vulnerables.
3	Specter	Es un honeypot inteligente basado en sistemas de detección de intrusos, vulnerables y atractivos a los atacantes.
4	Honeynet	Se puede considerar una herramienta de investigación. Es un tipo de Honeypot que consiste en una red diseñada para ser comprometida por atacantes en la red.
5	Mantrap	Emula una variedad de diferentes maquinas (FTP, HTTP, SMTP, ODBC) en una sola implementación.
6	Honeywall	Un “honeywall” puede proporcionar la seguridad básica de un honeypot y evitar que los ataques dirigidos contra este lleguen a tu sistema activo.
N°	Nombre	Descripción

Fuente: Elaboración propia.

a.2. *Paso dos*:

Este paso es necesario porque se indican las características de los honeypot, para poder elegir solo dos de estos para que sean considerados en la implementación virtual en la siguiente etapa. Se realizó una tabla comparativa de los honeypot mencionados en el paso anterior donde se consideró las siguientes características:

Licencia, existen dos tipos de licencia. La Licencia Privativa o una Licencia de Código Abierto; el primero

tiene un costo monetario y generalmente solo se permite el uso, pero no se da acceso al código fuente para desarrollar modificaciones o mejoras sobre el software; el segundo no tiene costo de por su uso y además permite acceder al código fuente para desarrollar modificaciones o mejoras sobre el código original.

Cantidad de puertos monitoreados, los puertos de red monitoreados pueden ser TCP o UDP y existen 65535 puertos.

Tipo, es el tipo de honeypot según el nivel de interactividad que tiene con el atacante, se clasifican en Alta y Baja interacción.

Virtualizable, es la capacidad de ser implementado de manera virtual.

En la siguiente tabla se muestra una lista de honeypots y las características mencionadas anteriormente:

Tabla 7:

Honeypot de baja interacción

N°	Nombre	Licencia	Cantidad de puertos monitoreados	Tipo
1	Honeyd	Código abierto	65535	Baja
2	KFSensor	Privativa	65535	Baja
3	Specter	Privativa	14	Baja
4	Honeynet	Código abierto	65535	Alta
5	Mantrap	Código abierto	65535	Alta
6	Honeywall	Código abierto	65535	Alta

Fuente: Elaboración propia

Es importante conocer la licencia, porque con ello podemos evaluar los costos necesarios para hacer uso y además el costo para tener acceso al código fuente y realizar las mejoras que se consideren en beneficio de mejorar la seguridad de la red de datos.

También, es importante conocer la cantidad de puertos monitoreados, porque mientras más cantidad de puertos pueda monitorear un honeypot hay mayor cantidad de información que se pueda recolectar de los ataques a los que esté sometido.

Además, saber si un honeypot es virtualizable, es importante porque esta característica permite conocer si un honeypot se puede implementar de manera virtual, con ello se puede hacer uso no exclusivo de un determinado hardware para una sola implementación virtual y el mismo hardware puede ser usado en varias implementaciones virtuales al mismo tiempo, esto permite ahorro de costos para las organizaciones.

En la elección se tuvo en consideración elegir un honeypot con licencia de código abierto y otro con licencia privativa, porque se desea comparar y conocer las ventajas y desventajas que cada una ofrece a la seguridad de una red de datos.

Luego, se tuvo en consideración elegir un honeypot que sea capaz de monitorear la mayor cantidad de puertos posibles, porque en la presente investigación se busca que las conclusiones obtenidas sirvan como referencia en la toma de decisiones en el momento de evaluar las opciones de cómo proteger una red de datos y de nada serviría tomar como referencia a un honeypot que está limitado en este aspecto de protección.

Se tuvo en consideración que ambos honeypot elegidos deben de ser de baja interacción, porque permiten ahorro de hardware y además no requieren de tiempo en mantenimiento.

Se tuvo en consideración que ambos honeypot elegidos deben de tener la capacidad de ser implementados virtualmente, porque la virtualización permite hacer uso no exclusivo de hardware, lo cual trae el beneficio de ahorro de recursos económicos y de hardware dentro de una organización.

Según lo expuesto, se eligió HoneyD porque es el único con licencia de código abierto en comparación con los demás y se eligió KFSensor porque tiene la capacidad de monitorear 65535 puertos de red TCP o UDP al contrastar con los 14 puertos de red soportados por el honeypot Specter, tal como fue señalado en la investigación "*Implementación de una honeynet para la Ciberdefensa de Infraestructuras Críticas*" del autor Eduardo, M. (2015).

En la siguiente tabla se muestra el resultado:

Tabla 8:

Honeypot de baja interacción elegidos para el desarrollo de la investigación

N°	Nombre	Licencia	Cantidad de puertos monitoreados	Virtualizable	Tipo
1	Honeyd	Código abierto	65535	Si	Baja
2	KFSensor	Privativa	65535	Si	Baja

Fuente: Elaboración propia

b. Etapa dos

Esta etapa se conformó por dos pasos, porque en el primer paso fue necesario implementar el ambiente virtual para luego en el siguiente paso se implementó los honeypot sean honeyd y kfsensor en el ambiente virtual. Los cuales fueron:

b.1. *Paso uno*: Implementación del entorno virtual.

En este paso se realizó la implementación del entorno virtual necesario para la virtualización de los honeypot de baja interacción seleccionados.

Para ello se realizó las siguientes tareas:

Se estableció el uso de una sola computadora con los requerimientos mínimos necesarios para implementar los honeypot elegidos anteriormente, porque al ser dos implementaciones virtuales solo es necesario hacer uso de un computador. A continuación, se muestra una tabla con los requerimientos mínimos del computador que se utilizó.

Tabla 9:

Características mínimas requeridas para la virtualización

N°	Característica	Valor
1	Procesador	AMD A10-7870K Radeon R7. 12 Compute Cores 4C+8G 3.90 GHz
2	Memoria RAM	16 GB
3	Disco duro	1 TB

Fuente: Elaboración propia

Se instaló el software de virtualización VMware Player en el hardware establecido, porque tiene soporte de DirectX 10.1 en Windows en comparación con Virtualbox y HyperV, lo cual eleva el rendimiento en interfaces gráficas. Además, VMware

tiene mejor rendimiento en la utilización de memoria RAM, lectura y escritura en disco duro.

Figura 8. Instalador de VMware. Esta figura es elaboración propia.

Se creó dos máquinas virtuales, porque cada una de ellas se utilizó para implementar cada tipo de honeypot de baja interacción elegido en la primera etapa.

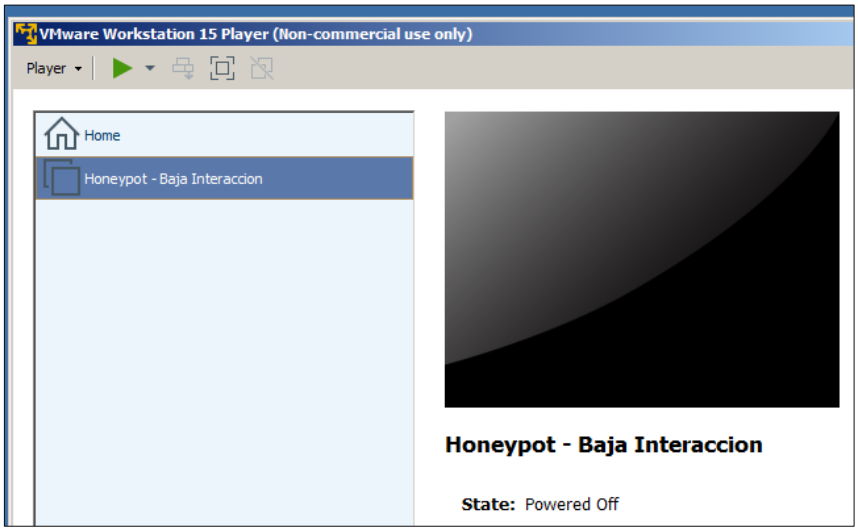


Figura 9. Creación de máquina virtual. Esta figura es una elaboración propia.

Se instaló en una de las máquinas virtuales una distribución de sistema operativo Linux llamado Ubuntu y en otra Windows 7.

Se utilizó Windows 7, porque los requerimientos de recursos (por ejemplo, memoria RAM, Procesador, Disco Duro) es el que menos recursos necesita en comparación con Windows 8, Windows 10 o Windows Server 2008/2012/2016, además para

una organización esto significa ahorro de recursos necesarios para la virtualización.

Figura 10. Instalación virtual de Windows 7. Esta figura es una elaboración propia.

Se utilizó Ubuntu porque este tipo de sistema operativo cuenta con los repositorios necesarios de donde se descargó las dependencias compatibles para instalar HoneyD.

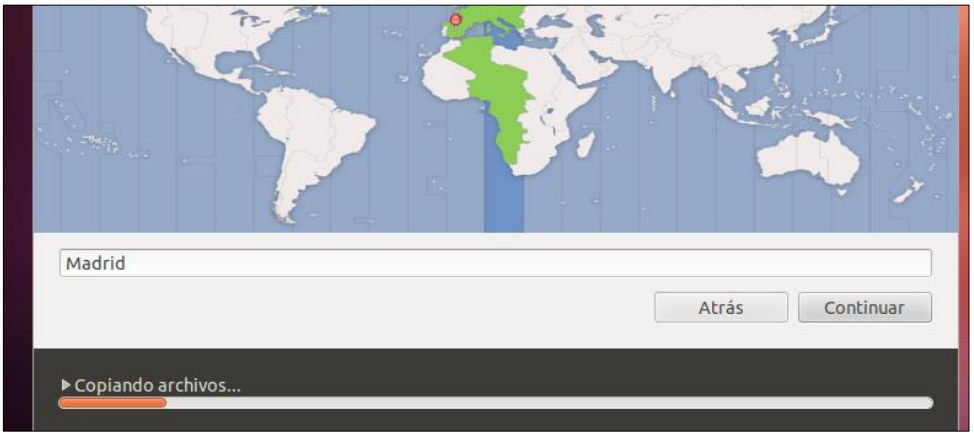


Figura 11. Instalación virtual de Ubuntu. Esta figura es una elaboración propia.

b.2 *Paso dos*:

En este paso se ejecutó la virtualización de los dos honeypot elegidos en la primera fase, luego que se estableció el entorno de virtualización, se procedió a ejecutar la virtualización de los dos honeypot elegidos en la primera fase, con la finalidad de

instalar y configurar cada uno de los honeypot en sus máquinas virtuales, para poder medir los indicadores de la presente investigación.

Para la virtualización de KFSensor se realizó las siguientes tareas:

Se determinó necesario instalar Npcap para el correcto funcionamiento del honeypot KFSensor.

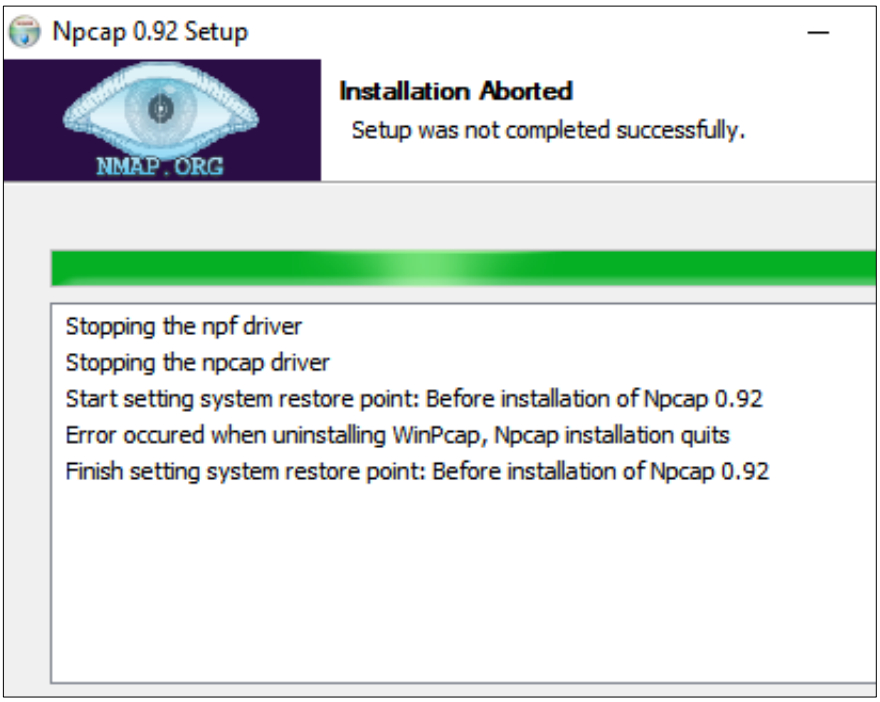


Figura 12. Instalación de Npcap. Esta figura es una elaboración propia.

Se descarga el instalador en versión de prueba de KFSensor.

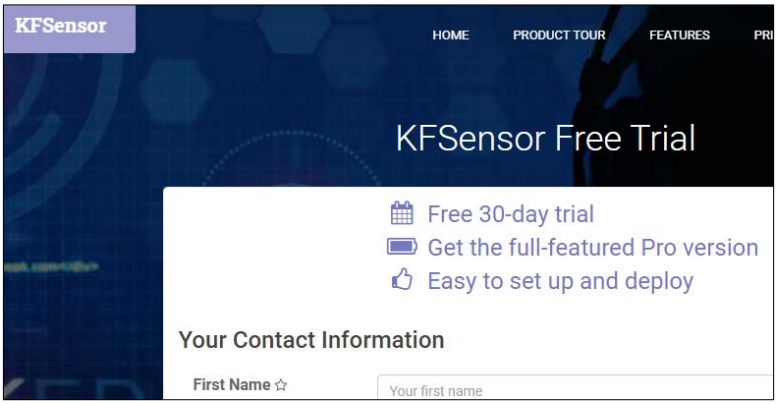


Figura 13. Versión de Prueba de KFSensor. Esta figura es una elaboración propia.

Se ejecuta el programa “services.msc” y se observa que KFSensor se instaló como un servicio dentro del equipo.

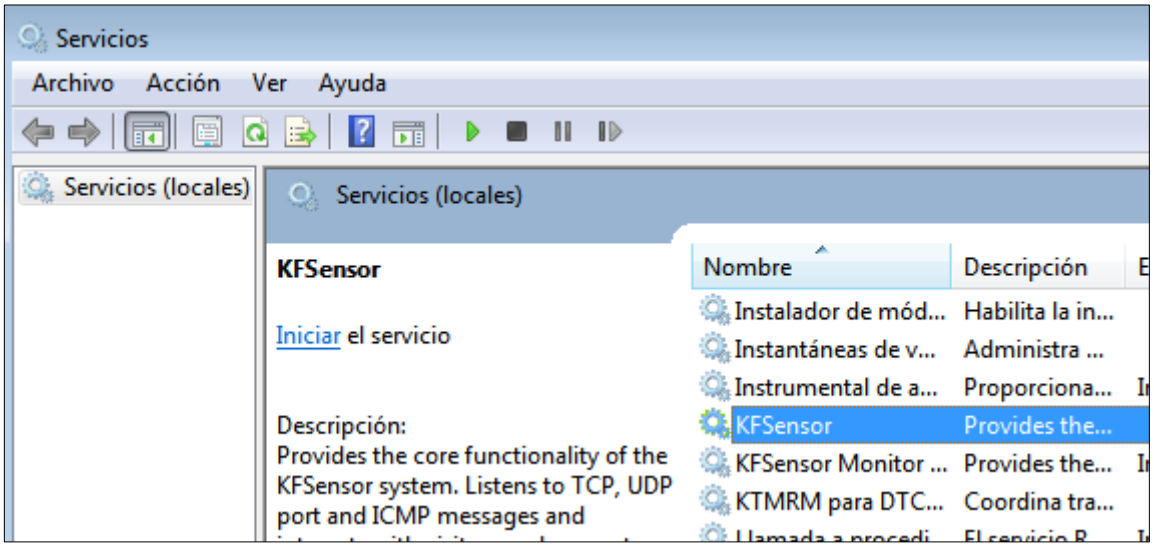


Figura 14. El programa "services.msc”. Esta figura es elaboración propia.

KFSensor tiene la opción llamada “Escenarios”, esta opción permite asignar los puertos que imitará virtualmente, como ejemplo se configuró un escenario donde KFSensor deberá comportarse como un servidor IIS Server Web con el puerto TCP número 80.

