UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA

CARRERA DE INGENIERÍA DE SISTEMAS

Trabajo de titulación previo
a la obtención del título
de Ingeniero de Sistemas

PROYECTO TÉCNICO:

"DESPLIEGUE DE HONEYPOT EN LA NUBE PRIVADA COMO PRIMER PERÍMETRO DE SEGURIDAD"

AUTOR:

HERNÁN JACINTO LEÓN LOJA

TUTOR:

ING. ERWIN JAIRO SACOTO CABRERA, PhD

CUENCA - ECUADOR

2021

CESIÓN DE DERECHOS DE AUTOR

Yo Hernán Jacinto León Loja con documento de identificación N° 0105007199, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación: **"DESPLIEGUE DE HONEYPOT EN LA NUBE PRIVADA COMO PRIMER PERÍMETRO DE SEGURIDA**D", mismo que ha sido desarrollado para optar por el título de: *Ingeniero de Sistema*s, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, noviembre de 2021.

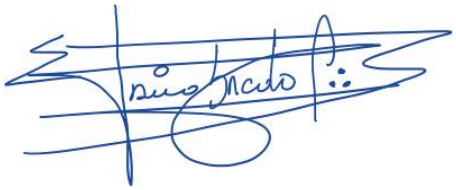
Hernán Jacinto León Loja

C.I 010507199

CERTIFICACIÓN

Yo, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **"DESPLIEGUE DE HONEYPOT EN LA NUBE PRIVADA COMO PRIMER PERÍMETRO DE SEGURIDAD",** realizado por Hernán Jacinto León Loja, obteniendo el *Proyecto Técnico*, que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, noviembre de 2021.



Ing. Erwin Jairo Sacoto Cabrera, PhD.

C.I. 03011852229

DECLARATORIA DE RESPONSABILIDAD

Yo, Hernán Jacinto León Loja con documento de identificación N° 0105007199, autor del trabajo de titulación: **"DESPLIEGUE DE HONEYPOT EN LA NUBE PRIVADA COMO PRIMER PERÍMETRO DE SEGURIDAD",** certifico que el total contenido del *Proyecto Técnico*, es de mi exclusiva responsabilidad y autoría.

Cuenca, noviembre de 2021.

Hernán Jacinto León Loja

C.I 010507199

DEDICATORIA

Dedico este logro mis padres Pascual y María, a mi familia, el principal cimiento de este desarrollo de mi vida profesional, que hicieron de mí una persona de bien. Ellos son el motor de apoyo y motivación que tuve durante mi vida académica y que sin duda en el futuro aún cuento con ellos.

También a mi tutor de tesis, Ing. Jairo Sacoto quien estuvo pendiente y ayudó a que este proyecto se realice y culmine satisfactoriamente. Así también dedico este trabajo a todas las personas que me han apoyado e hicieron posible uno de los anhelos mas deseados en mi vida.

Finalmente, a todos aquellos que puedan usar esta investigación para su beneficio profesional.

Hernán Jacinto León Loja

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios por darme sabiduría y fortaleza en el transcurso y finalización de mis estudios superiores obteniendo una experiencia satisfactoria en la Universidad Politécnica Salesiana.

También agradezco a quienes estuvieron conmigo durante mi formación académica, de manera especial a mis padres Pascual León y María Loja, a quienes amo y doy gracias por su esfuerzo, amor y sacrificio para poder darme una buena educación desde mi niñez, inculcando valores en mi, que me han servido para mi formación tanto personal como profesional.

De la misma manera agradezco a la Universidad Politécnica Salesiana, a los docentes de la carrera de Ingeniería de Sistemas, en especial al Ing. Pablo Gallegos y a mi Tutor Ing. Jairo Sacoto, por la ayuda brindada para la realización de este proyecto siendo guías y fuentes de conocimientos, que permitió el desarrollo y finalización de este proyecto.

Por último, quiero dar gracias a mis hermanos Klever, Henry, Nelson y John, por el apoyo incondicional que me han brindado, en los momentos mas difíciles que se me han presentado, a mis familiares, amigos, compañeros y demás personas que fueron parte de este proceso de mi vida.

Hernán Jacinto León Loja

INDICE

INDICE DE FIGURAS

INDICE DE TABLAS

CAPITULO 1

GLOSARIO DE TERMINOS

Docker: Software que permite empaquetar y ejecutar aplicaciones en un mismo entorno.

Dockerizar: Proceso de empaquetar una aplicación en un binario, de forma aislada respecto a otros programas.

FTP: Protocolo de transferencia de Archivos, usado para la transferencia de archivos basándose en un conjunto de reglas.

Honeypot: Herramienta informática que registra ataques para mejorar la seguridad en la red.

HTTP: Protocolo de Transferencia de Hipertexto, es el protocolo de comunicación usado para transferir información a través de archivos definidos.

HTTPS: Protocolo de Transferencia Segura de Hipertexto, es una evolución de HTTP, donde la transferencia de información se hace de manera segura.

Router: Dispositivo que permite la interconexión de varias redes.

SMTP: Protocolo para transferencia simple de correo, es un protocolo usado para el envío de correos electrónicos a través de internet.

SSH: Secure Shell es el protocolo que permite la administración remota y transferencia de información mediante un canal seguro.

Terminal: Es un programa que viene por defecto en los sistemas UNIX, permitiendo a los usuarios la ejecución de todo tipo de tareas en modo texto.

Virtualización: creación de un recurso tecnológico, mediante la utilización de software.

RESUMEN

En la actualidad, el aumento de ataques a los sistemas empresariales ha hecho que la seguridad sea un punto fundamental en el entorno de cualquier entidad moderna como es la gestión de las TI, sustentadas sobre una infraestructura de integración de redes que se encuentran en un alto índice de vulnerabilidad. Cada vez más empresas necesitan herramientas que ayuden a mejorar la seguridad de su infraestructura tecnológica para ello en el presente proyecto técnico se hará uso de honeypots para recolectar información que nos ayudara a saber el posible comportamiento de los atacantes durante y después de la penetración o los servicios de una empresa.

Para ello se analizó uno de dos posibles escenarios donde pueden ser colocados los honeypots, tomando en cuenta la cantidad de información a recolectar además de la precisión que brinde.

Después se implementó el escenario seleccionado de manera virtual, haciendo uso de software Open Source como VirtualBox y GNS3, además de seguir un plan de pruebas que permitió analizar qué tan eficaces puede llegar a ser el sistema multi-honeypot implementado.

Finamente el trabajo expuesto se complementa con el análisis de los resultados obtenidos a través de herramientas de recolección de datos que nos facilitan la interpretación de la información recolectada por los honeypots a fin de cumplir con los objetivos planteados.

Palabras clave: *honeypot, ataques, virtualización, simulación*

ABSTRACT

Currently, the increase of cyber-attacks to business systems has caused cyber security to be a fundamental parameter of consideration regarding the arrangement of any modern entity such as the management of IT, which are sustained upon an integrated network infrastructure considered to present a high vulnerability index. Progressively, more business systems need digital tools to improve the protection of their technological infrastructure; therefore, in the following technical project, the use of HONEYPOTS to collect useful information in order to know the possible behavior of hackers during and after the penetration to business systems services will be carried out.

To do so, one out of two possible scenarios in which honeypots can be used has been analyzed by considering the amount of information to be collected, as well as the accuracy the use of honeypots may pose.

Afterwards, the selected scenario has been implemented virtually by making use of the Open-Source software as VirtualBox and GNS3, besides, an analytic trial-plan to test the overall efficacy of an implemented multi-honeypot system has been followed.

Finally, the present project is complemented with the analysis of the results obtained using data-collection tools that ease the comprehension of the obtained data by honeypots to fulfill the proposed objectives.

Keywords: *honeypot, attacks, virtualization, simulation*

CAPITULO 2

INTRODUCCIÓN

El rápido crecimiento del consumo de los servicios web hace que la infraestructura sobre la cual estos funcionan se vuelva vulnerable ante cualquier ataque mientras más incrementa el número de usuarios consumiendo varios y diferentes servicios, los cuales son más propensos a sufrir ataques, por ello surge la necesidad de guardar, almacenar información de los usuarios, así mismos, a medida que se interconecten más equipos a la red, son vulnerables a distintos incidentes de seguridad que ponen en peligro la integridad de los sistemas de información. Las herramientas basadas en honeypots, permiten registrar información, teniendo altas tasas de falsos negativos, proporcionando datos a tiempo para asegurar los servicios vulnerables.

Esto se lo denomina una superficie de ataque que puede ocurrir cuando un usuario no autorizado puede manipular o extraer datos usando métodos de violación como contraseñas débiles y predeterminadas, contraseñas reutilizadas, ingeniería social, configuraciones incorrectas

Así también recopilar información forense para ayudar en el procedimiento de enjuiciamiento, ya que un honeypot puede tomar diversas formas y puede situarse en cualquier lugar en la red, de ahí su entorno dinámico también en un sistema de cortafuegos puede ser ubicado dentro, fuera o en la Zona Desmilitarizada (DMZ) y bien puede parecerse a un servidor web, un servidor de archivos, una tabla de base de datos, un host simple, pero el valor dependerá de la cantidad de información que puede reunir, tal como se indica por los autores en [1] y [2].

Desde luego, cualquier actividad que entre en el honeypot puede señalarse como una actividad maliciosa debido que no tiene ningún valor real en información. Además, cualquier tráfico que salga del honeypot es un indicador. Para mitigar estos efectos se puede hacer uso del engaño que proporcionara tiempo adicional para obtener información para preparar la defensa y prevenir el ataque.

CAPITULO 3

PROBLEMA

3.1 Definición

Hoy en día, dentro de las empresas se hace uso de varios servicios informáticos para que dentro de la misma se pueda desenvolver diariamente las actividades, por tanto, la seguridad que se tenga en la empresa es un factor primordial para que la información que reposan sobre grandes centros de datos permanezca en forma segura sin riesgo de pérdida o robo de esta. También las políticas internas que se tenga sobre los servicios informáticos ayuda a que dicha información se mantenga segura, pero existe causas por las que la información pueda verse comprometida, entre estas está el ataque a los servicios por parte de un ente externo a la empresa y que trata de infiltrarse en la red y provocar fallos en el sistema o extraer información de manera no autorizada, de tal forma que las empresas necesitan de un sistema que reporte de qué manera se comportan los ataques para sus servicios implementados, sin que éstos últimos se vean comprometido.

3.2 Justificación

Conforme a los problemas expuestos en el anterior punto, donde se describe los riesgos en la seguridad informática y el peligro latente que tienen los datos en una empresa. La propuesta de solución a los problemas expuestos, se encuentran en la tecnología de los honeypots, los cuales funcionan como un señuelo para un atacante, es decir, el ente que hace los ataques asume que está afectando a un equipo dentro del sistema con el objetivo de desestabilizar el funcionamiento de este, pero en realidad es un equipo diseñado específicamente para recibir estos tipos de ataques a los cuales, obtiene datos para un posterior análisis y corrección del error del sistema.

En el honeypot se puede determinar que alcance queremos que tenga, es decir que servicios se utilizaría e incluso que datos compartiría para lograr engañar al atacante y así determinar nuevas técnicas de desestabilizar un sistema existen o que aún no se han registrado

CAPITULO 4

OBJETIVOS

4.1 Objetivo General

Desplegar un Honeypot como servicio, como primer perímetro de seguridad usando herramientas OpenSource.

4.2 Objetivos específicos

- OE1. Investigar bibliografía de las herramientas que constituyen un honeypot y la integración con los perímetros de seguridad del centro de datos.
- OE2. Desarrollar un escenario empresarial y simulación de los vectores de ataque sobre una infraestructura del centro de datos.
- OE3. Desplegar el Honeypot sobre nuestro escenario y desarrollar el protocolo de pruebas.
- OE4. Analizar el resultado y aportes de nuestra investigación.

CAPITULO 5

MARCO TEÓRICO

5.1 Antecedentes y Estado del arte

De acuerdo con lo descrito en [3], la evolución de las Tecnologías de la Información y Comunicación (TICs), ha hecho que las personas tengan mayor accesibilidad a los sistemas de información, este tipo de tecnologías permite que cualquier individuo pueda tener acceso a la información, por ello la seguridad informática en la actualidad es indispensable para dar soporte a las nuevas tecnologías.

A medida que se logre crear un sistema lo suficientemente grande para que los piratas informáticos lo ataquen, estos intentan obtener el acceso utilizando fallas de seguridad en el sistema, al rastrear la violación en los servicios, este no sabrá si está en un sistema real o en honeypot.

Es importante comprender el funcionamiento interno de estas herramientas, por lo tanto, el siguiente capítulo proporciona información básica necesaria para comprender que distingue los servicios vulnerables de otras herramientas. Se introduce el concepto que es la base fundamental de la mayoría de las herramientas.

5.2 Honeypot

El autor de [4], define un honeypot como el recurso de seguridad informática que adquiere valor al momento de ser investigado, atacado o comprometido.

En el mismo sentido, en [5] se menciona que la definición de un honeypot implica el ser atacado para recopilar cualquier información, por lo tanto, tienen que actuar como una herramienta de engaño capaz de engañar a los atacantes pretendiendo ser muy importante dentro de la red que contiene información valiosa. Sin embargo, la tarea principal de un honeypot es monitorear y registrar toda la actividad del intruso que rápidamente se puede analizar para conocer sus motivos y curso de acción.

Un honeypot configurado correctamente y desplegado dentro de la red sirve como una herramienta de detección de alerta temprana y una herramienta de vigilancia de seguridad avanzada que puede facilitar información detallada sobre las posibles lagunas del sistema.

5.2.1 Objetivos de un honeypot

- Captura de ataques a vulnerabilidades conocidas

- Identificación de nuevas vulnerabilidades
- Descubrimiento de riesgo en los sistemas.

5.2.2 Ciclo de funcionamiento de un honeypot

Cuando un atacante entre a la red, lo primero que se encuentre es el sistema Honeypot que se conFigura y los ataques se centran en ellos. Esta herramienta es útil para detectar y registrar los datos de los ataques que reciben, pero nunca lo va a parar, en la *Figura 1* se presenta el ciclo del honeypot.

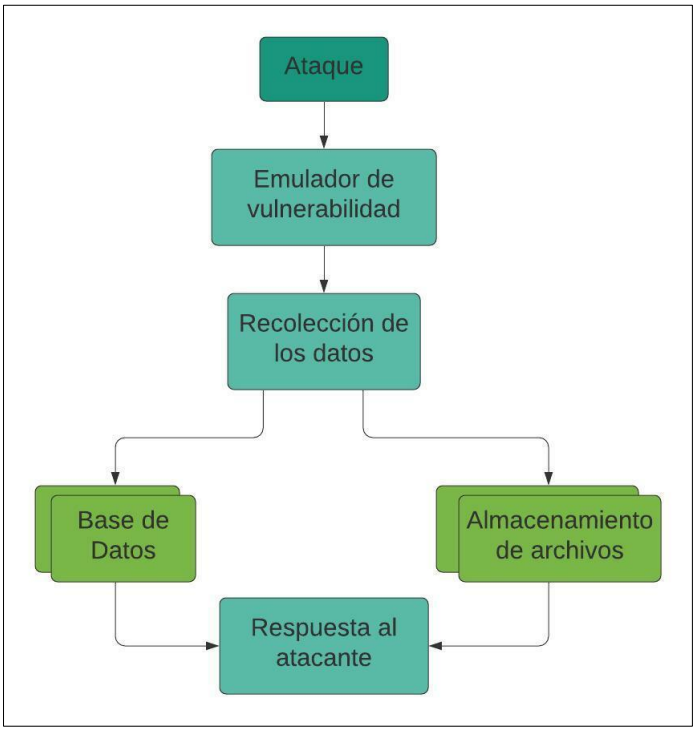


Figura 1. Funcionamiento de un Honeypot [Fuente: Elaborado por el autor]

5.2.3 Clasificación de los Honeypots

A continuación, se clasifica los honeypots en base a diferentes criterios, ya sea de uso o el nivel de interacción que brindan estas herramientas, como se visualiza en la *Figura 2*.