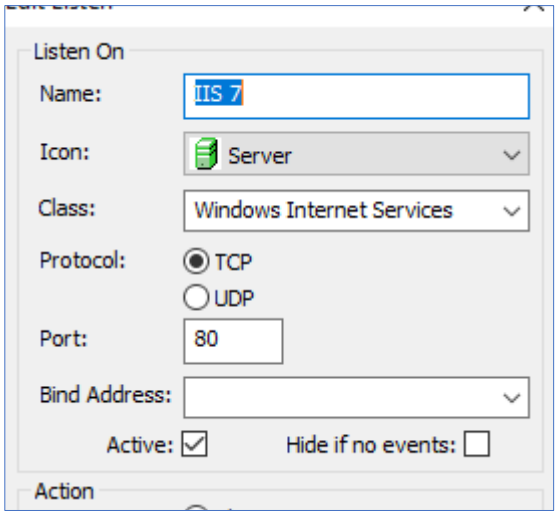


Figura 15. Configuración del puerto TCP número 80. Esta figura es elaboración propia.

Con lo expuesto anteriormente, KFSensor quedó listo para ser sometido a pruebas virtuales de ataques para poder evaluar los indicadores de consumo de CPU (Unidad Central de Procesamiento) y consumo de memoria RAM. Todo lo expuesto anteriormente se puede ver con más detalles en el anexo VI.

Para la virtualización de HoneyD se realizó las siguientes tareas:

Se ingresó a la máquina virtual que no fue utilizada en la virtualización anterior.

Se ingresó a la terminal de consola.

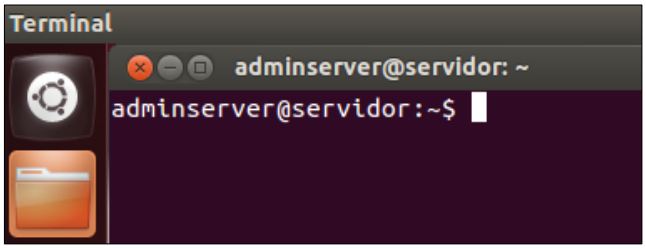


Figura 16. Interfaz de consola de maquina virtualizada hasta este paso. Esta figura es elaboración propia.

Se accedió como super usuario, para poder tener los permisos como usuario para instalar y/ configurar algún programa dentro del sistema, con el siguiente comando:

```
sudo su
```

Figura 17. Comando para acceder como super usuario

Se utilizó el siguiente comando para instalar Honeyd:

```
apt-get install honeyd
```

Figura 18. Comando para instalar honeyd

Se instaló las dependencias, usando los comandos:

```
sudo apt-get install libevent-dev libdumbnet-dev libpcap-dev libpcre3-dev libedit-dev bison flex libtool automake zlib1g-dev python
```

Figura 19. Comando para instalar dependencias

Se creó un archivo de configuración para definir el puerto 80 para que el honeypot pueda virtualizar un servicio web. Primero se crea el archivo “honeyd.conf” y luego se ejecuta honeyd con el siguiente comando.

```
sudo nano honeyd.conf

# /usr/share/honeyd/scripts/win32/web.sh

sudo honeyd -d -f honeyd.conf
```

Figura 20. Comando para crear el archivo honeyd.conf

Con todo lo anterior se dio inicio al servicio de honeyd para que entre en funcionamiento. Todo lo expuesto anteriormente se puede observar a detalle en el anexo VII.

c. Etapa tres

Esta fase se conformó por dos pasos. Los cuales fueron:

c.1. *Paso uno*:

En este paso se realizó 20 ataques de DDOS desde la terminal de consola a cada servidor virtual implementado, para que durante los ataques se recolecte en 20 instantes los datos de consumo de memoria RAM y consumo de CPU. Esto se puede observar con más detalle en los anexos VIII y IX

c.2. *Paso dos*:

En este paso se midió el consumo de Memoria RAM y consumo de CPU, en donde se utilizó programas de software libre y herramientas integradas en el sistema operativo utilizado.

Por ello se consideró las herramientas Top y HTOP en SO de Distribución Linux; comandos que se pueden usar en SO de distribución Linux, al ejecutarlo en terminal, estos programas muestran en tiempo real el consumo de Memoria RAM y uso del CPU. El comando para utilizar Htop se necesitó instarlo con el comando:

```
sudo apt-get install -y htop
```

Figura 21. Comando para instalar el programa htop

Luego para usar Htop se utilizó:

```
htop
```

Figura 22. Comando para ejecutar el programa htop

Y para utilizar Top se usó el comando:

```
top
```

Figura 23. Comando para ejecutar el comando top

En la máquina virtual con sistema operativo Windows 7, se utilizó la herramienta MSI Afterbunner, para poder ver en tiempo real el consumo de memoria RAM y consumo de CPU.

d. Etapa cuatro

Esta fase se conformó por un paso, en el cual se generó las tablas y figuras que permitan comparar la implementación virtual de los honeypot de baja interacción HoneyD y KFSensor. Todos los datos recogidos fueron recolectados y pasados a un archivo con extensión XLSX, para procesar los datos y analizar los diferentes indicadores, para así comparar analizando todos los datos obtenidos, ventajas y desventajas de cada una de ellas y así obtener las conclusiones de nuestra presente investigación. Esto se puede observar con más detalle en los anexos VIII y IX.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones.

Se seleccionó a honeyd y kfsensor como los dos únicos honeypot de baja interacción para ser comparados en su implementación virtual en la presente investigación.

Se ejecutó la implementación virtual de los honeypot de baja interacción de honeyd y kfsensor con el software de virtualización VMware, donde se utilizó como sistema operativo Windows 7 y Ubuntu, para implementar kfsensor y honeyd respectivamente.

Se analizó comparativamente la eficiencia del rendimiento de honeyd y kfsensor implementados virtualmente, teniendo en cuenta dos indicadores, los cuales son consumo de CPU y consumo de memoria RAM, en los resultados obtenidos se encontró que honeyd consume menos CPU con 0.0897 GHZ frente a 0.31785 GHZ y menos memoria RAM con 403.1 Megabytes frente a 981.5 Megabytes en comparación con KFSensor.

Se comparó los honeypot de baja interacción honeyd y kfsensor implementados virtualmente, en donde kfsensor consume más memoria RAM y CPU que honeyd, con lo cual se acepta la hipótesis donde se indica

que el honeypot de baja interacción honeyd presenta más ventajas en comparación al honeypot kfsensor.

4.2. Recomendaciones.

Se recomienda que la implementación de los honeypot de baja interacción debe de ser virtualizado con el software VMWare para obtener resultados semejantes a los de la presente investigación.

Para trabajos futuros se considere realizar las pruebas de ataques para medir la eficiencia en la protección de los datos con simulación de ataques reales.

REFERENCIAS

Aleksey, A., Sergey, V., Igor, M., & Dmitry, S. (2017). Development and Implementation of a Honeypot-Trap. *IEEE*, 382 - 385.

Álvarez, A. (2017). Honeypot, añadiendo una capa de seguridad en una empresa de telecomunicaciones. *(Tesis de Pregrado).* Universitat Oberta de Catalunya, Catalunya.

Avilés, Y. (2016). Implementación de una herramienta honeypot para detección y respuesta a ataques. *(Tesis de Postgrado).* Escuela Superior Politécnica del Litoral, Guayaquil.

B., P., Sa Pham D., S., S., N., J., Y., & M., P. (2019). Dynamic Virtual Network Honeypot. *School of Electronic Engineering*, 375 - 377.

Correa, A. (2016). Simulación de un honeypot para detectar ataques y vulnerabilidades en redes IPV6. *(Tesis de Pregrado).* Universidad Tecnológica Equinoccial, Quito.

Dios, S., & Ortiz, D. (2018). Diseño lógico y simulación de una red espejo virtual (honeynet) para la detección de intrusos informáticos en la zona perimetral. *(Tesis de Pregrado).* Universidad Nacional de Trujillo, Trujillo.

Eduardo, M. (2015). Implementación de una honeynet para la Ciberdefensa de Infraestructuras Críticas. *(Tesis de Pregrado).* Instituto Universitario Aeronautico, Córdova.

Eliot, N., Kendall, D., & Brockway, M. (2018). A Flexible Laboratory Environment Supporting Honeypot Deployment for Teaching Real-World Cybersecurity Skills. *IEEE*, 34884 - 34895.

Espí, S. (2017). Desarrollo de un honeypot para la monitorización y prevención de ataques. *(Tesis de Pregrado).* Universidad Politécnica de Valencia, Valencia.

Fernández, A. (2017). Desarrollo de una herramienta honeypot para un uso eficiente en seguridad informática. *(Tesis de Pregrado).* Universidad de Jaén, Andalucía.

Flórez, I., & Quintana, J. (2018). Sistema de detección de ataques inforrmáticos a redes de datos empresariales soportado en honeypots. *(Proyecto de Investigación).* Universidad de Cartagena, Cartagena.

Gautam, R., Kumar, S., & Bhattacharya, J. (2015). Optimized Virtual Honeynet with Implementation of Host Machine as Honeywall. *IEEE*.

Harlyn, A. (2018). Implementación de honeypot para la correción de vulnerabilidades en la red de datos de la Municipalidad Distrital de Huambos. *(Tesis de Pregrado).* Universidad Señor de Sipán, Pimentel.

Hasheminejad, H. (2015). Honeynet as a service deployment apprroach in enabling virtual crime scene investigation. *(Tesis de Pregrado).* Universidad Putra Malasia, Malasia.

Heredia, C. (2015). Diseño e Implementación de una Honeynet para la red de datos de la Universidad Nacional de Loja, utilizando software Libre. *(Tesis de Pregrado).* Universidad Nacional de Loja, Loja.

Kumar, D., & Girdhar, A. (2017). Network Monitoring & Analysis along with Comparative Study of honeypots. *Actas de la Conferencia Internacional sobre Sistemas Inteligentes Sostenibles*, 736 - 739.

M. Kendrick, M., & A. Rucker, Z. (2019). Energy-grid threat analysis using honeypots. *(Tesis de Postgrado).* Naval Postgraduate School, Monterey.

Martinez, K. (2018). Honeypot, hacia un protocolo de seguridad más eficiente y competitivo. *(Anteproyecto de Grado).* Universidad Nacional Abierta y a Distancia, Colombia.

Matías, J., & Gabriel, M. (2015). Análisis y desarrollo de mejoras a un sistema honypot para mitigar ataques en servicios de VoIP. *Memoria Investigaciones en Ingeniería, núm. 13*, 63-78.

Matus, J. (2017). Análisis e Implementación de una Solución Honeypot para un Entorno Experimental. *(Tesis de Pregrado).* Universidad de Quintana Roo, Chetumal Quintana Roo.

Mohsin, H., Flaih, N., & A., A. (2017). Designing a smartphone honeypot system using performance counters. *Karbala International Journal of Modern Science 3*, 46 - 22.

N., B., & B., A. (2018). Intrusion Detection Using Honeypots. *5th IEEE International Conference on Parallel*, 412 - 417.

Palmay, M. (2017). Propuesta de mejores prácticas para el establecimiento de políticas de seguridad informática basado en honeypot virtuales. *(Tesis de Pregrado).* Escuela Superior Politécnica de Chimborazo, Riobamba.

Sekar, K., Gayathri, V., Anisha, G., Ravichandran, K., & Manikandan, R. (2018). Dynamic Honeypot Configuration for Intrusion Detection. *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics*, 1397 - 1401.

Sembiring, I. (2016). Implementation of Honeypot to Detect and Prevent Distributed Denial of Service Attack. *Proc. of 2016 3 rd Int. Conf. on Information Tech., Computer, and Electrical Engineering (ICITACEE)*, 345-350.

Valdiviezo, J. (2020). Desarrollo de una Red Honeypot para la Detección de Intrusiones en la Municipalidad Distrital de Víctor Larco Herrera - Trujillo. *(Tesis de Pregrado).* Universidad César Vallejo, Trujillo.

Vaño, F. (2016). Configuración, Ampliación e Implantación de un Honeypot. *(Tesis de Postgrado).* Universidad Carlos III de Madrid, Madrid.

Vargas, J. (2015). Honeypot como herramienta de prevención y detección de ciberataques en las redes de datos de la facultad de Ingeniería en Sistemas, Electrónica e Industrial. *(Tesis de Pregrado).* Universidad Técnica de Ambato, Ambato.

Velásquez, J. (2019). Honeypot inteligente de dificultad incremental. *(Tesis de Pregrado).* Universidad EIA, Colombia.

Yucta, B. (2019). Implantación de un aplicativo para optimizar la gestión centralizada de logs en un ambiente honeynet en el datacenter UNACH. *(Tesis de Pregrado).* Universidad Nacional de Chimborazo, Riobamba.

ANEXO 1.

Resolución de aprobación del proyecto de investigación

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°2325-2020/FIAU-USS

Pimentel, 17 de noviembre de 2020

VISTO:
El oficio N°0237-2020/FIAU-IS-USS de fecha 12 de noviembre de 2020, de la Dirección de Escuela profesional de INGENIERÍA DE SISTEMAS con el que remite el Acta de reunión N°2610-2020 del Comité de investigación de la referida Escuela profesional, para el desarrollo de la Tesis presentada por estudiantes del Programa de estudios de INGENIERÍA DE SISTEMAS, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48º que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y *tesis* son *aprobados por el Comité de Investigación* y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El *periodo de vigencia de los mismos será de dos años*, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24º señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; *es individual o en pares para obtener un título profesional*. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.".

Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar la ampliación de la vigencia de las tesis que se detallan en el Acta de reunión N°2610-2020, de la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y AMBIENTE, a cargo de estudiantes del Programa de estudios INGENIERÍA DE SISTEMAS.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO ÚNICO: AMPLIAR VIGENCIA, de la Tesis a cargo de los estudiantes del Programa de estudios de **INGENIERÍA DE SISTEMAS** que se detallan en el anexo de la presente Resolución, perteneciente a la línea de investigación de **INFRAESTRUCTURA, TECNOLOGÍA Y AMBIENTE**, hasta el **31 de julio de 2021.**

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE

Dr. Mario Fernando Ramos Mosol
Decano - Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

MBA. María Noelia Slaler Rivera
Secretaria Académica / Facultad de Ingeniería, Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Cc: Interesado, Archivo

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°2325-2020/FIAU-USS

Pimentel, 17 de noviembre de 2020

ANEXO

N°	APELLIDOS Y NOMBRES	TEMA DE TESIS
1	DELGADO VALVERDE JALLY JACQUELINE	COMPARACIÓN DE ALGORITMOS DE CLASIFICACIÓN EN MINERÍA DE DATOS PARA EL ANÁLISIS DE VENTAS EN EMPRESAS DE TRANSPORTE INTERPROVINCIAL DE PASAJEROS
2	ALVAREZ CAIPO DONAR JESUS	EVALUACIÓN DE HERRAMIENTAS OPEN SOURCE PARA LA SEGURIDAD PERIMETRAL EN DATECENTERS DEL SECTOR FARMACIAS
3	VALLEJOS RODRIGUEZ JAIR ADBEEL	COMPARACIÓN DE ALGORITMOS DE DETECCIÓN DE BORDES Y VECTORIZACIÓN DE IMÁGENES DE MOLDES TEXTILES
4	GALLARDO GUTIERREZ SANDRA ISABEL	MODELO DE GESTIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE BASADO EN GUÍAS DE BUENAS PRÁCTICAS Y METODOLOGÍAS ÁGILES. CASO DE ESTUDIO "INGENIEROS Y SOLUCIONES SRL"
5	LOPEZ LOZANO YAHAIRA CYNTHIA FIORELLA	DESARROLLO DE UN MODELO DE GESTIÓN DE SERVICIOS DE TI PARA MEJORAR LOS PROCESOS DE ATENCIÓN A CLIENTES EN ENTIDADES FINANCIERAS PERUANAS DE TIPO CAJA DE AHORRO Y CRÉDITO
6	NEYRA PRADO ERNESTO FRANCISCO	EVALUACIÓN DEL RENDIMIENTO DE LAS TÉCNICAS DE CIFRADO DE DATOS EN LA COMUNICACIÓN DE DISPOSITIVOS DE INTERNET DE LAS COSAS CON SERVICIOS DE CLOUD COMPUTING
7	TUESTA CASTILLO JAIR SLEYTER	IDENTIFICACIÓN AUTOMÁTICA DE ÁRBOLES DE CERATONIA SILIQUIA (ALGARROBO) DEL BOSQUE DE POMAC UTILIZANDO IMÁGENES DIGITALES
8	ROJAS FLORES WALTER ALONSO	ANÁLISIS COMPARATIVO DE ALGORITMOS DE CLASIFICACIÓN PARA LA DETECCIÓN DE MICRO CALCIFICACIONES EN IMÁGENES MAMOGRÁFICAS
9	JIMENEZ ESPINOZA LUIS ALBERTO	ANÁLISIS DE MÉTODOS DE RECONOCIMIENTOS DE EXPRESIONES FACIALES
10	CAMPOVERDE VEGA VICTOR ANDREE	ANÁLISIS COMPARATIVO DE RENDIMIENTO EN GESTORES DE BASES DE DATOS RELACIONALES Y NO RELACIONALES
11	VENEGAS BRAVO JOSE ALEXANDER	ANÁLISIS COMPARATIVO DE RENDIMIENTO DE GESTORES DE BASE DE DATOS NOSQL DOCUMENTALES
12	CHAVEZ VALDEZ JORGE LUIS	ANÁLISIS COMPARATIVO DE HERRAMIENTAS DE EVALUACIÓN DE CALIDAD DE SOFTWARE: PRUEBAS UNITARIAS
13	PAZ COLCHON LUIS ANGEL	ANALISIS DE RENDIMIENTO DE LOS PROTOCOLOS DE COMUNICACIÓN XMPP Y DSS PARA INTERNET DE LAS COSAS
14	CHAPOÑAN SANTISTEBAN DAVID	ANÁLISIS COMPARATIVO DE HONEYPOT DE BAJA INTERACCIÓN HONEYD Y KFSENSOR IMPLEMENTADOS VIRTUALMENTE
15	CLAVO TAFUR CRISTIAN JESUS	ANALISIS COMPARATIVO DE TÉCNICAS DE MITIGACIÓN DE ATAQUE DE DDOS EN CLOUD COMPUTING
16	YARLEQUE REYES JHONATAN GREGORIO	ANÁLISIS DE MÉTODOS DE CLASIFICACIÓN DE MACHINE LEARNING PARA IDENTIFICAR ATAQUES PHISHING
17	DE LA CRUZ GASTELO CINTHYA ARACELI	COMPARACIÓN DE PROTOCOLOS DE REDES PRIVADAS VIRTUALES IPV4 E IPV6 ALÁMBRICAS E INALÁMBRICAS PARA DETERMINAR LA CALIDAD DE SERVICIO
18	GONZALEZ FLORES PAUL GUSTAVO	ANÁLISIS COMPARATIVO DE ALGORITMOS DE CLASIFICACIÓN PARA DIAGNOSTICAR TIPOS DE LEUCEMIA INFANTIL
19	FARRO MEDINA LARRY AUGUSTO	PLATAFORMA DE MONITOREO DE GRANJAS AVÍCOLAS MEDIANTE INTERNET DE LAS COSAS