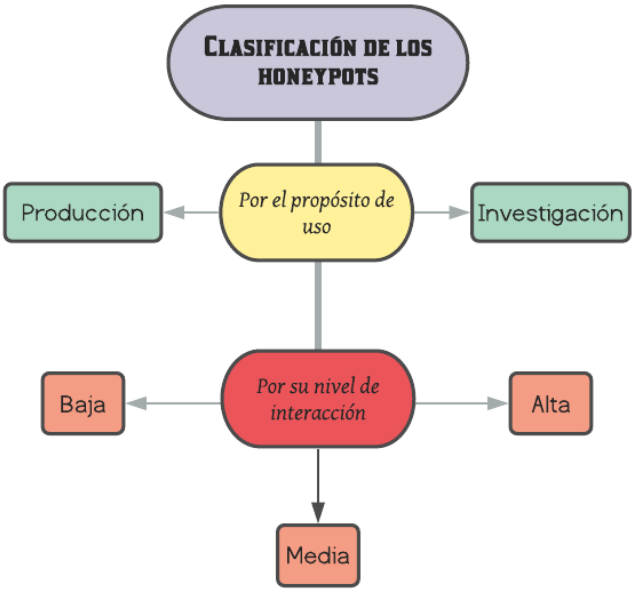


Figura 2. Clasificación de los honeypots [Fuente: Elaborado por el autor]

5.2.3.1 Clasificación por su propósito de uso

Esta clasificación está basada en el uso que tienen los honeypots ya que estos ayudan a mejorar la seguridad y reducir los riesgos en una organización. Para esta clasificación se tienen 2 categorías:

a) Honeypot de producción

El autor de [6], detalla que son sistemas que utilizan las organizaciones para para investigar porque motivo reciben ciberataques de los ciberdelincuentes, por esta razón la finalidad del honeypot de producción es investigar por qué están interesados en atacar en esa empresa, por ello simulan diferentes servicios con el único propósito de ser atacado, posterior a esto el atacante es descubierto y se toma medidas como denegar el acceso o limitar las capacidades de un servicio.

b) Honeypot de Investigación

Por otra parte, en [6] y [7] se explica que estos honeypots utilizados por organizaciones sin ánimos de lucro e instituciones educativas. El principal objetivo que se buscan con estos honeypots es investigar las maneras y motivos por las que los ciberdelincuentes buscan atacar los sistemas. Los investigadores hacen uso de esta herramienta para aprender nuevos métodos y obtener información sobre los atacantes. El honeypot de investigación toma un mecanismo de control de datos efectivo para evitar que sea un salto para atacar otro sistema informático.

5.2.3.2 Clasificación por su nivel de interacción

Debido a la arquitectura en la cual se encuentre basado el honeypot se los puede clasificar de por: Honeypots de baja interacción, honeypots de media interacción y honeypots de alta interacción.

a) Honeypots de baja interacción

Estudios realizados en [8], explican que el Honeypot de baja interacción tienen una interacción casi nula, en esta se presenta los servicios emulados por el atacante con una limitación que separan de un servidor y su funcionalidad se limita a imitar aplicaciones u otros sistema o equipos de la red.

b) Honeypots de media interacción

El autor indica en [9] este honeypot crece la relación entre el atacante y el sistema incluyendo recursos falsos, como ficheros o servidores del Protocolo de Transferencia de Archivos (FTP) o Secure Shell (SSH) ofreciendo más capacidad de interactuar que los honeypot de baja interacción, pero con menos funcionalidad que los honeypots de alta interacción. Este honeypot también tendrá la capacidad para aplicarse en el Protocolo de Transferencia de Hipertexto (HTTP) para simular la implantaciónn de algún proveedor conocido y proporcionando más información sobre el atacante que un honeypot de infección baja.

c. Honeypots de alta interacción

Así mismo en [10], se indica que el honeypot de alta interacción hace posible que el hacker pueda interactuar con el sistema. Normalmente son equipos con sistemas reales que tienen los mismos servicios que tendrían los servidores reales, por lo general existe poca restricción en el sistema sobre la actividad que realiza el hacker.

5.3 Seguridad de la información

Según estudios realizados por el autor en [11], indican que la seguridad de la información es la disciplina que se encarga de proporcionar la evaluación de riesgos y amenazas. En donde se debe realizar un plan de acción y adecuación, con el fin de reducir

los riesgos tomando en cuenta la normativa y/o las buenas prácticas para asegurar la confidencialidad, integridad y la disponibilidad de la información de los activos al momento de hacer uso de esta.

La seguridad de la información garantiza lo siguiente:

- confidencialidad,
- integridad y
- disponibilidad de la información.

La Seguridad de la información se apoya en la política de seguridad que se desarrolla mediante la elaboración y aprobación por el director de seguridad o Alta Dirección, este es de vital importancia puesto que se especifican los lineamientos de seguridad que deben ser cumplidos dentro de la entidad organizativa.

Es importante establecer objetivos de seguridad de la información. De esta manera se garantiza su confidencialidad, integridad y disponibilidad. Por otra parte, la Organización Internacional de Estandarización (ISO) ha desarrollado normas de gestión de la seguridad como la norma ISO 27001 que establece un modelo estándar para el Sistema de Gestión de la Seguridad de la Información (SGSI) como se representa en la *Figura 3*.

Figura 3. Normas ISO 27001 [12]

5.4 Centro de Datos

En [13], se indica que un centro de datos es aquel recurso informático que almacena, distribuye información donde las empresas mantienen y operan la infraestructura de Tecnología de Información (TI), por lo cual está en constante crecimiento, en el centro de datos está conformado por un conjunto de servidores instalados en la empresa que son propiedad de esta o son gestionados de manera privada,

el centro de datos se puede alojar servidores y sistemas de almacenamiento para la ejecución de aplicaciones y proporcionar seguridad y confiabilidad a sus datos.

5.4.1 Clasificación de los centros de datos

En [14] un centro de datos se clasifica por niveles tomando en cuenta su diseño general. Estos niveles están enumerados de I a IV en donde el último nivel, representa una mayor confiabilidad del centro de datos.

a) Nivel I (Centro de datos Básico): De acuerdo con [15], este nivel es el más básico, con componentes no redundantes. Además de incluir refrigeración y distribución de energía, este nivel puede o no incluir un UPS o un motor generador.

b) Nivel II (Centro de datos Redundante): Este nivel entró en vigor en octubre del 2010, con el objetivo de añadir componentes redundantes, además de una métrica de eficiencia energética para clasificar a los servidores de acuerdo con su consumo energético, como lo indica [16].

c) Nivel III (Centro de datos concurrentemente Mantenibles): En [17] se menciona que en este nivel se añaden formas de tolerancia a fallas, manteniendo las características de los niveles anteriores. Esto permite una configuración activa/pasiva.

d) Nivel IV: El autor de [18] establece que en este último nivel se tiene una ruta energética extra a comparación de la única ruta energética en el nivel anterior, además es capaz de tolerar cualquier falla de un equipo, sin que esto afecte la carga del sistema.