ANEXO 2

Instrumentos de recolección de datos

Instrumento de Observación **1**				
Ataques	Honeypot Honeyd		Honeypot Kfsensor	
	Consumo CPU (GHz)	Consumo RAM (MB)	Consumo CPU (GHz)	Consumo RAM (MB)
1				
2				
3				
…				
19				
20				
Observaciones:				

ANEXO 3

Instalación del Software de virtualización VMWare y creación de máquinas virtuales

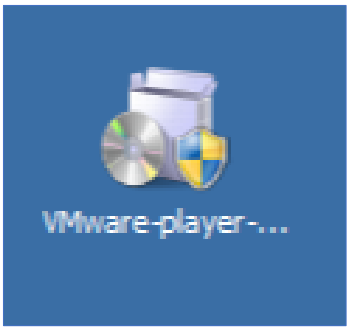


Figura 24. Instalador de VMWare

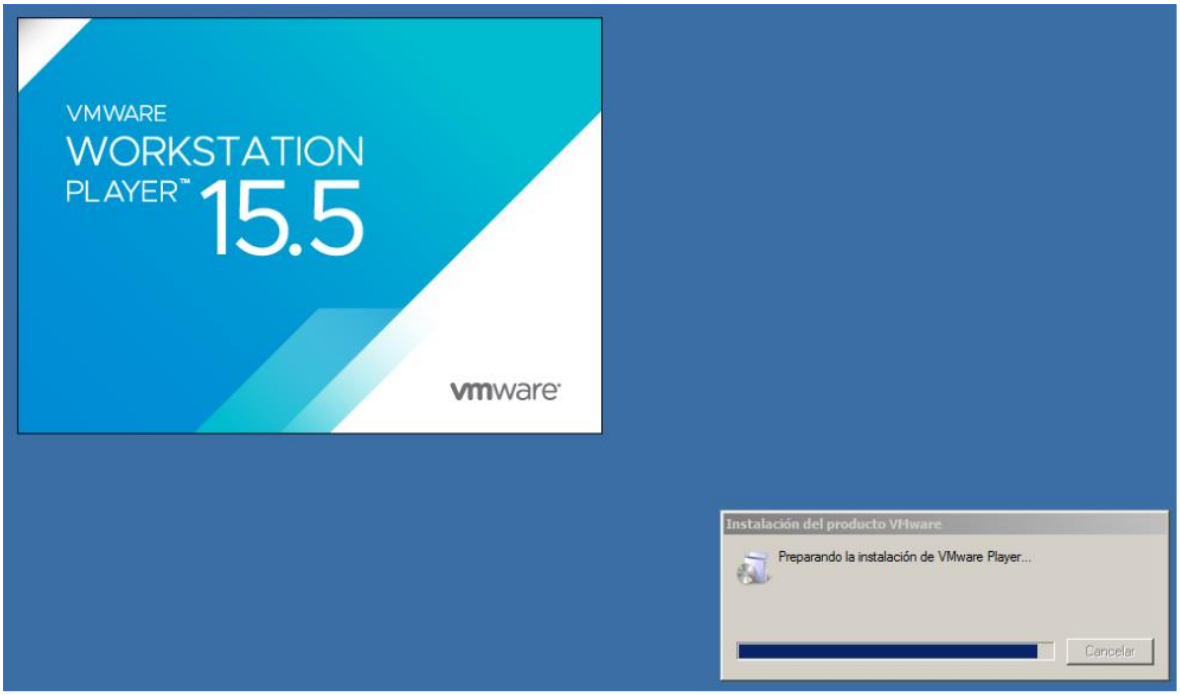


Figura 25. Cargando la instalación de VMWare

Figura 26. Instalador de VMWare listo para seguir indicaciones

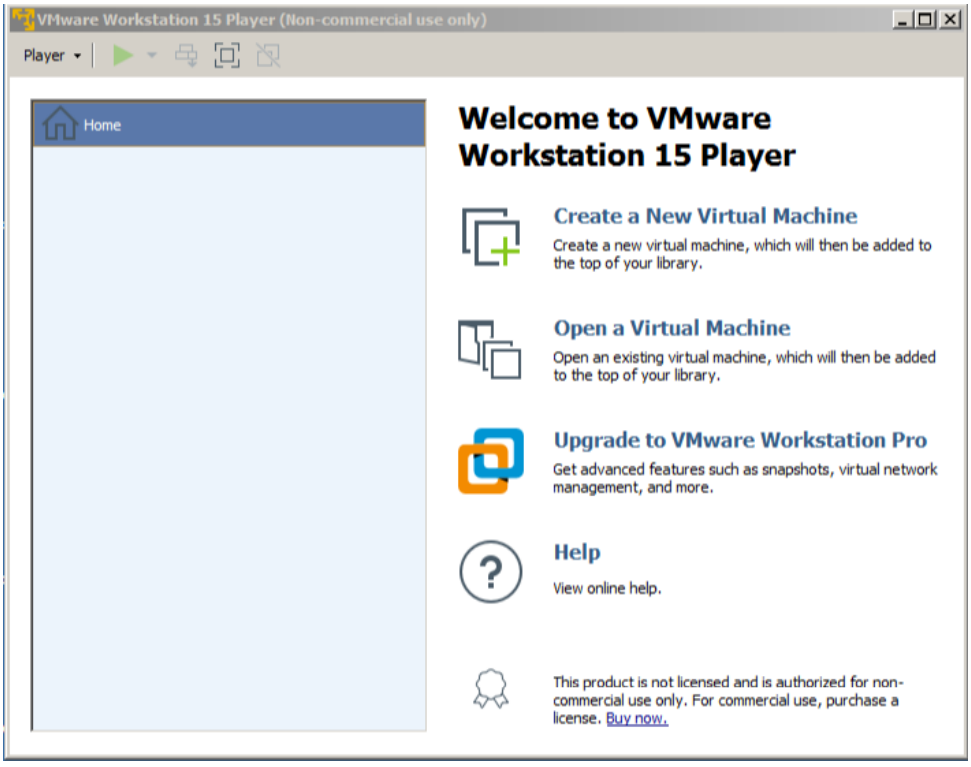


Figura 27. Panel principal del Software VMWare

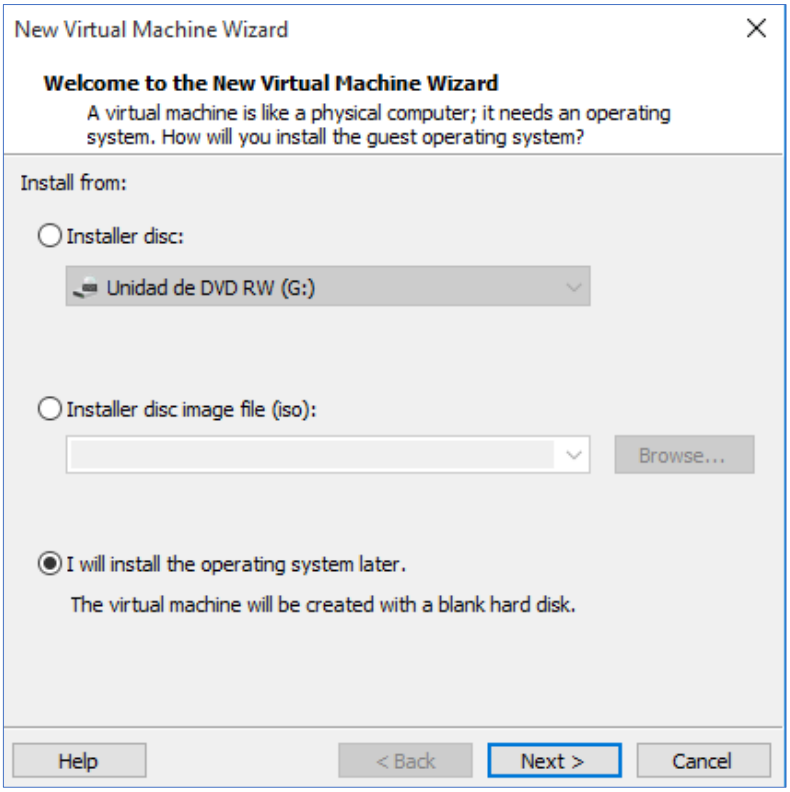


Figura 28. Crear nueva máquina virtual

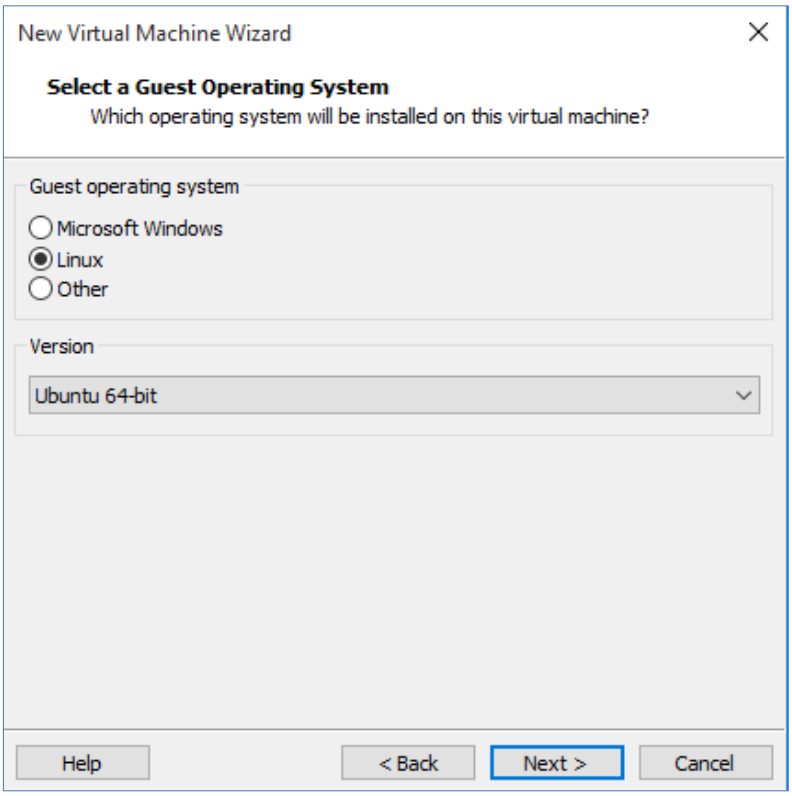


Figura 29. Crear nueva máquina con Sistema Operativo Ubuntu

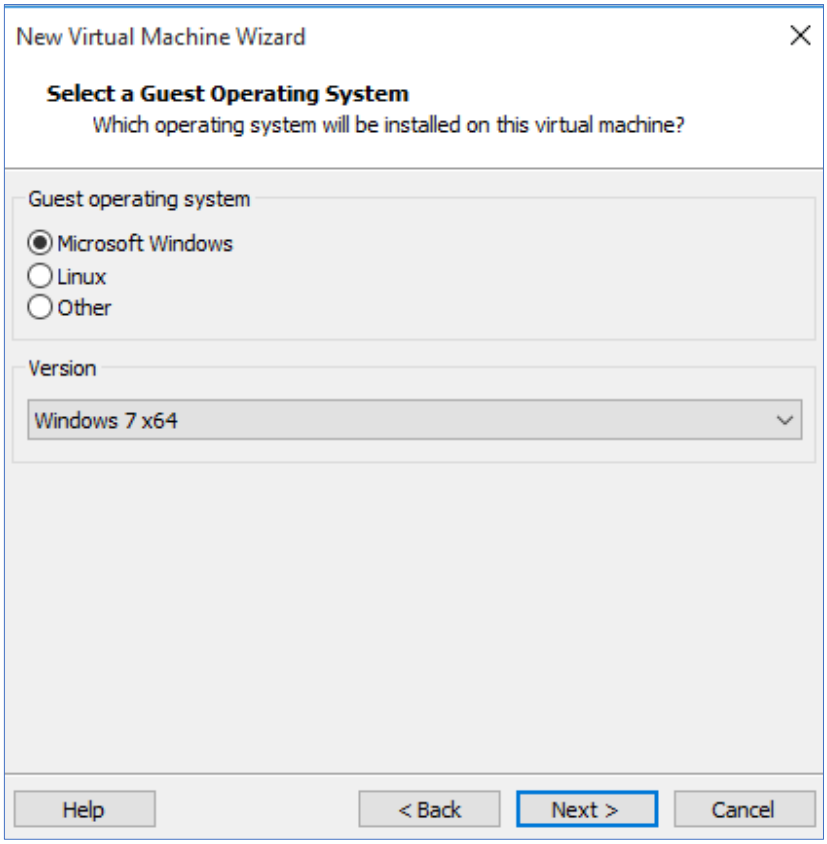


Figura 30. Crear nueva máquina con Sistema Operativo Windows 7

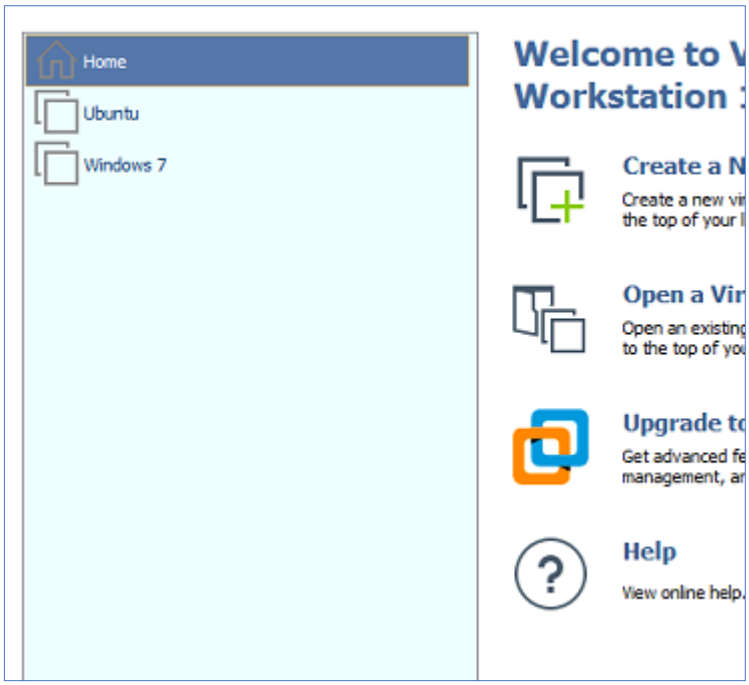


Figura 31. Panel de VMWare con lista de máquinas virtuales

ANEXO 4

Instalación del Sistema Operativo Windows 7

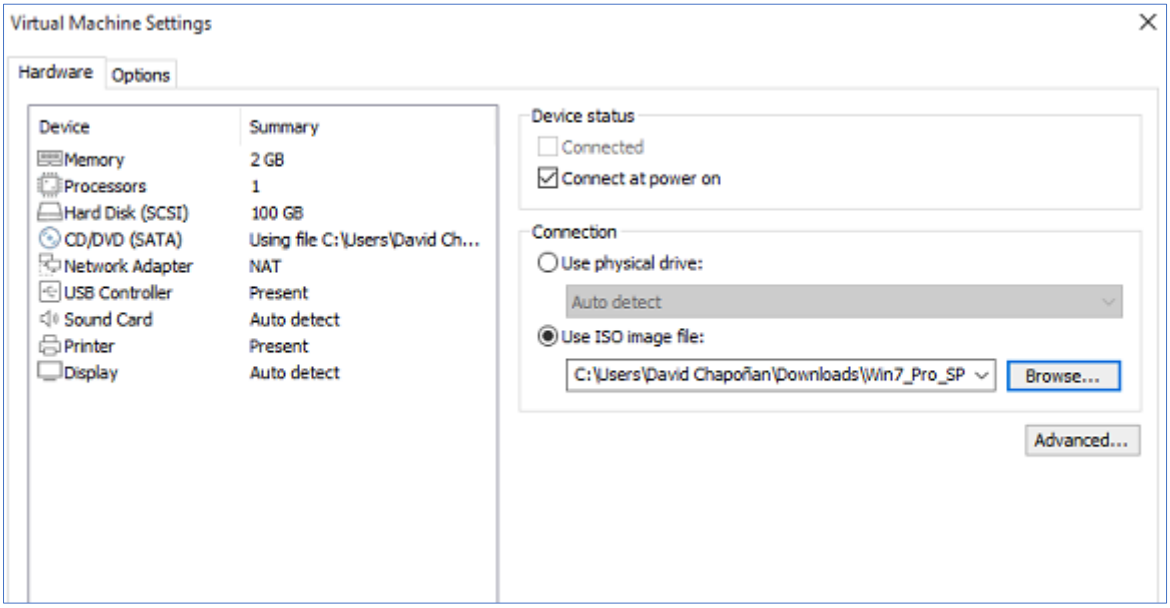


Figura 32. Asignar el disco de instalación virtual de Windows 7

Figura 33. Elegir el idioma del sistema operativo Windows 7

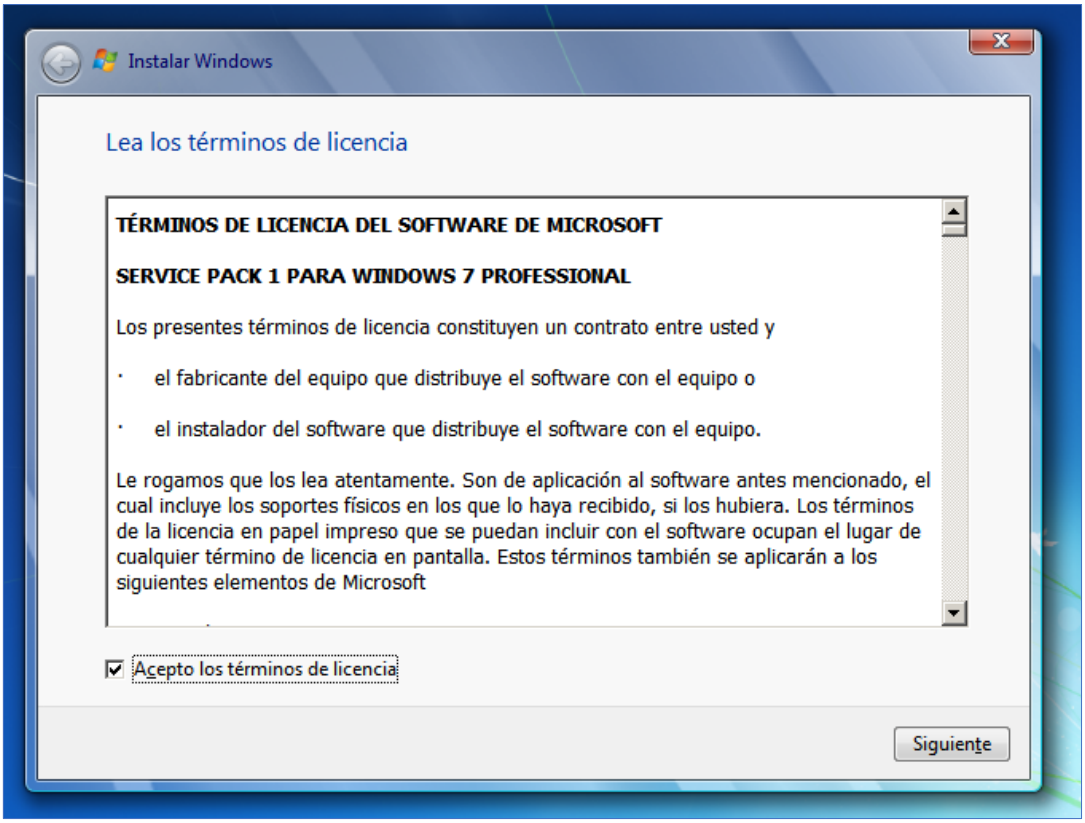


Figura 34. Aceptar los términos de licencia de instalación

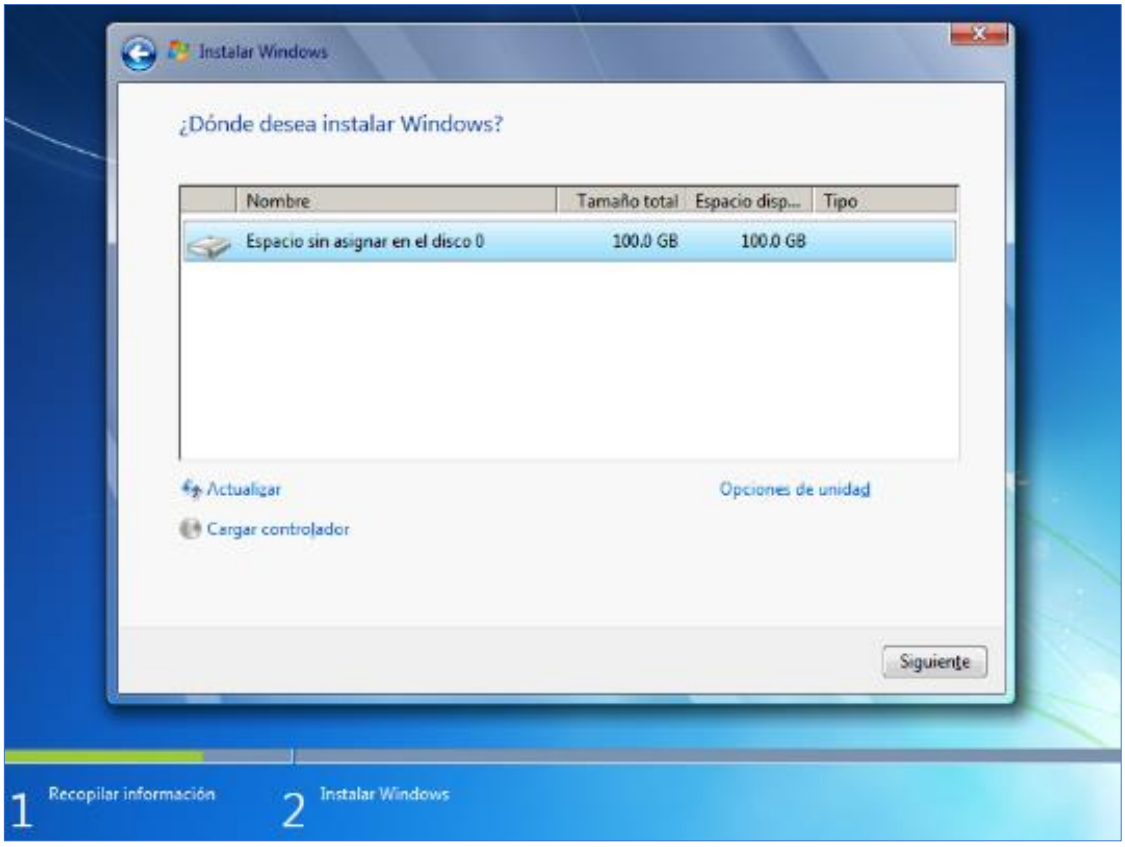


Figura 35. Particionado de disco virtual

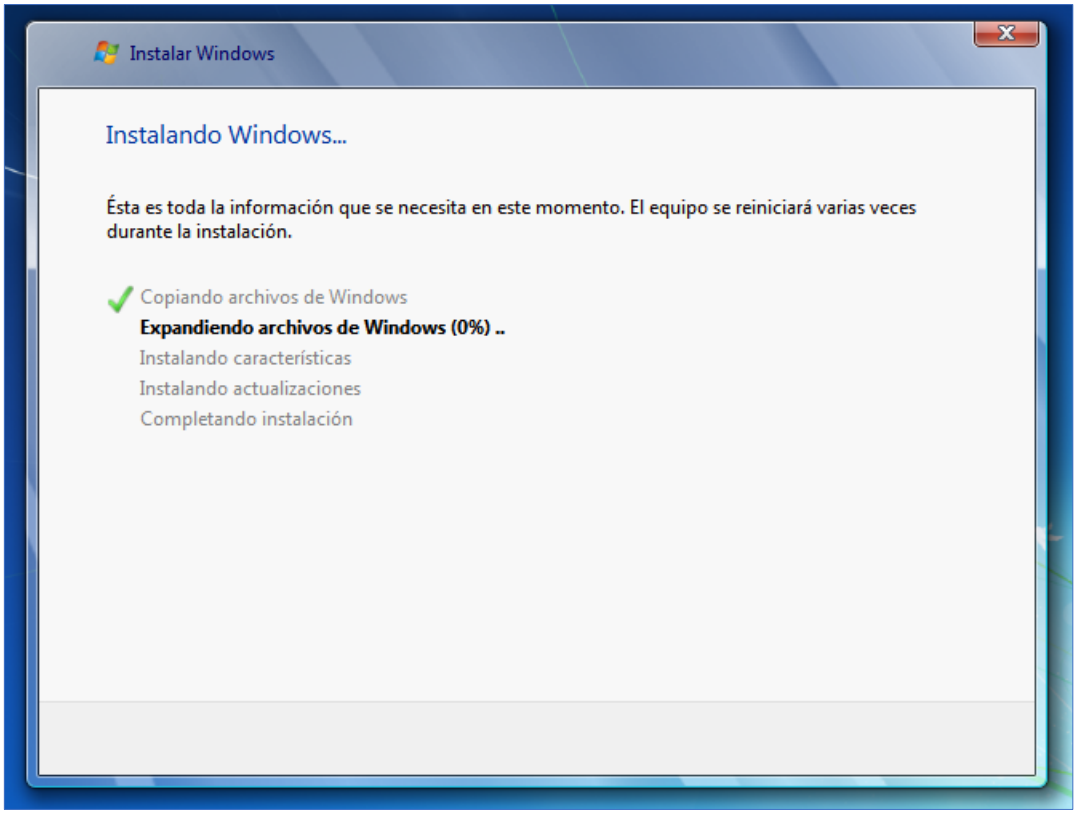


Figura 36. Instalación del Sistema Operativo en curso

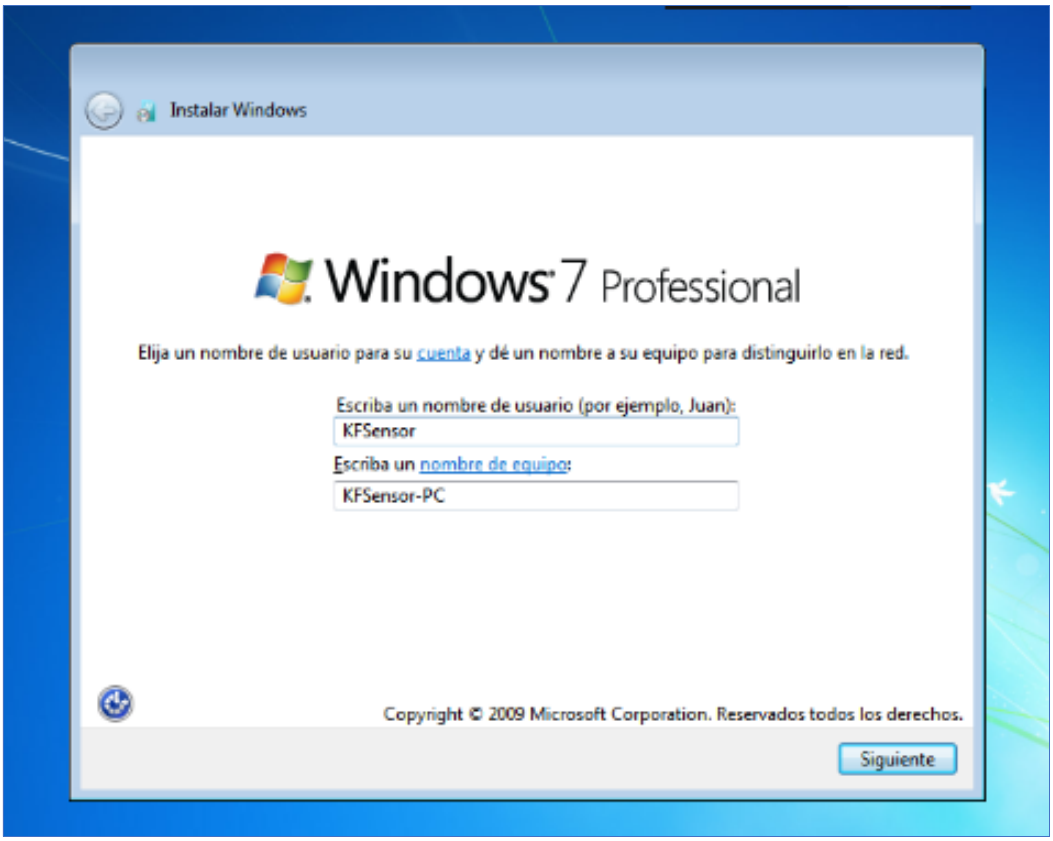


Figura 37. Asignar nombre de equipo y usuario del sistema

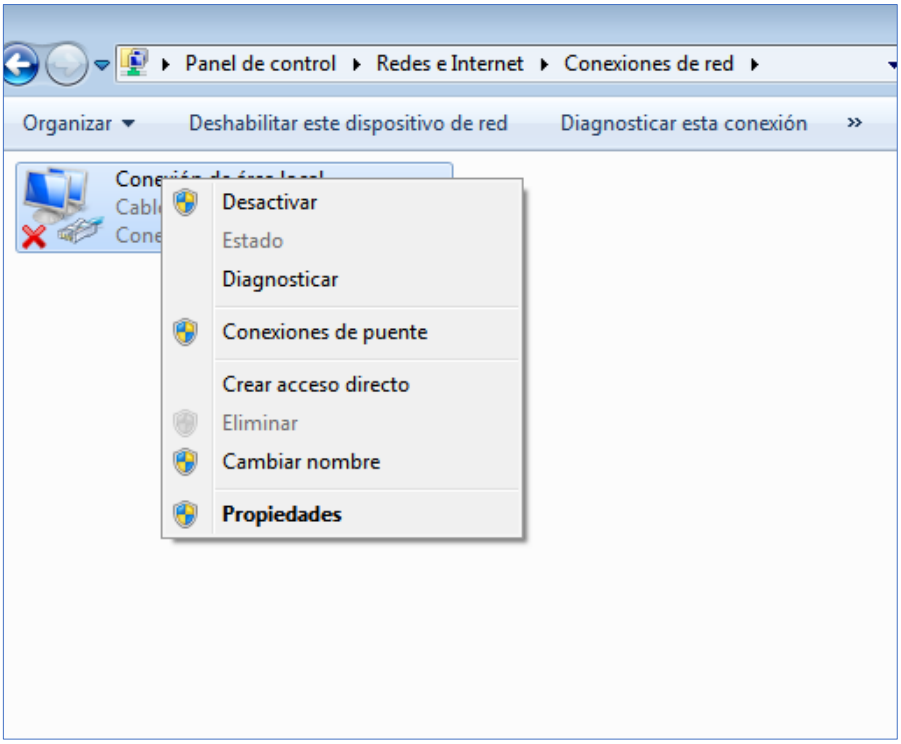