5.5 Perímetros de Seguridad

El autor de [19], indica que un perímetro es un punto o puntos donde se separan las redes locales seguras de las redes externas no seguras. Están marcados por barreras o líneas de cortafuegos que separan de manera lógica las distintas zonas de una infraestructura. Los perímetros de seguridad constituyen sistemas destinados a proteger de intrusos la red. La única diferencia es que, en lugar de un espacio físico, se protegen las redes privadas de un sistema informático. Estas medidas de seguridad dependen de cada organización, como se puede destinar una instalación determinada para vigilancia de seguridad con un método de verificación de identidad, debido a que los servidores están expuestos en peligro, habitualmente la seguridad perimetral deben ser circuitos cerrados que visualicen las áreas vulnerables, en este campo el perímetro de seguridad evita el acceso a usuarios no autorizados a la red, otro objetivo es muy importante es identificar a los atacantes y alertar a cerca de ellos así como filtrar o bloquear tráfico ilegítimo.

5.6 Vectores de Ataques

Los vectores de ataques se definen en [20], como métodos utilizados para atacar servicios aprovechando las fallas o agujeros presentes en la red, permitiendo el acceso de los hackers informáticos. Estos vectores explotan las debilidades propias de los usuarios ya que se aplica ingeniería social desde cualquier dispositivo que tenga acceso a las credenciales del usuario.

5.6.1 Clasificación de ataques

a) De intermediario

Estudios realizados por los autores en [21] , indican que un ataque Man in The Middle (MITM por sus siglas en Ingles), consiste en colocar el dispositivo atacante en medio de la comunicación (entre dos sistemas), esto permite recoger la información del transmisor para procesarla, interpretarla y posteriormente reenviarla al receptor sin que otros dispositivos lo detecten. El secuestro de sesión es una práctica muy común en entornos de red vulnerables, por ello la autentificación basada en certificados se puede utilizar contra estos ataques.

b) Fuerza Bruta

Así mismo en [21]**,** el ataque de Fuerza Bruta intenta descifrar la combinación de contraseña y nombre de usuario en el servidor web utilizando todas las posibles iteraciones. Aprovecha las contraseñas débiles que se utilizan en los sistemas de autenticación al iniciar sesión con las contraseñas. Para protegerse de este ataque, recomienda crear contraseñas largas y complejas, limitar el número de intentos de inicio de sesión fallido con la función de bloqueo de la cuenta.

c) Ataque de denegación de Servicio

En [22] se define al Ataque de Denegación de Servicio (DoS), como una acción que interrumpe totalmente al sistema o usuarios de los recursos requeridos para efectuar su normal funcionamiento, los ataques DoS aprovechan la fuerza bruta con el único objetivo de afectar el sistema mediante una sobrecarga de paquetes y datos en los servidores, estos ataques muchas veces se pueden solucionarse aplicando parches para limpiar o bloquear la carga excesiva del sistema, pero existe la desventaja que al momento que el atacante envíe paquetes enmascarados o por difusión estos son imposibles de detener. Como se indica en [23], en los ataques DoS el atacante sobrecarga el ancho de banda del tráfico de red haciendo que estos recursos no estén disponibles para otros usuarios, debido a otros nodos no pueden enviar datos después de detectar el medio ocupado.

d) Explotación de día cero

Los autores en [24]**,** definen como explotación de día cero a una vulnerabilidad de la red que es nueva y reciente, antes que los parches sean liberados por los fabricantes, por lo tanto, la prevención de los atacantes de día cero requieren supervisiones constantes, practicas ágiles, también se involucran los usuarios de software que no actualizan los parches de seguridad con regularidad. Por lo cual, al no hacerlo conduce a diversas consecuencias en forma de ciberataques. El día cero también depende el modo de detención, es decir los hackers de sombrero blanco identifican las vulnerabilidades,

estos permiten mantener un bajo perfil hasta que se lance el parche respectivo de seguridad.

e) Secuencias de comandos entre sitios (XSS)

Como se indica en [20] y [25], un ataque XSS son secuencias de comandos entre sitios, se introducen en una aplicación web fiable, pero esta app es vulnerable la cual, el usuario al ejecutar la aplicación recibe el contenido malicioso que aparenta ser como código legítimo de la aplicación, sin darse cuenta la victima termina ejecutando el script malicioso debido a que desconoce el contenido legítimo de la aplicación. Esto indica que el atacante no se dirige a la víctima sino utiliza las fallas en las aplicaciones web vulnerables para enviar código malicioso

5.7 Firewall e IPS

Según los autores en [26], mencionan que el firewall es un sistema que busca proteger la información de una red del resto de las redes. Las vulnerabilidades que son posibilidad de ocurrencia de una amenaza sobre un activo y la amenazas evento, exitoso o no, que atenta sobre el buen funcionamiento del sistema. Específicamente el firewall son equipos que hacen cumplir las políticas de seguridad lo que se quiere evitar es que los usuarios de internet no autorizados ingresen dentro de la red empresarial evita tener acceso a los recursos o servicios externos. EL firewall es más una puerta que controla la entrada y salida de paquetes es un guardián de quien puede pasar y no. Actividad que realiza controla y gestiona a través del bloqueo de puertos [27].

En [28] se expone que un sistema de prevención de intrusos (IPS) es una evolución a los tradicionales IDS, el cual, toma contramedidas o respuestas para detener ataques detectados, realiza tareas para proteger los activos de información. El IPS comparado con el Firewall usa reglas estéticas es decir requieren actualizaciones continúas requiriendo monitoreo constante.

5.8 DMZ

De acuerdo con los autores de [1], una DMZ, es una red local agregada entre una red protegida una red externa para proporcionar una capa de seguridad. El objetivo de una DMZ es que las conexiones desde la red interna a la DMZ, como primera línea de la red es proteger los recursos valiosos de la entidad y desde la red externa a la DMZ que en lo general las conexiones solamente se permitan a la red externa; de este modo, se impide generalmente que los equipos de la DMZ se conecten directamente con la red interna o bien se puede utilizar una DMZ para aislar maquinas especificas en la red de otras máquinas.

5.9 Sistema de Detección de Intrusiones (IDS)

En [29] se indica que un IDS es una dispositivo o software que monitorea constantemente uno o varios recursos en la red, tratando de encontrar actividad malintencionada o una violación a las políticas de la empresa. Un IDS es parecido a un firewall, pero difieren en la manera que detectan las intrusiones, ya que un firewall detecta la intrusión, pero no identifica este ataque, mientras que el IDS ayuda a los cortafuegos a la detección de ataques dado que existen anomalías que son imposibles de ser detectadas permitiendo mantener sistemas de defensa permanente advirtiendo cualquier intento o actividad sospechosa que ocurra en el sistema, además de tomar medidas de protección perimetral. Además, existen IDS que están estrechamente relacionados con el sistema de protección para cuando detectan una amenaza, bloquean el trafico que esta genere o a su vez bloquean el dispositivo desde donde provenga la amenaza. Aunque este tipo de seguridad es muy bueno no deja de tener errores ya que con esto se puede llegar a bloquear o denegar los servicios que opere en la infraestructura.

5.10 T-POT

Tal como se describe en [30], T-POT es una plataforma multi-honeypot en la cual se encuentran varios honeypots dockerizados en un mismo Sistema Operativo (SO), logrando tener servicios de red de detección de intrusos y un motor de seguimiento. Esta permite emular servicios de red como: Routers, Protocolo de Oficina de Correo (POP3), postgreSQL, Protocolo para transferencia simple de correo (SMTP), FTP, HTTP, Bloque de Mensajes de Servidor (SMB) entre otros.