Figura 38. Acceder a configuración de Red

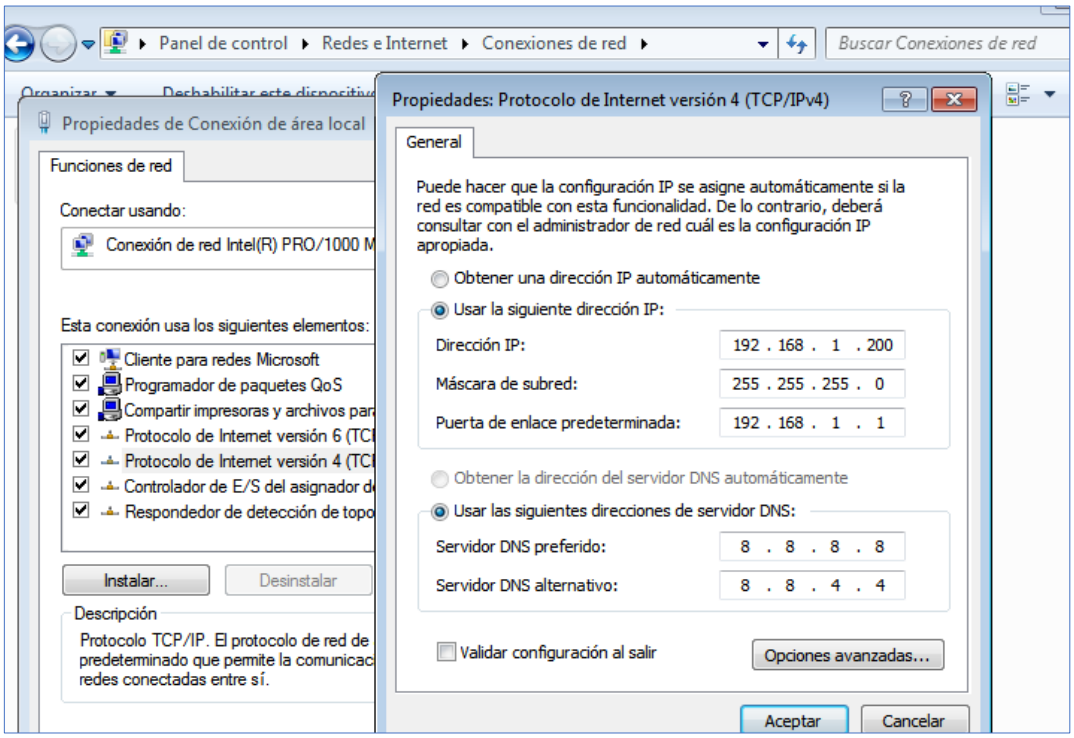


Figura 39. Configuración de red para la máquina virtual

ANEXO 5

Instalación del Sistema Operativo Ubuntu

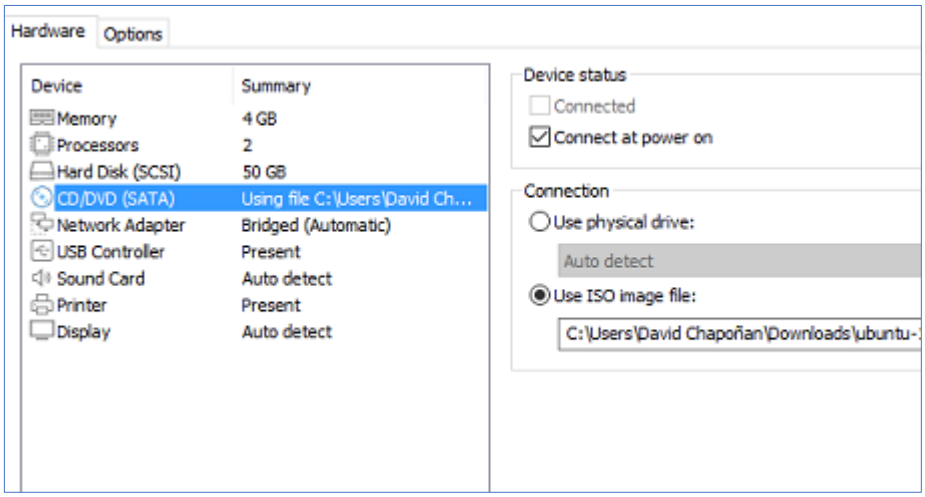


Figura 40. Asignar el disco de instalación virtual de Ubuntu

Figura 41. Instalador de Ubuntu

Figura 42. Elegir el particionado del disco o modo de instalación

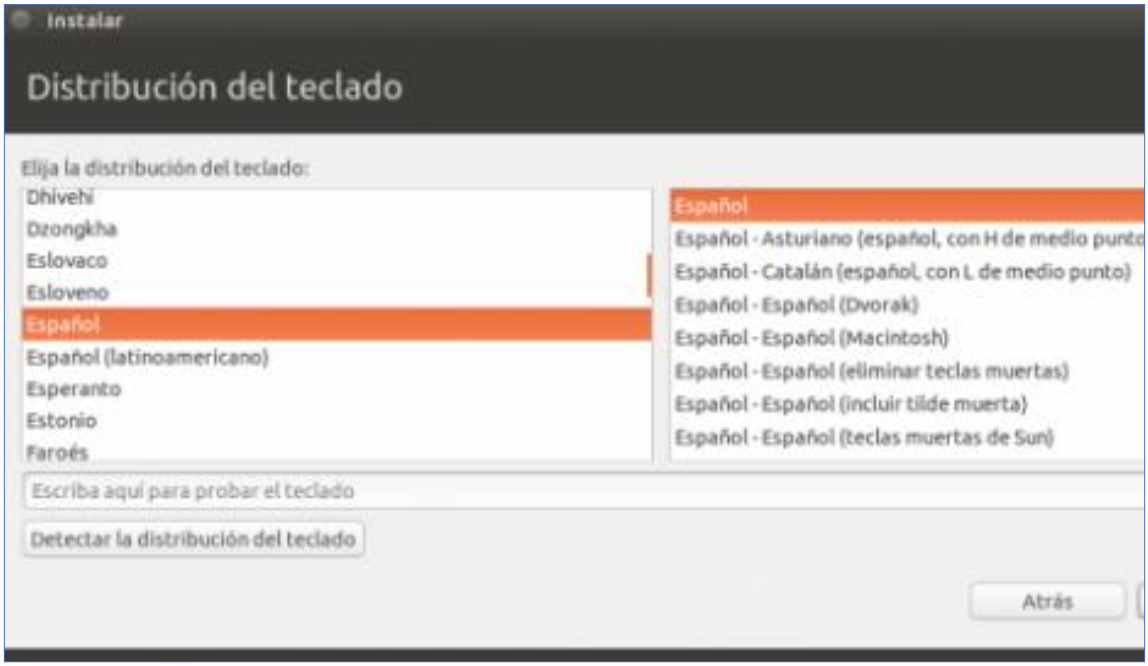


Figura 43. Elegir el idioma del teclado

Figura 44. Asignar el usuario del sistema y su contraseña

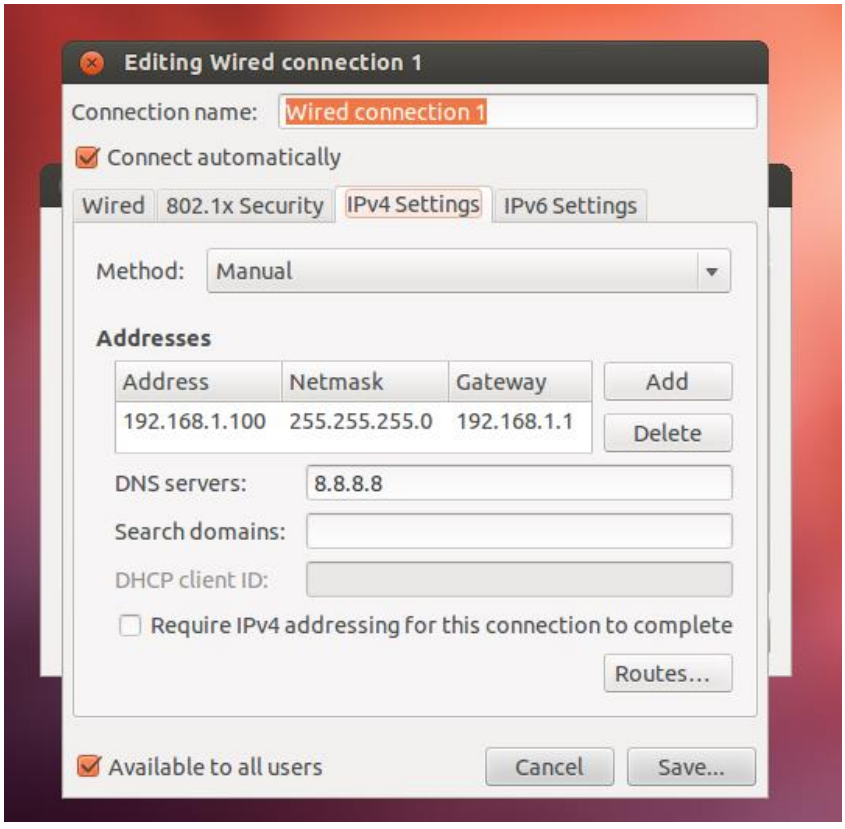


Figura 45. Configuración de Red de la máquina virtual

ANEXO 6

Instalación y configuración del honeypot KFSensor

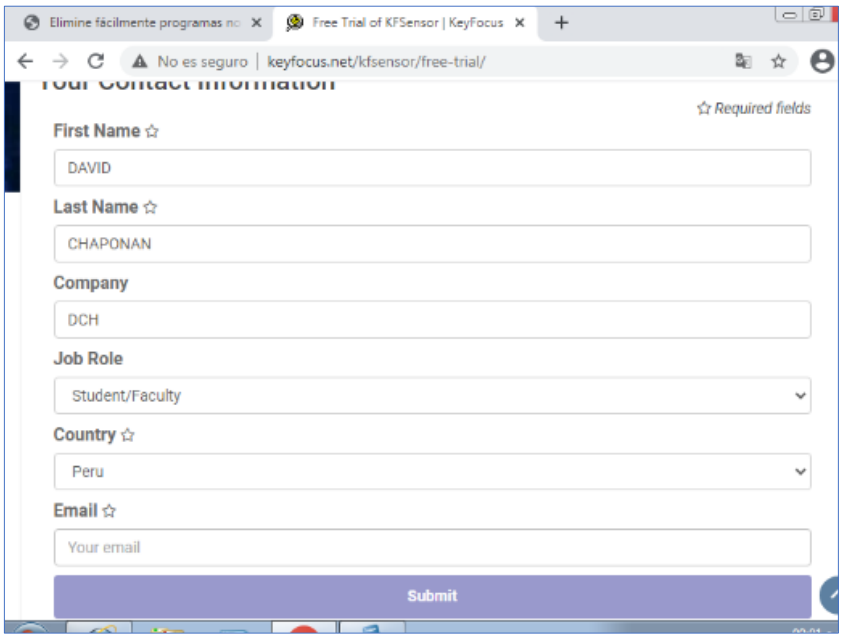


Figura 46. Sitio Web de descarga de versión de prueba de KFSensor

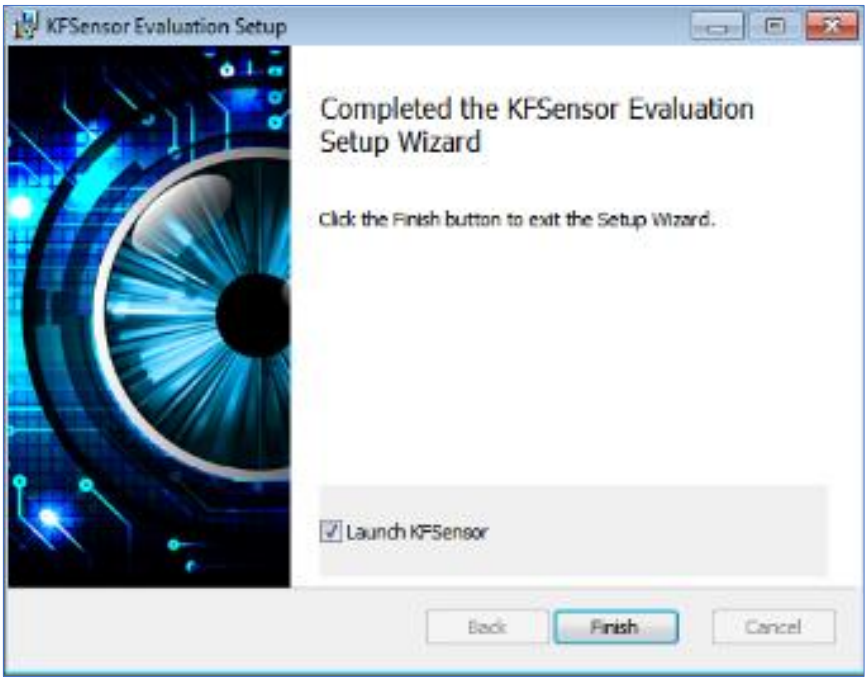


Figura 47. Instalador de KFSensor

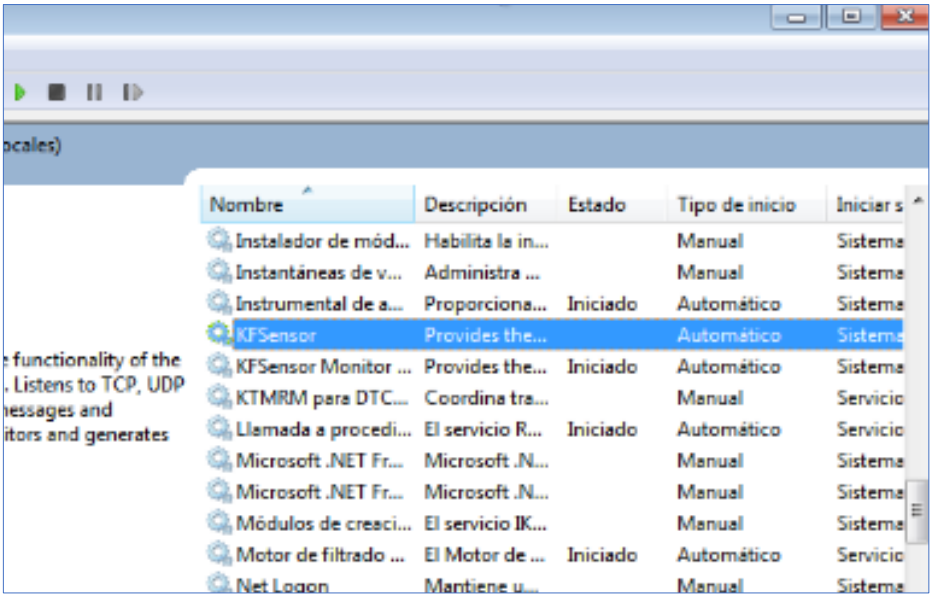


Figura 48. Verificar que el servicio KFSensor se haya instalado

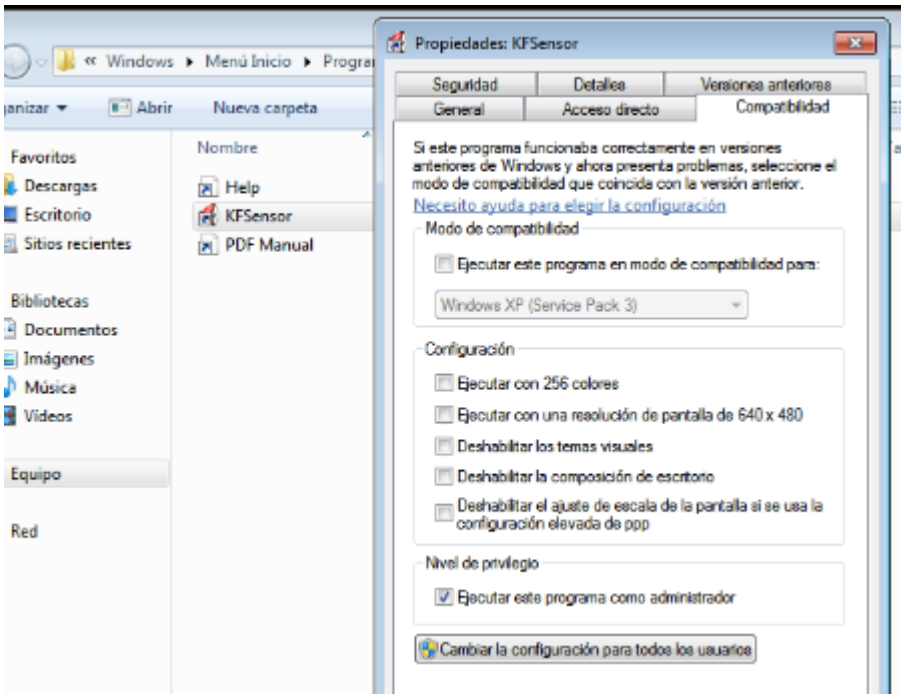


Figura 49. Configurar que KFSensor se ejecute como Administrador

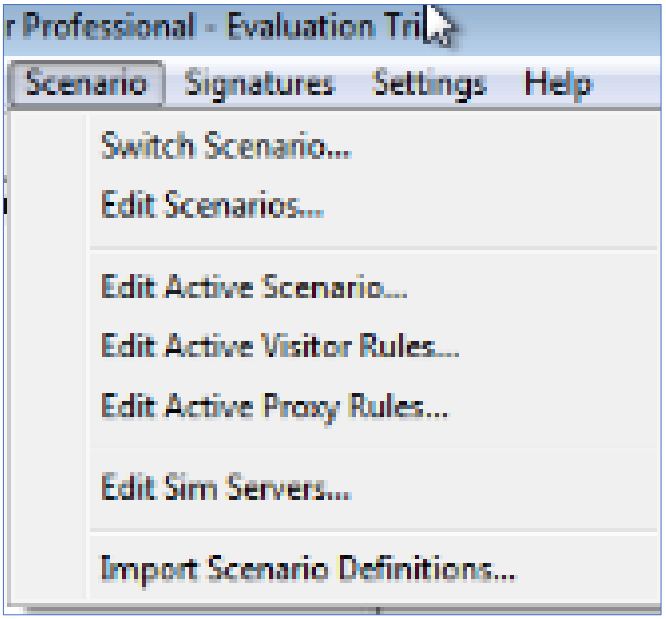


Figura 50. Configurar los escenarios del comportamiento del honeypot KFSensor

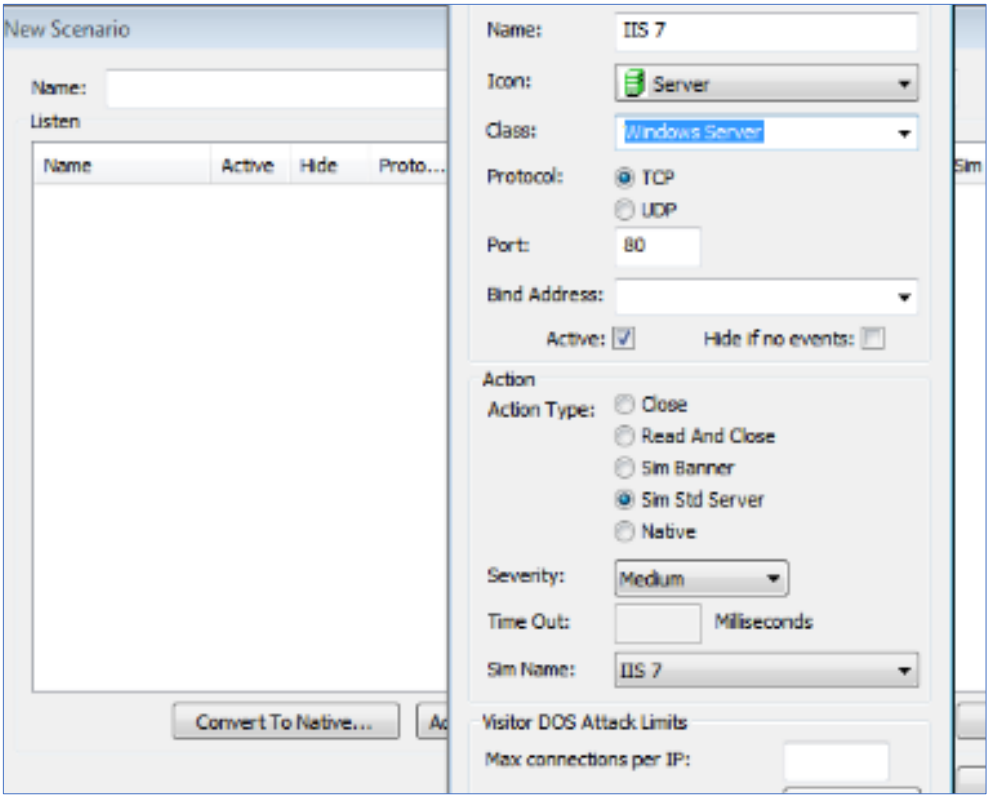


Figura 51. Configurar el escenario con el servicio web en puerto 80

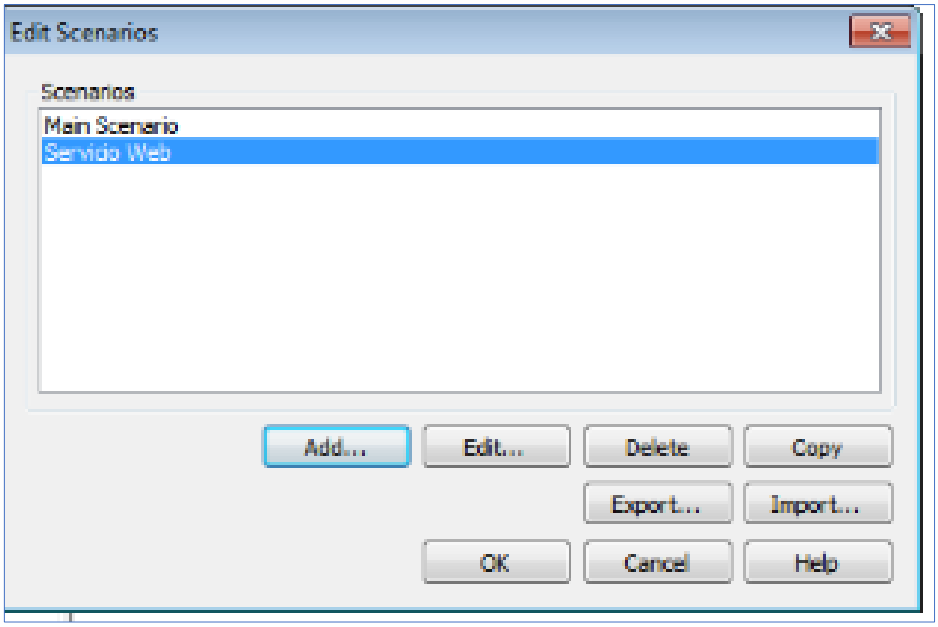


Figura 52. Lista de escenarios configurados

ANEXO 7

Instalación y configuración del honeypot HoneyD

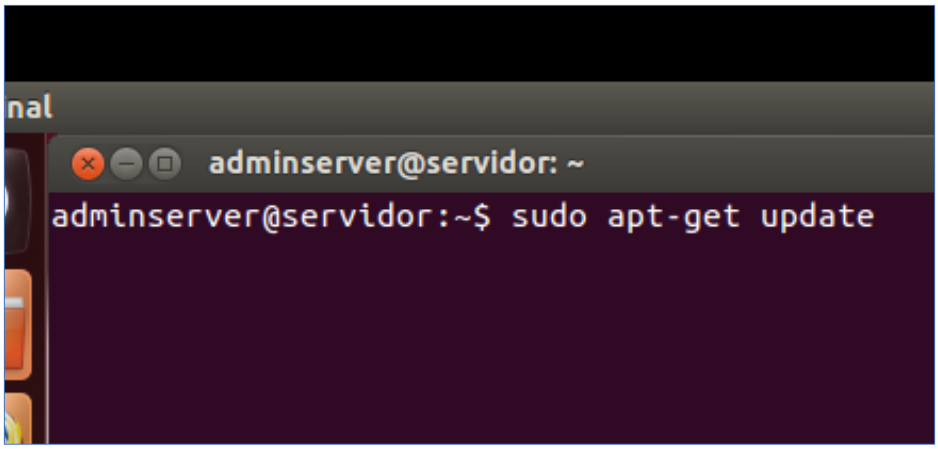


Figura 53. Actualizar repositorios del sistema operativo

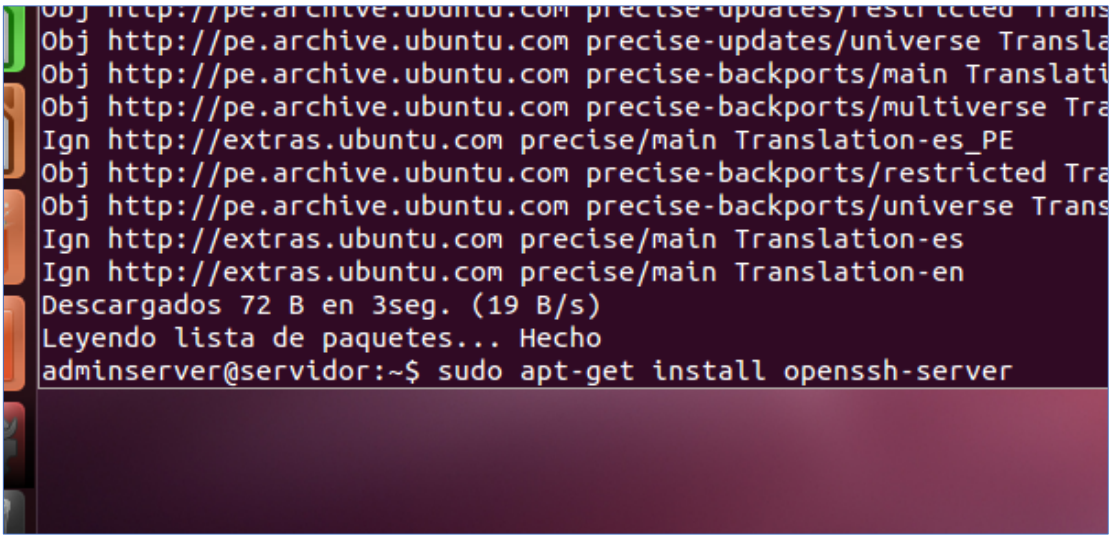


Figura 54. Instalar servicio SSH para entrar a la terminal del sistema remotamente