5.10.1 Honeypots incluidos en TPOT

a) Cowrie

En [31] se explica que un Honeypot Cowrie simula ser un servidor SSH y Telnet con una interacción alta, además de registrar las formas en que actúa un atacante para intentar penetrar el sistema. Gracias a las prestaciones del honeypot Cowrie se permite a los atacantes acceder al honeypot usando un cliente Telnet o SSH emulado y capaz de registrar toda la información de la sesión que realiza el atacante en formato JSON siendo útil puesto que este formato puede ser utilizado por varias herramientas, tal como se expone en [32].

b) Mailoney

Las investigaciones realizadas en [33] describen a mailoney como un honeypot en donde se simulará el servicio de correo electrónico por el puerto 25, lo que nos permitirá registrar los diferentes emails enviados cuando el servicio se configure como open relay, además de registrar las credenciales usadas en los intentos de inicios de sesión.

c) Snare

El autor en [34] menciona que es un honeypot de aplicaciones web, siendo el sucesor de Glastopf, heredando muchas características similares a las existentes en

Glastopf, así como la capacidad de convertir páginas web existentes en superficies de ataque con TANNER.

d) Citrixhoneypot

De acuerdo con el autor en [35], este crea un sitio web falso sobre el Protocolo de transferencia Segura de Hipertexto (HTTPS) en donde los posibles atacantes trataran de ingresar usando una forma de autenticación para vulnerar la seguridad de la página.

e) Rdpy

Los autores en [36] indican que, este honeypot es desarrollado bajo Python, para simular un servicio de Protocolo de Escritorio Remoto (RDP) de Microsoft, funcionando como un MITM para registra la sesión. Admite la capa de seguridad RDP estándar, RDP sobre Capa de Sockets Seguros (SSL), entre otros.

f) Dionaea

Los autores en [37] mencionan que un honeypot Dionea de baja interacción escrito en C y Python está diseñado para la simulación de servicios que contengan vulnerabilidades. Estos servicios pueden ser SMB, HTTP, FTP, MYSQL entre otros. Dionaea recopila información sobre las vulnerabilidades utilizados por el programa maligno de esta manera se obtiene una copia secuestrada del programa maligno utilizado por el atacante.

g) Heralding

Estudios desarrollados por [38] y [35] indican que es un honeypot simple que únicamente recoge información de inicio de sesión o credenciales utilizados en los diferentes protocolos de red como: FTP, HTTP, POP3, TELNET, SSH, RDP, entre otros.

h) Adbhoney

De acuerdo con [39], es un honeypot de baja interacción que usa el protocolo Android Debug Bridge que simula teléfonos, TVs conectados al host.

i) Dicompot

En [2], se indica que Dicompot es un honeypot que simula un servidor de Imagen Digital y Comunicación en medicina (DICOM) completamente funcional con un toque, el cual es un estándar de transmisión de imágenes médicas y datos entre hardware de propósito médico.

j) Ciscoasa

En [40] se explica que este honeypot se desempeña a manera de baja interacción en el componente Cisco Adaptive Security Appliance (ASA) capaz de detectar CVE-2018-0101, una vulnerabilidad de ejecución remota de código y Sistema Operativo de Disco (DOS).

k) Conpot

Así mismo en [41], se describe a Conpot como un honeypot de baja interacción, el cual permite emular una infraestructura industrial compleja, siendo fácil su

implementación, modificación y extensión. El objetivo es recopilar información sobre los motivos y métodos de los adversarios que apuntan a la entidad u organización.

l) HoneySAP

En [42], se afirma que es un honeypot de baja interacción centrado específicamente en la investigación relacionados a servicios de Aplicaciones y Productos de Sistema (SAP). Principalmente busca descifrar las técnicas centrado en la investigación específico para servicios SAP.

5.11 Plataformas para la virtualización

Estudios realizados en [43], indican que las máquina virtuales asumen que los recursos que poseen son propios de ellos y ven a otras máquinas virtuales como sistemas independientes, por ello los servidores pasan a ser alojados en entes lógicos conocidos como (maquina virtuales) destinadas a comunicarse y compartir recursos físicos de los que disponen, independientemente como estén organizadas e integradas dentro de una infraestructura virtual estamos hablando de virtualización de la plataforma que será útil para el desarrollo de los honeypot.

5.11.1 Virtualbox

En [43] se indica Virtualbox que es una herramienta de gestión de máquinas virtuales encargada de virtualizar todos los componentes de un ordenador, permite instalar múltiples sistemas operativos, se puede ejecutar maquitas virtuales de 32 y 64 bits como: Linux, Microsoft Windows, Solaris con diferentes características, Virtualbox trabajo con interfaces gráficas y basados en línea de comandos de esta forma manipulan las máquinas virtuales, configurar diferentes formas de conexiones de red entre el dispositivo.

5.11.2 Máquinas virtuales

Las máquinas virtuales de acuerdo con [43], son las que corren paralelamente sobre una máquina física anfitrión o host de manera que tiene permisos y hacen uso de los recursos físicos del hardware que son abstraídos de él, cada máquina cree que usa sus recursos propios de hardware cuando en realidad lo hacen de modo virtual, las máquinas virtuales corren distintos sistemas operativos, diferentes servicios o aplicaciones.

5.11.3 Sistemas operativos

Los sistemas operativos en [44] son un conjunto de programas de un sistema informático que gestiona recursos del hardware y es el principal programa que se ejecuta en toda la computadora de propósito general, en él se ejecutan lenguajes de programación en los cuales están desarrollados y que se puede desarrollar para ellos, tiene un conjunto básico de funciones ofrece interfaces graficas a los usuarios.

5.12 Linux y Distribuciones

5.12.1 Kali Linux

Es una distribución Linux basada en Debían destinada a pruebas de penetración y auditoria de seguridad [45]. Kali es una herramienta esencial para tareas de seguridad de la información como: pruebas de penetración, ingeniería inversa e informática forense.

5.12.2 Ubuntu Server

En [46] se define a Ubuntu Server como un SO sin entorno grafico al cual se realiza peticiones, acciones mediante la consola, aunque también se le puede instalar un entorno gráfico. El manejo de Ubuntu Server es muy similar a otros sistemas basados en Linux, es una variante de Ubuntu, está dedicada para uso en servidores, sobre todo en el uso de servicios web, servidores de base de datos, Protocolo de Configuración Dinámica de Host (DHCP), Sistema de Nombres de Dominio (DNS), entre otras. La funcionalidad de este SO, ha hecho que la configuración de servicios y los procesos vinculados, sean realizados con mayor eficiencia y aprovechados por servidores.

5.13 Windows 10

En [47] se indica que Windows 10, pertenece a la familia NT5 siendo el sucesor del SO Windows 8 destinado para el uso ya sea en cliente y/o servidor. Windows 10 es una edición completa diseñada para toda la familia de los productos de Microsoft como: teléfonos inteligentes, laptops, tabletas, entre otros.

5.14 Router Virtual Mikrotik con RouterOS:

En [48] se define a Mikrotik RouterOS como un SO basado en Linux. La empresa que lo desarrolla es SIA Mikrotīkls. Es propio del mikrotik tener su propio SO de configuración fácil que al virtualizarlo se convierte en un router lo que permite funciones como firewall, VPN Server y Cliente, Gestor de ancho de banda, Calidad de Servicio (QoS), punto de acceso inalámbrico y otras características utilizado para el enrutamiento y la conexión de redes.