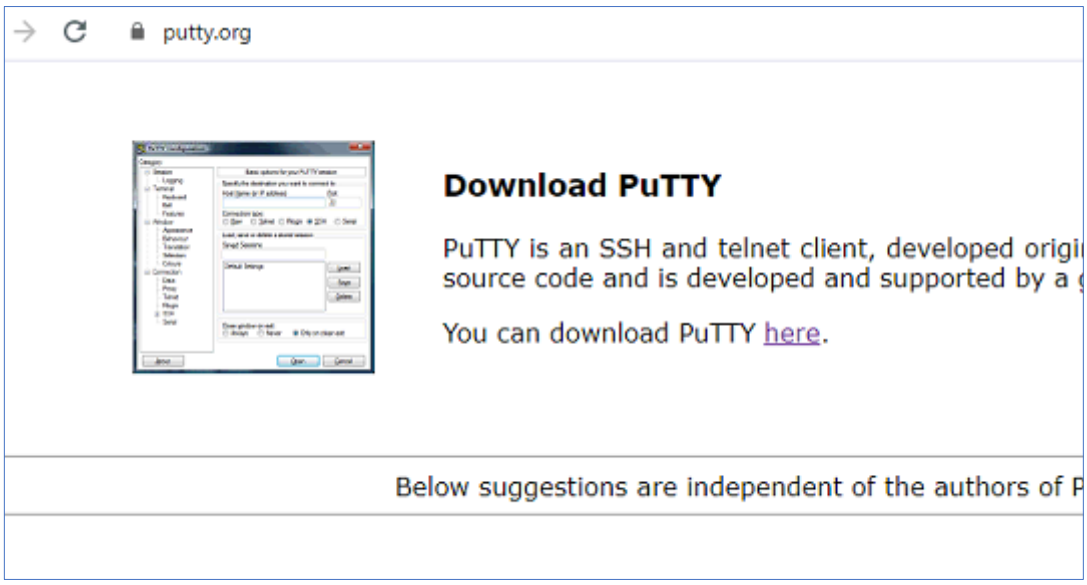


Figura 55. Página de descarga de Cliente SSH llamado Putty

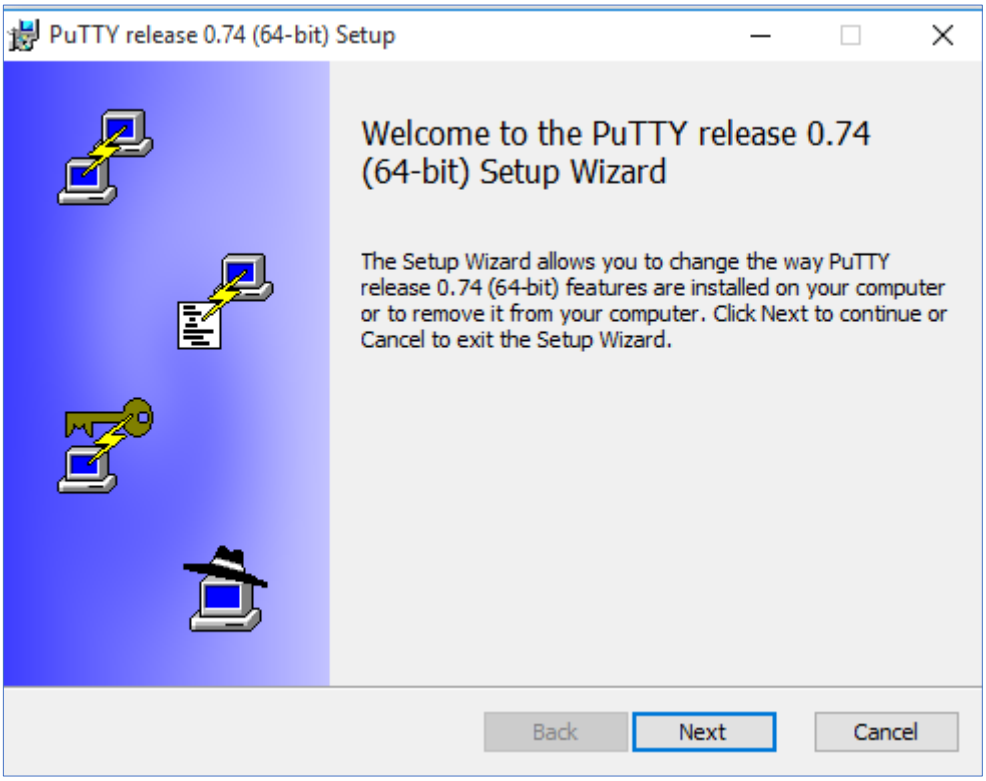


Figura 56. Instalación de Putty, para conectarse al servidor Ubuntu remotamente

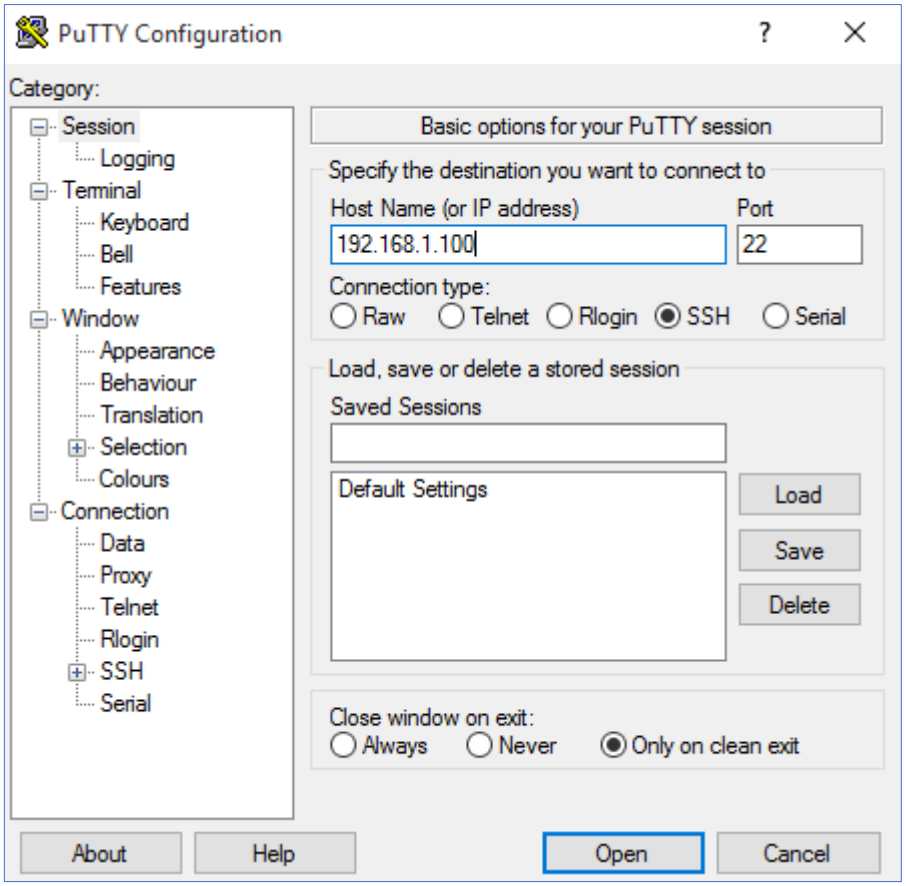


Figura 57. Conexión remota por SSH al servidor usando Putty

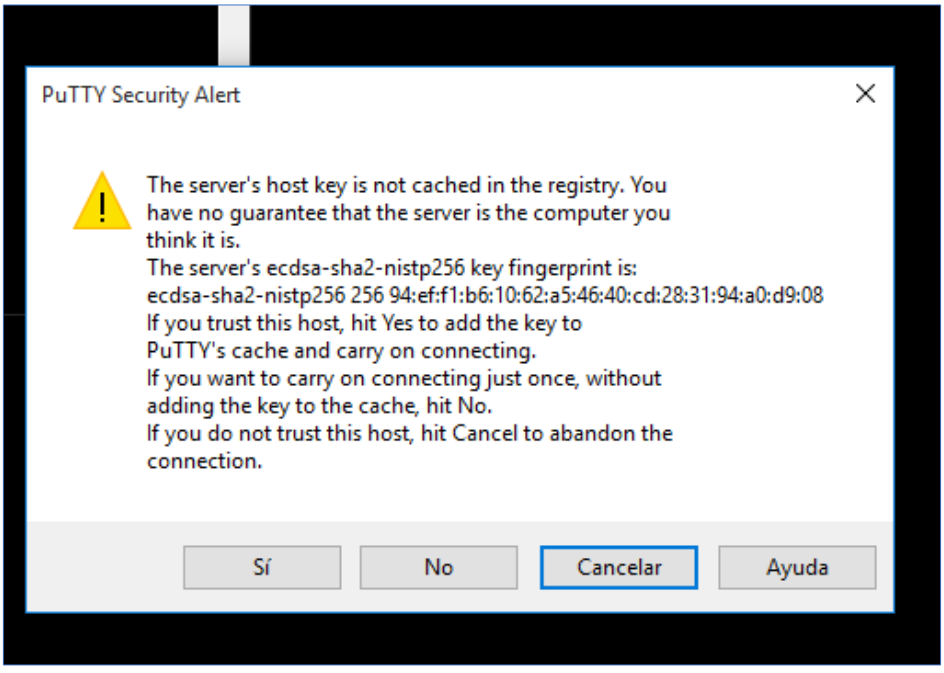


Figura 58. Mensaje de seguridad sobre la conexión SSH

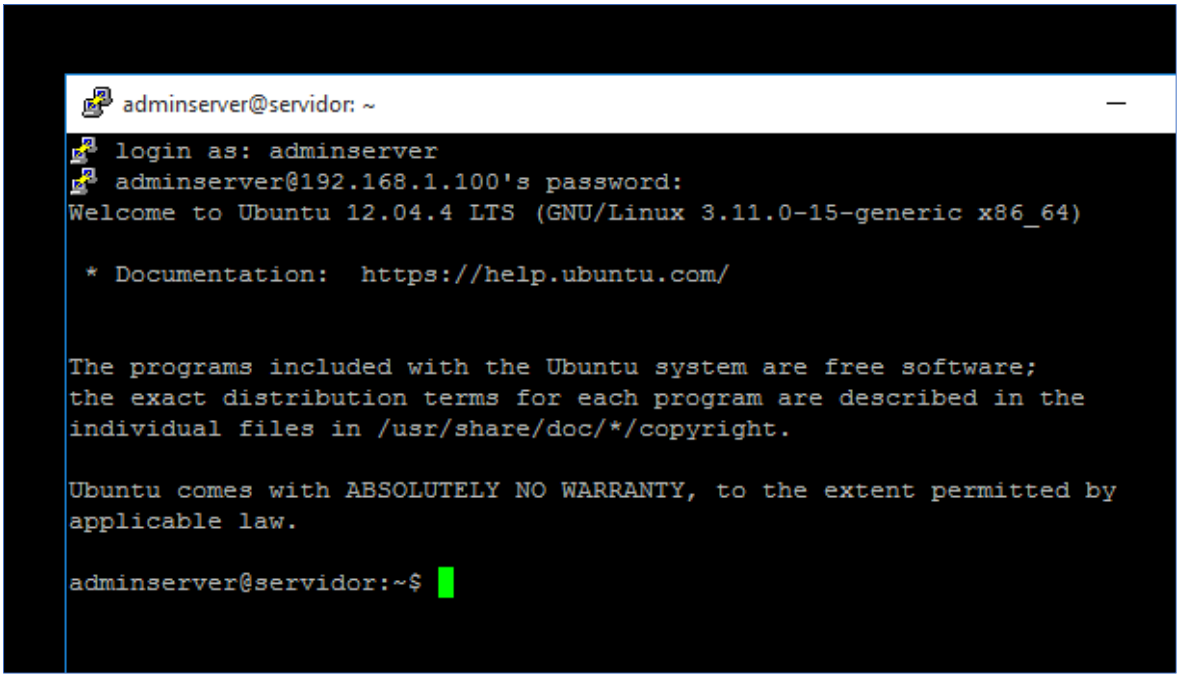


Figura 59. Terminal remota del servidor Ubuntu

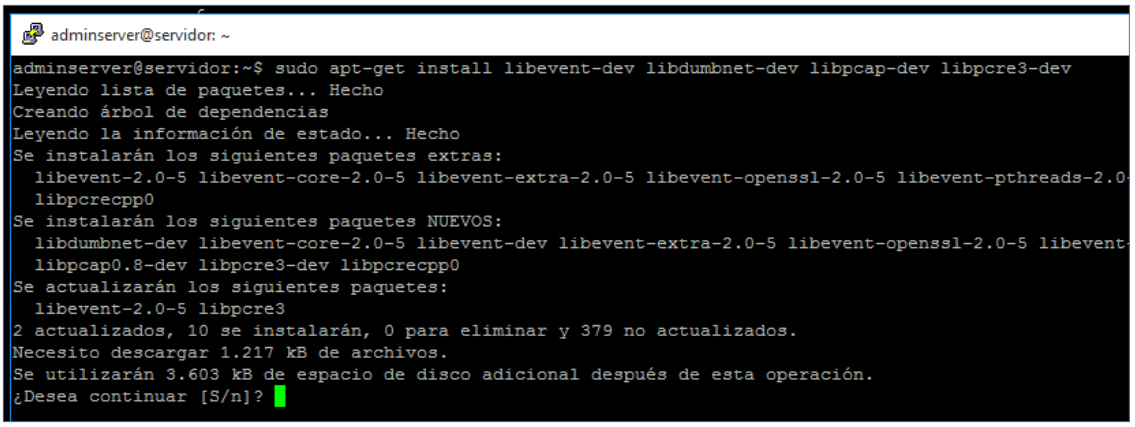


Figura 60. Instalación de librerías que utiliza honeyd

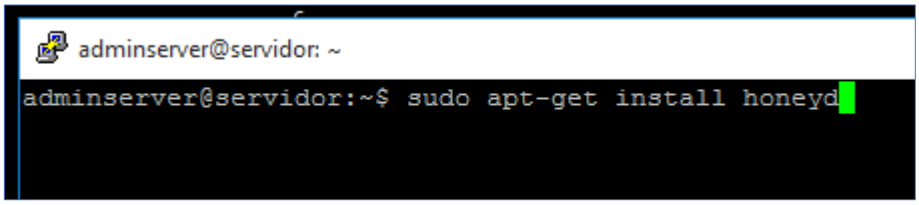


Figura 61. Instalar Honeyd

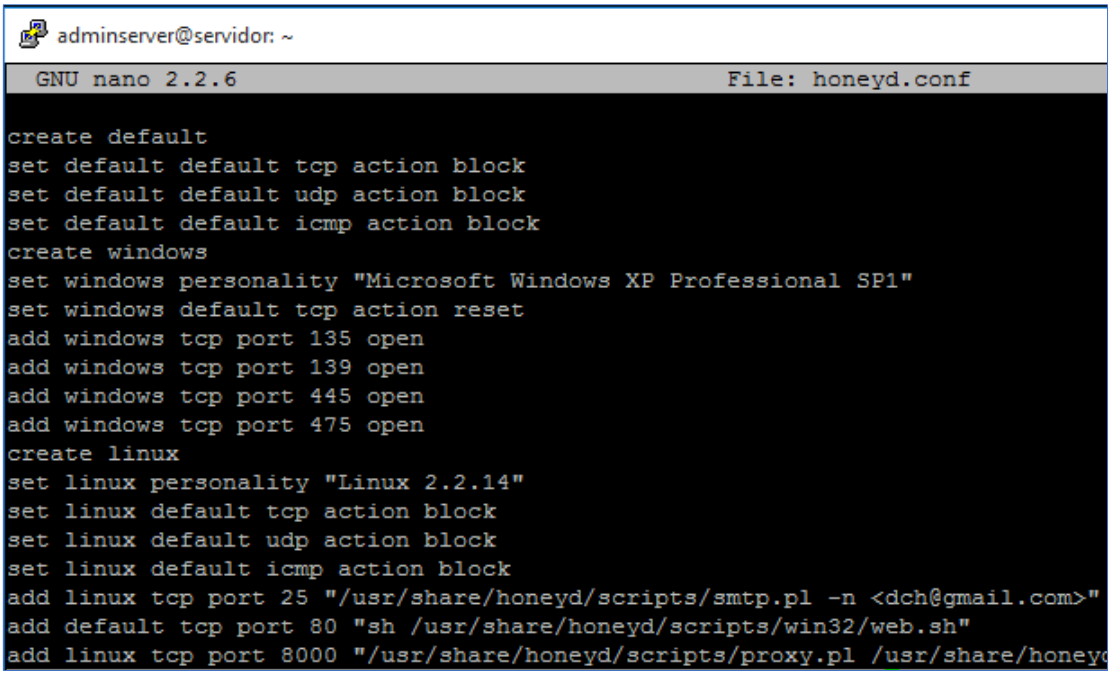


Figura 62. Configurar el escenario con el servicio web en el puerto 80

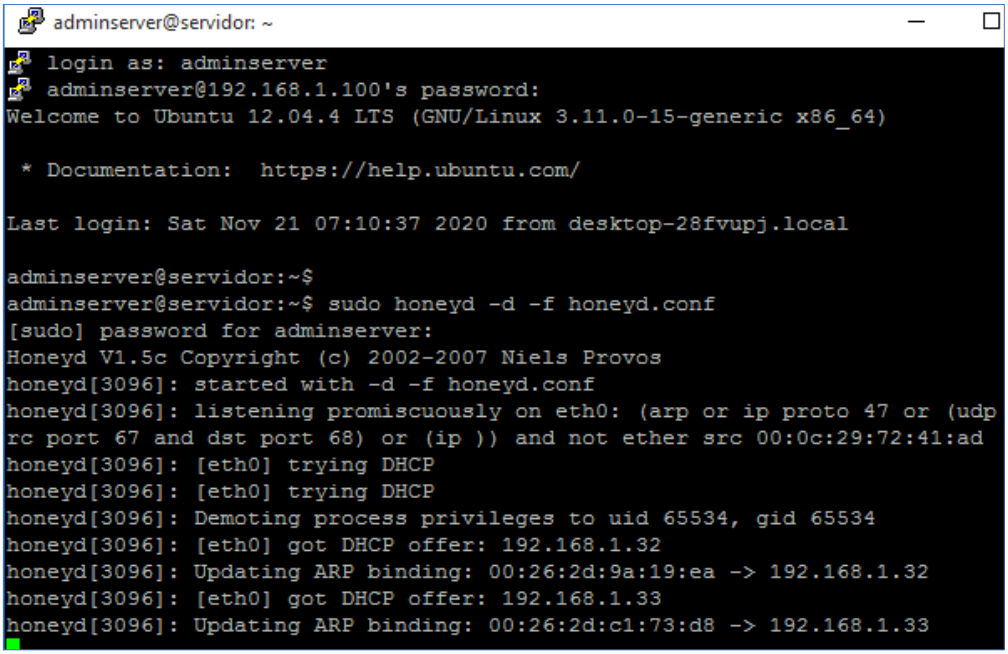


Figura 63. Ejecutar honeyd

ANEXO 8

Pruebas controladas de ataques DDOS, además recolección de consumo de CPU y consumo de memoria RAM en el honeypot KFSensor

Plan de ataques

Virtualizar Maquina atacante

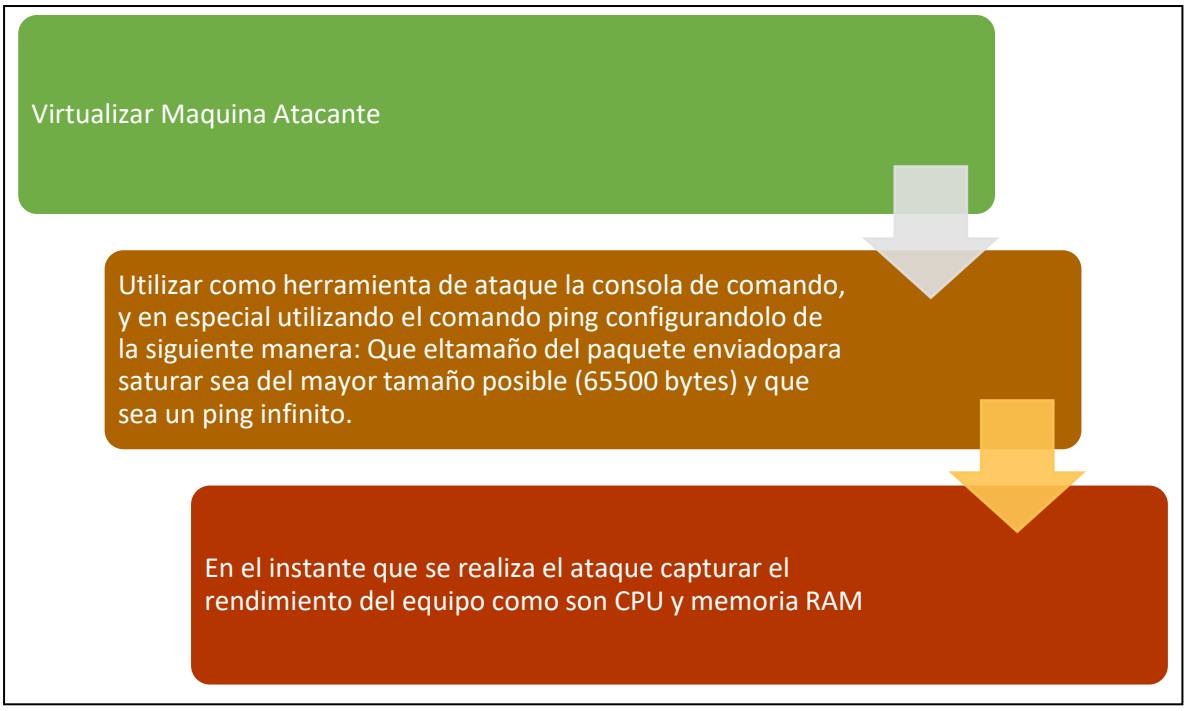


Figura 73. Plan de ataques

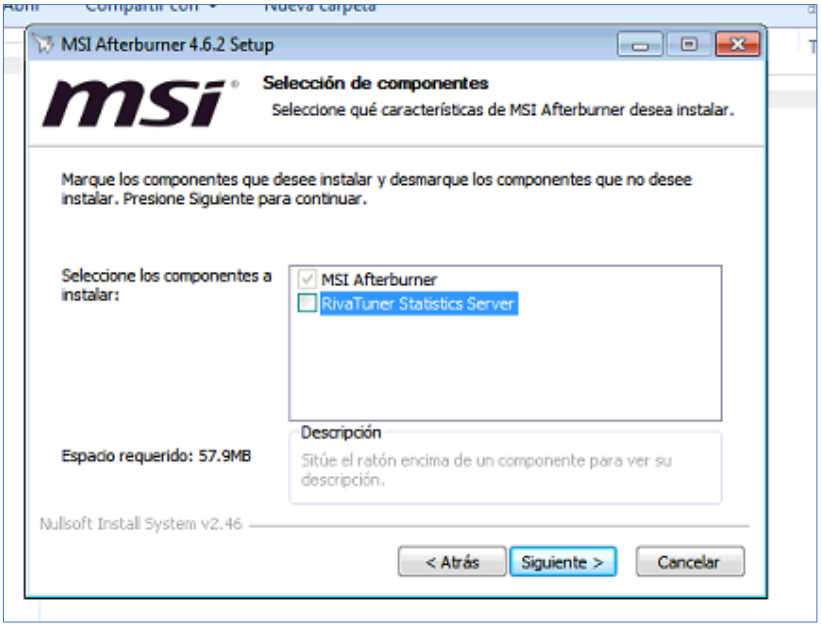


Figura 64. Instalador de MSI Afterbunner

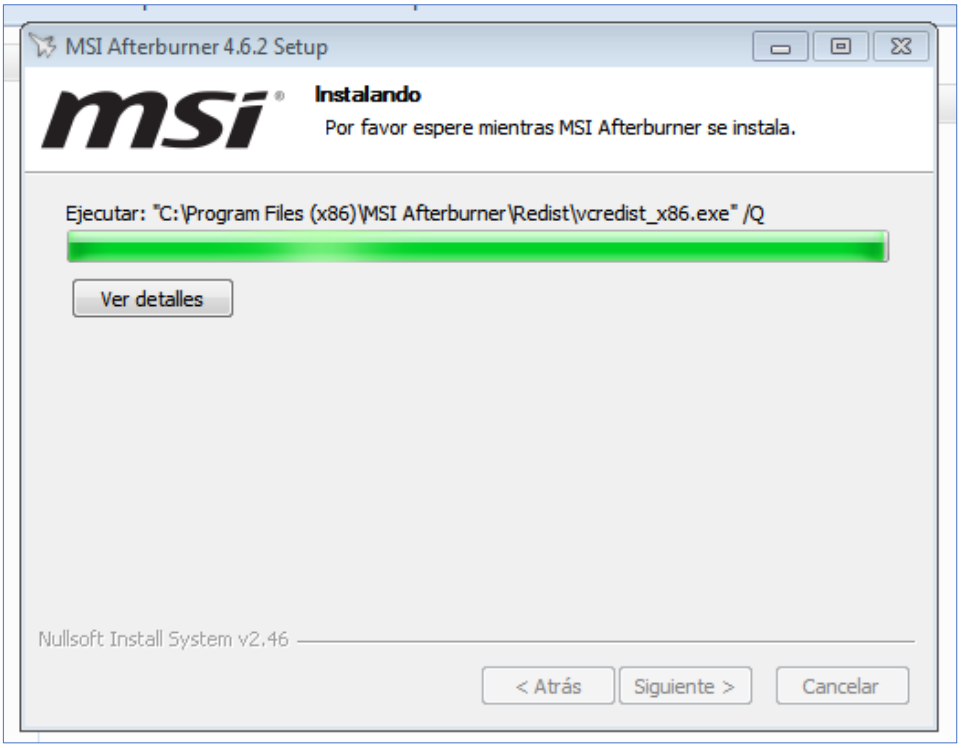


Figura 65. Proceso de instalación de MSI Afterburner

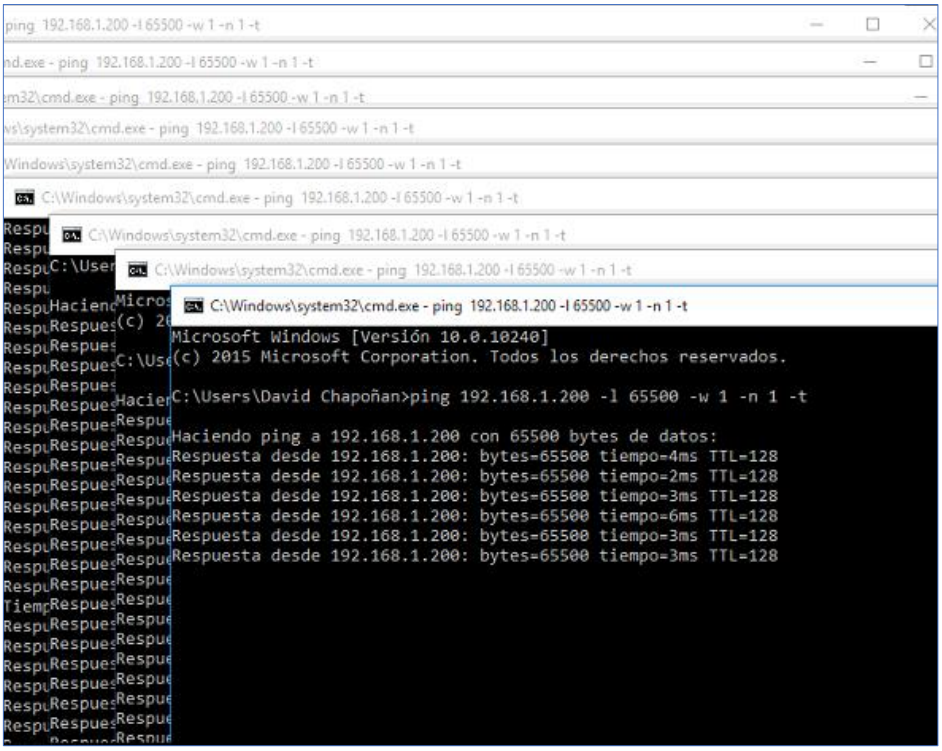
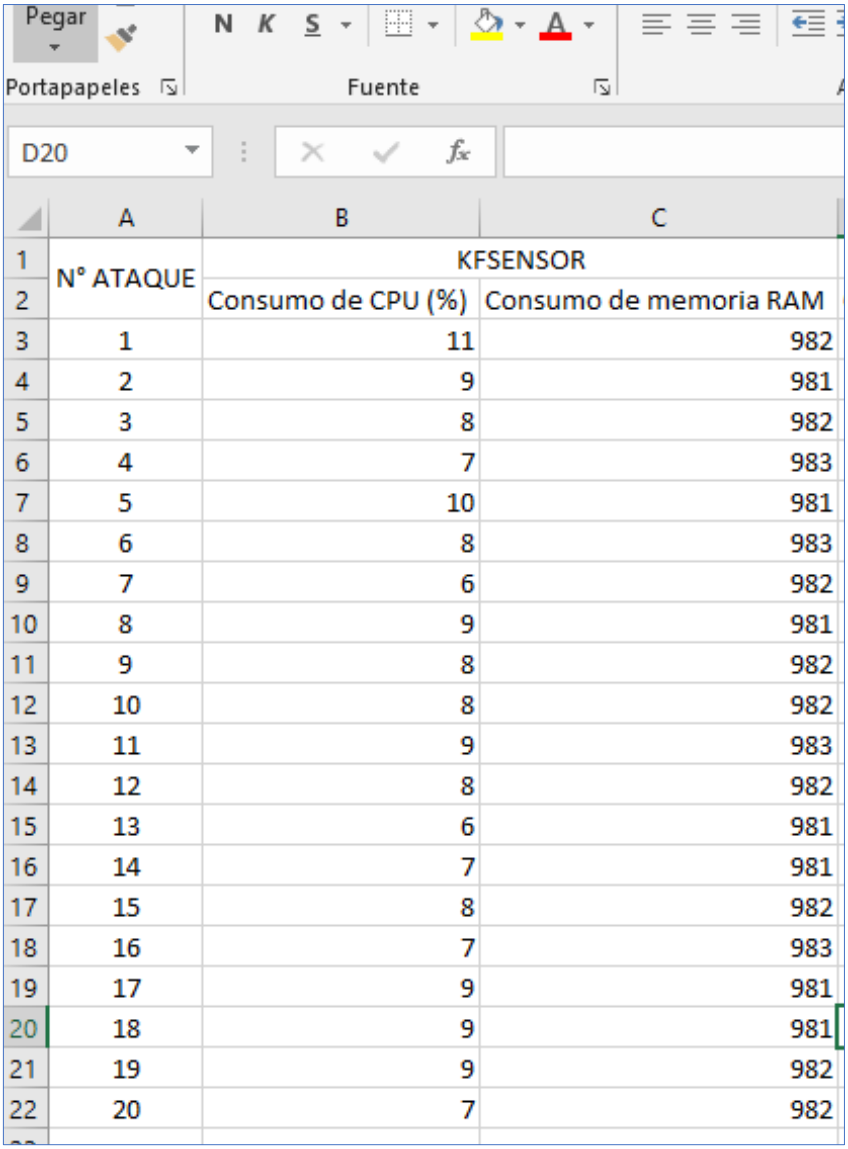


Figura 66. Ataque DDOS controlados desde consola

Figura 67. Captura de información en 20 instantes de tiempos



	A	B	C
1	N° ATAQUE	KFSENSOR	
2		Consumo de CPU (%)	Consumo de memoria RAM
3	1	11	982
4	2	9	981
5	3	8	982
6	4	7	983
7	5	10	981
8	6	8	983
9	7	6	982
10	8	9	981
11	9	8	982
12	10	8	982
13	11	9	983
14	12	8	982
15	13	6	981
16	14	7	981
17	15	8	982
18	16	7	983
19	17	9	981
20	18	9	981
21	19	9	982
22	20	7	982

Figura 68. Información recolectada durante las pruebas

ANEXO 9

Pruebas controladas de ataques DDOS, además recolección de consumo de CPU y consumo de memoria RAM en el honeypot HoneyD

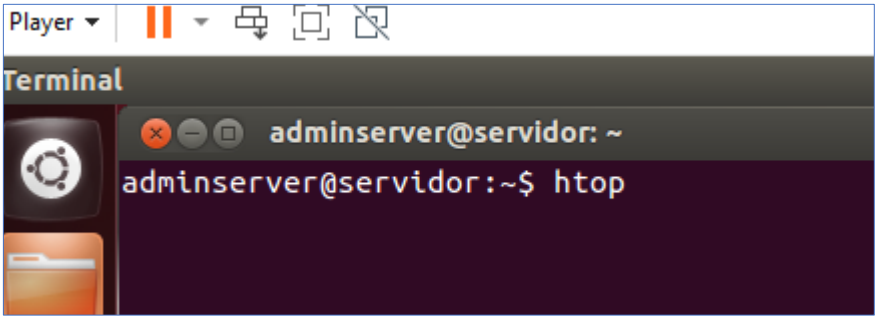


Figura 69. Comando para monitoreo de consumo de memoria RAM y consumo de CPU

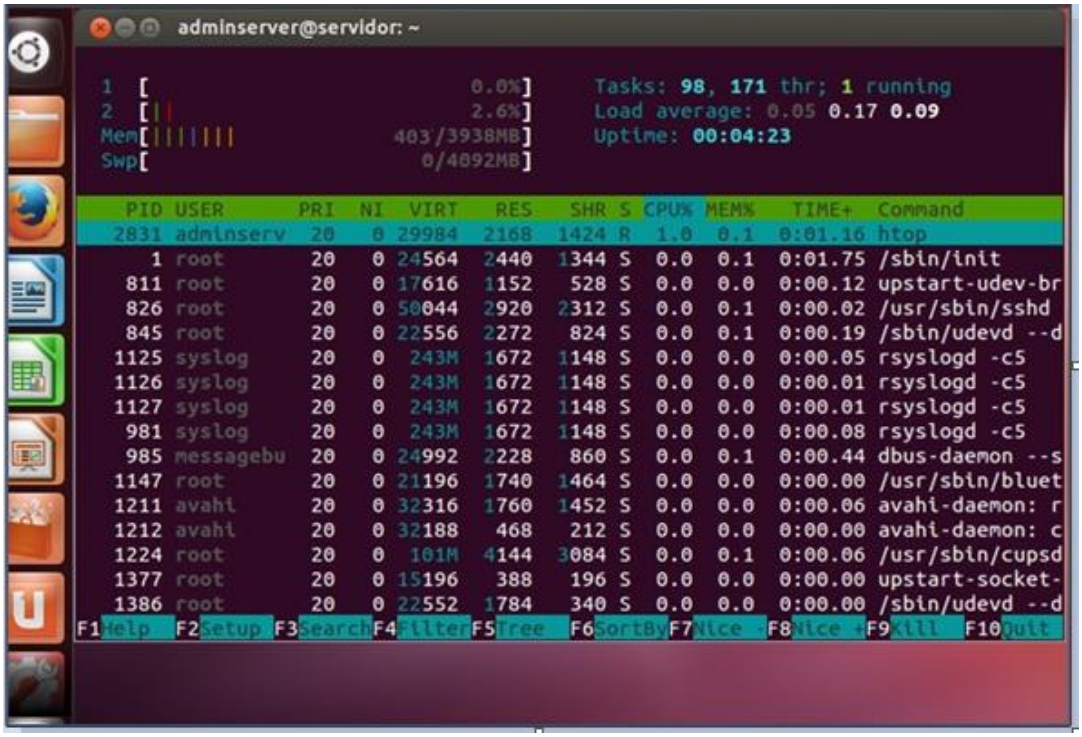


Figura 70. Aplicación HTOP

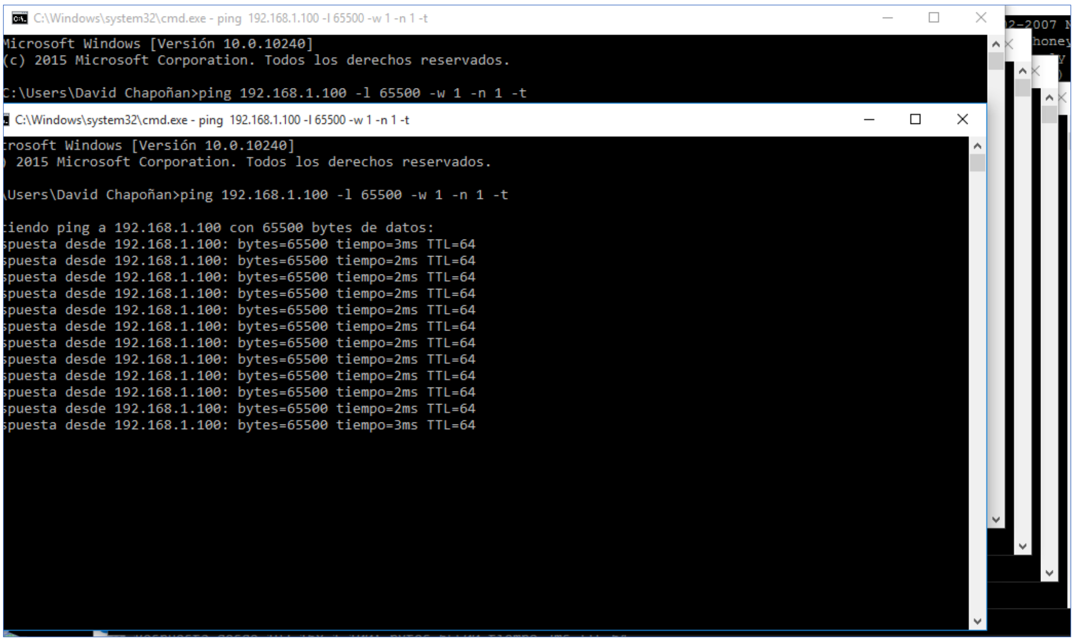
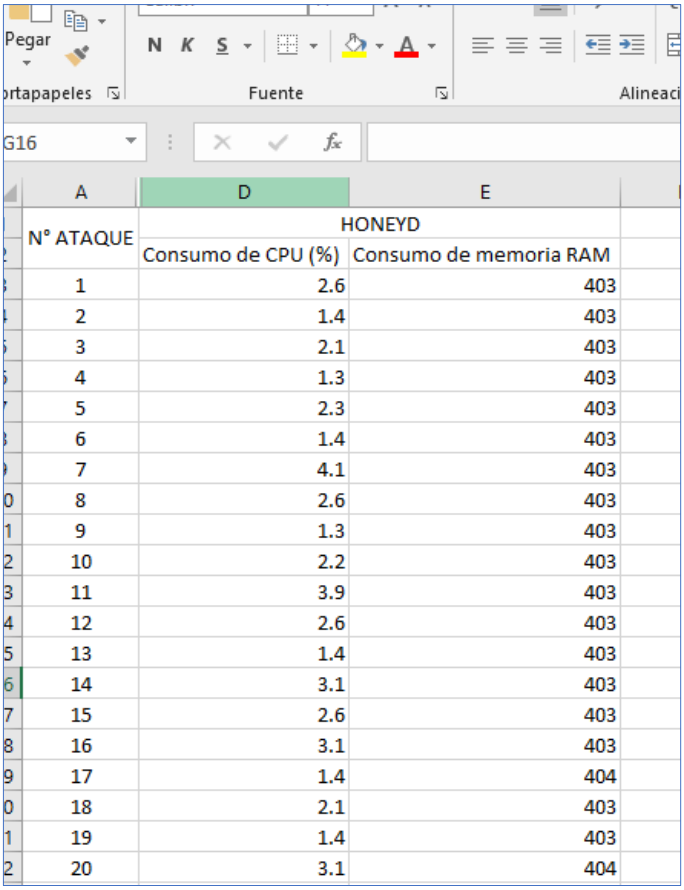


Figura 71. Ataques DDOS controlados desde consola



	A	D	E
	N° ATAQUE	HONEYD	
		Consumo de CPU (%)	Consumo de memoria RAM
3	1	2.6	403
4	2	1.4	403
5	3	2.1	403
6	4	1.3	403
7	5	2.3	403
8	6	1.4	403
9	7	4.1	403
10	8	2.6	403
11	9	1.3	403
12	10	2.2	403
13	11	3.9	403
14	12	2.6	403
15	13	1.4	403
16	14	3.1	403
17	15	2.6	403
18	16	3.1	403
19	17	1.4	404
20	18	2.1	403
21	19	1.4	403
22	20	3.1	404

Figura 72. Información recolectada durante las pruebas