Configuraciones de RouterOS soportan métodos de configuraciones como:

- Consola serial con una terminal
- Accesos vía Telnet y SSH vía una red
- Interfaz Gráfica WinBox
- Acceso local vía teclado y monitor
- Una API de configuracion dedicada al desarrollo de aplicaciones

5.15 GNS3

Como se indica en [49], GNS3 es un simulador libre, con un entorno grafico amigable, fácil de manejar caracterizándose por la obtención de una IOS de CISCO, permite diseñar topologías de red complejas de alta calidad y realizar simulaciones sobre la misma.

5.15.1 QEMU

De acuerdo con lo expuesto en [50], Qemu es un virtualizador y emulador de máquinas genéricas de código abierto, permitiendo la emulación completa de equipos. Qemu se diferencia por emular procesadores, procesos que han sido compilados para un procesador diferente al que se tiene instalado en el propio sistema, emula solamente el kernel y ejecuta el código de área de usuario, por lo que no ejecuta el código generado por las máquinas virtuales de forma nativa, sino que lo interpreta tal como se explica en [43, pp. 64 - 65].

5.16 Herramientas de visualización

5.16.1 Logstash

En [51] se explica que es una herramienta de recopilación de datos en tiempo real que puede consumir mensajes de fuente diferentes como HTTP en varios marcos de registros.

5.16.2 Kibana

Esta herramienta es una plataforma de virtualización para la configuración de datos de Elastisearch, permitiendo el paso de registros persistentes, presentando ilustraciones gráficas, diagramas y tablas obteniendo una descripción general de los servidores, de acuerdo con lo expuesto en [52].

5.16.3 Elasticsearch

En [53] y [51] se explica que es un motor de analítica y análisis distribuido, abierto para todos los tipos de datos estructurados o no estructurados. Elasticsearch es el componente principal de Elastic Stack, un conjunto de herramientas gratuitas para el almacenamiento y análisis y la visualización de los datos, usada para indexar varios tipos de contenidos como: búsqueda de aplicaciones, búsqueda empresarial, Búsqueda de sitios web, analítica de seguridad.

5.17 Herramientas para pruebas

5.17.1 Network Mapper (NMAP)

De acuerdo con [54], NMAP es una herramienta de escaneo de seguridad que detecta hosts, servicios y crea mapas de redes, siendo utilizado frecuentemente para tareas como el inventario de la red, gestión de programas y la supervisión del tiempo de actividad del servicio. NMAP es compatible con todos los sistemas operativos, por lo tanto, es una herramienta flexible que admite técnicas avanzadas para enrutamiento de redes, escaneo de puertos, IP, firewalls, enrutadores entre otros.

5.17.2 Hydra

En [55] se define a hydra como una herramienta para descubrir contraseñas, mediante la utilización de un script o automatización del proceso o través de líneas de código, es un complemento presente en Kali Linux.

5.17.3 Patator

En [29] se explica que es una herramienta multi-hilo bastante completa y flexible para ejecutar ataques de fuerza bruta escrita en Python, Patator guarda cada respuesta en un archivo de registro para una posterior revisión, para el ataque se usa FTP, SSH en la maquina Kali Linux atacante

Las tecnologías antes detalladas han soportado el desarrollo de redes de Quinta Generación (5G) y el desarrollo de diferentes modelos de negocio como los planteados en [62][63][64][65][66], así como soportar servicios de Internet de las Cosas (IoT) como los descritos por los autores en [67][68][69][70].

CAPITULO 6

METODOLOGÍA DEL TRABAJO

6.1 Topología Propuesta

En cumplimiento con el segundo objetivo "*Desarrollar un escenario empresarial y simulación de los vectores de ataque sobre una infraestructura del centro de datos*" de esta tesis. En este apartado se desarrolló dos propuestas de topologías o escenarios:

a) Escenario 1 – Pruebas

En este escenario denominado como 'Escenario de Pruebas', se dispuso de un servidor TPOT en una subred local de la empresa como se observa en la *Figura 4*, junto a los servidores que desplegaron los diferentes servicios de la empresa como se describe a continuación.

- Router Mikrotik: Router virtual mikrotik que se usó como un firewall para acceder a los servicios.
- Servidor FTP: Servidor encargado de la transferencia de archivos usando el protocolo FTP.
- Servidor de Correo: Servidor encargado de la gestión de los correos electrónicos dentro de la empresa.
- Servidor SSH: Servidor usado para administrar remotamente un equipo de manera segura.
- TPOT-Server: Servidor donde se alojó los múltiples honeypot que simularon los diferentes servicios en donde los atacantes realizaron sus ataques.
- ATK1: Representó la/las maquinas atacantes que trataron de vulnerar la seguridad de los servidores.
- Clientes: PC1, PC2, PC3: Usuarios que hicieron uso de los diferentes servicios de la empresa.

En este caso el servidor de honeypots al no estar expuesto de manera directa al internet, nos ayudó a saber cuándo un atacante se saltó las diferentes defensas de la infraestructura, pero no recolectó muchos datos recolectados debido a la naturaleza de su ubicación.

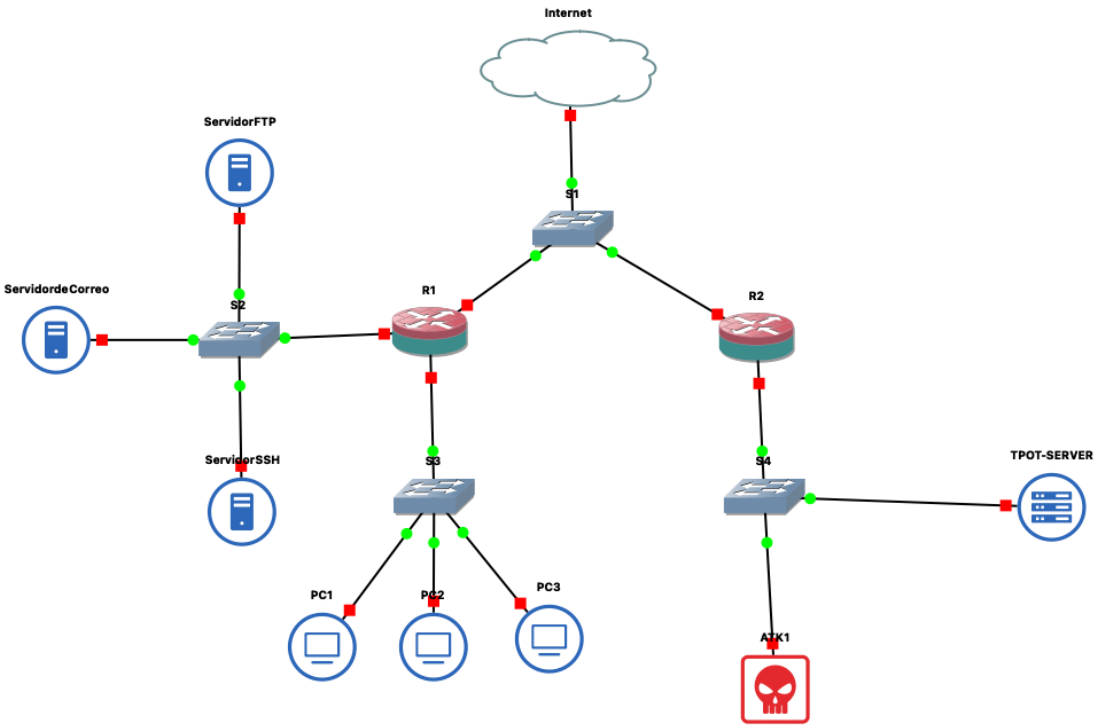


Figura 4. Escenario 1 - Pruebas

b) Escenario 2 – Escenario Empresarial

En este escenario también denominado “Escenario Empresarial” se ubicó el servidor TPOT en la DMZ de la empresa, como se observa en la *Figura 5*. En este diseño de laboratorio se simuló la exposición directa del servidor a internet y donde fue propenso a un mayor número de ataques con respecto al Escenario de Pruebas mostrado en la *Figura 4*. El objetivo de este escenario fue, representar la implementación de un Honeypot para una empresa, conformado por servicios Web, Correo, SSH y FTP colocados en la DMZ empresarial.

Descripción de componentes utilizados en el **Escenario 2 - Empresarial**:

- DMZ: Zona donde funcionaban los servicios de la empresa expuestas a internet.
- A1, A2: Son posibles computadoras atacantes los cuales realizaron una serie de ataques a los servicios implementados en la colmena de honeypot.
- Servidor de Correo: encargado de la administración del correo de la empresa.
- Servidor Web: Servidor en donde se colocaron los recursos web accesibles desde internet.
- Servidor SSH: Servidor que se encargó de administrar remotamente las actividades y procesos de la maquina y/o máquinas.
- Servidor FTP: Servidor encargado de la transferencia de archivos para lo cual hizo uso de FTP.

- Clientes: Cliente1, Cliente2: Son usuarios que hicieron uso de los servicios implementados por la empresa para realizar tareas necesarias dentro de ella.

En este caso el servidor de honeypot, al estar expuesto de manera directa a internet, recolectó un mayor volumen de datos con respecto al escenario anterior, lo que permitió de manera temprana detectar las posibles violaciones de seguridad a los servidores expuestos a la vez que se protegió a la red interna con ataques que se puedan dar desde dentro del honeypot.

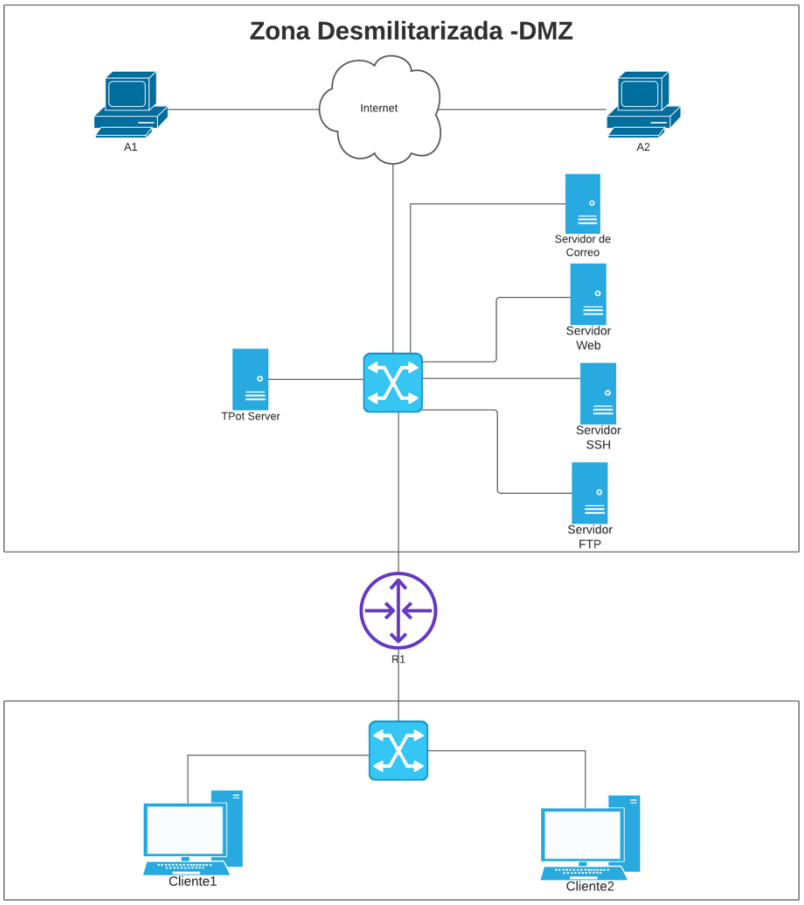


Figura 5. Escenario 2 - Empresarial

6.2 Fase Diseño

Luego de analizar los escenarios en donde puede ubicarse el servidor TPOT, se puede decir que, para mayor recolección de datos, respecto a los ataques que se recibe, lo óptimo fue colocarlo en la DMZ como se indica en la *Figura 5*, para saber el comportamiento de los atacantes en su intento por saltar las defensas de la infraestructura de la empresa.

Para la realización del Escenario elegido fueron necesarios los siguientes componentes:

- 1 maquina física
- Programa de virtualización
- Programa simulador de red
- Plataforma Multi-Honeypot
- Diccionarios con nombres de usuarios y contraseñas
- Plan de pruebas

A continuación, se detalla algunos de los componentes usados para el Escenario seleccionado. Los demás se desglosa más adelante.

6.2.1 Maquina Física

Es una computadora de escritorio (PC)

- Sistema operativo: Windows 10 Education
- Versión: 10-2004
- Procesador: Intel Core i7-3770K
- Gráfica: Intel(R) HD Graphics 4000
- RAM: 32GB DDR3
- Discos: 490GB SSD, 500GB HDD
- Configuración regional: español - Ecuador

Figura 6. Características Físicas de hardware

La instalación de Windows 10 se detalló en el **ANEXO 1**.

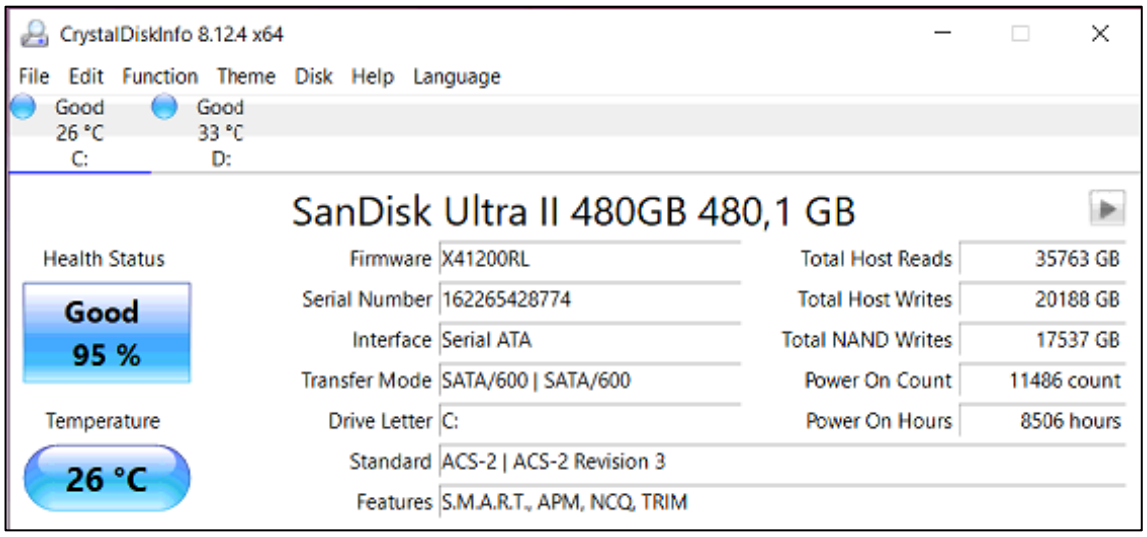


Figura 7. Características del disco SSD

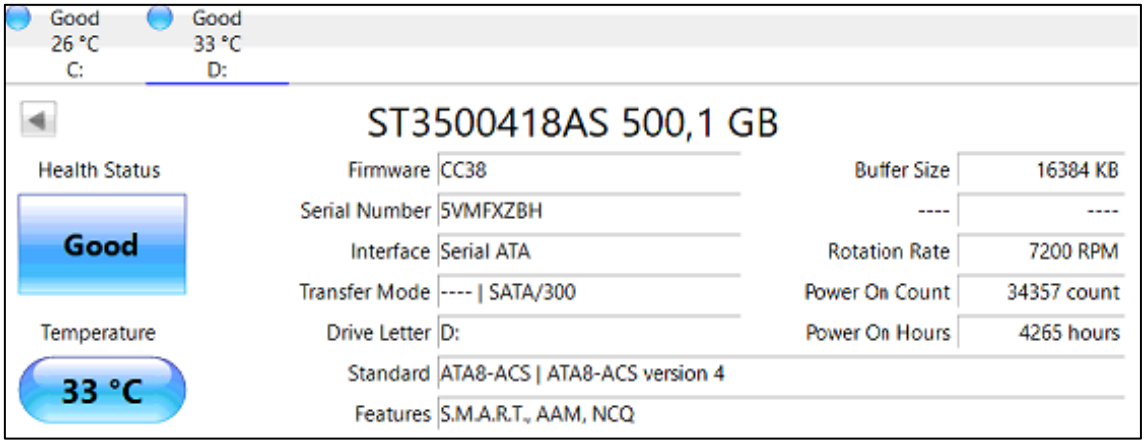


Figura 8. Características del Disco HDD

6.2.2 Programa virtualizador

Como programa virtualizador se optó por Virtualbox debido a que es Open Source bajo la licencia GNU/GPL, además de la versatilidad que ofrece para arrancar varias máquinas e incluso las herramientas administrativas para cada una de ellas. La instalación de esta herramienta se encuentra en el **ANEXO 2**.

6.2.3 Programa simulador de red

El programa simulador de red escogido es GNS3, en donde se puede realizar topologías de red completas pudiendo así combinar dispositivos reales y virtuales. La instalación de este programa simulador de red se encuentra detallada en el **ANEXO 3**.

6.2.3 Plan de pruebas

También denominado Protocolo de Pruebas fue necesario para verificar el complimiento de los objetivos 2 "Desarrollar *un escenario empresarial y simulación de los vectores de ataque sobre una infraestructura del centro de datos*" y 3 "Desplegar *sobre nuestro escenario y desarrollar el protocolo de pruebas"*. Este protocolo consistió en la realización de un plan donde se planteó una serie de ataques al servidor multi-honeypot, donde se tomó en cuenta los protocolos y puertos a los cuales atacar así también saber los honeypot encargados de registrar los ataques para su posterior visualización por herramientas especializadas en interpretar dichos datos, de una manera entendible para el usuario. En la *Tabla 1,* se anotó los protocolos definidos, así como sus puertos, los cuales son usados por defecto para dichos protocolos.

Protocolo	**Puerto**	**HONEYPOT**	**Herramienta**
SSH	22 TCP	Cowrie	Hydra
FTP	21 TCP	Dionaea	Hydra
SMTP	25 TCP	Mailoney	Patator
HTTP	80 TCP	Tanner	Patator

Tabla 1. Protocolos que serán puestos a prueba en el ataque

Luego de definir los protocolos y puertos para los ataques en la *Tabla 1,* se realizó la *Tabla 2*, donde se colocó los diferentes casos para realizar los ataques. Por ejemplo, en el primer caso se tuvo 1 maquina con un diccionario de 100 nombres de usuarios y otro diccionario con 1000 contraseñas, los cuales al momento que la herramienta fue realizando los diferentes ataques por fuerza bruta, combinó 1 nombre de usuario con las 1000 contraseñas del diccionario, esto quiere decir que la maquina realizó 100000 ataques a cada uno de los protocolos con las herramientas definidas en la *Tabla 1*. Para el número total de ataques que recibió el protocolo se multiplicó el número total de ataques por maquina por el número de máquinas que realizaron el ataque. En este caso como es una sola maquina fue el mismo número de ataques. Para los demás casos cambió el número ya que en el caso 2 se usaron 3 máquinas y en el caso 3 se usaron 5 máquinas, cada una con su propio diccionario para cada uno de los protocolos.

Caso	Numero de Maquinas	Numero usuarios por maquina	Numero de contrasenas por maquina	Numero de ataques por maquina atacante	Numero total de ataques	Protocolos
1	1	100	1000	100.000,00	100.000,00	SSH,FTP,SMTP,HTTP
2	3	300	3000	900.000,00	2.700.000,00	SSH,FTP,SMTP,HTTP
3	5	500	5000	2.500.000,00	12.500.000,00	SSH,FTP,SMTP,HTTP

Tabla 2. Plan de pruebas

Cabe recalcar que para el caso 2 y 3 mostrados en la *Tabla 2*, los ataques al servidor se realizaron de manera simultánea, lo que conllevó a la demanda de buenos recursos y buen rendimiento del servidor, ya que mientras crecía el número de ataques, el servidor usó más recursos para el registro de los ataques por lo que se vio afectado su rendimiento.

6.3 Fase Implementación

En esta fase, con el fin de dar cumplimiento al tercer objetivo *"Desplegar sobre nuestro escenario y desarrollar el protocolo de pruebas"* de la presente tesis, se explicó los procedimientos necesarios para la implementación del honeypot, así como una descripción de las configuraciones realizadas, además de sus recursos necesarios.

Para realizar la implementación, se seleccionó varios Honeypot que vienen integrados en TPOT, los cuales son: Cowrie, Dionaea, Tanner (Snare) y Mailoney. Además, se instaló máquinas virtuales con el SO Kali Linux, el cual incluye una serie de herramientas para la auditoria y seguridad informática, lo que nos permitió realizar una serie de ataques a cada uno de los servicios ofrecidos por los honeypot seleccionados para analizar las vulnerabilidades que explotaron estas herramientas en la infraestructura.

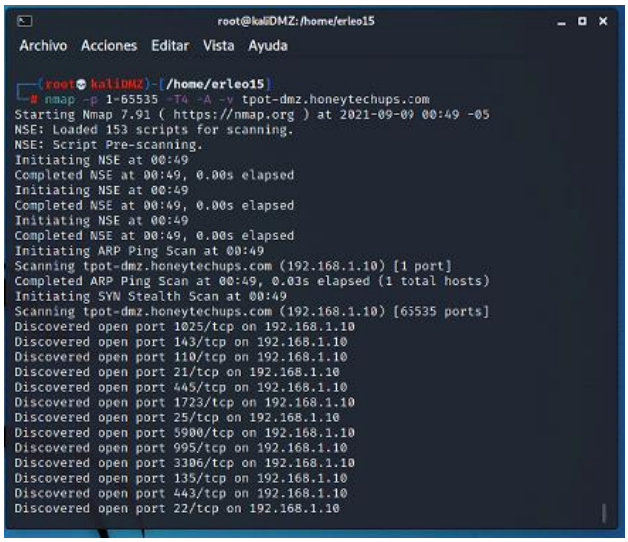


Figura 9. Escaneo de puertos con NMAP

Así también se hizo una serie de pruebas con Nmap (como se muestra en la *Figura 9*) la cual ayudó a escanear los puertos habilitados en un host. Además, se usó la herramienta TeraTerm (cliente de conexión SSH, Telnet, entre otros) la cual facilitó realizar tentativas de conexión hasta el host virtual y en su posterioridad se verificó que dicho dispositivo virtual era percibido como sistema real.

6.3.1 Implementación del Servidor Multi-Honeypot (TPOT)

A continuación, se ilustra la instalación del sistema multi-honeypot y la correspondiente configuración.

a. Se procedió a descargar un archivo de disco, desde su repositorio oficial en GitHub (https://github.com/telekom-security/tpotce/releases). En este caso se descargó la versión 20.06.2.
b. A continuación, en VirtualBox se creó una máquina virtual como se muestra en el **ANEXO 5**, con las características mínimas requeridas por este sistema multi-honeypot, las cuales son: 6-8GB RAM y 100GB Almacenamiento como se muestra en la *Figura 10*.

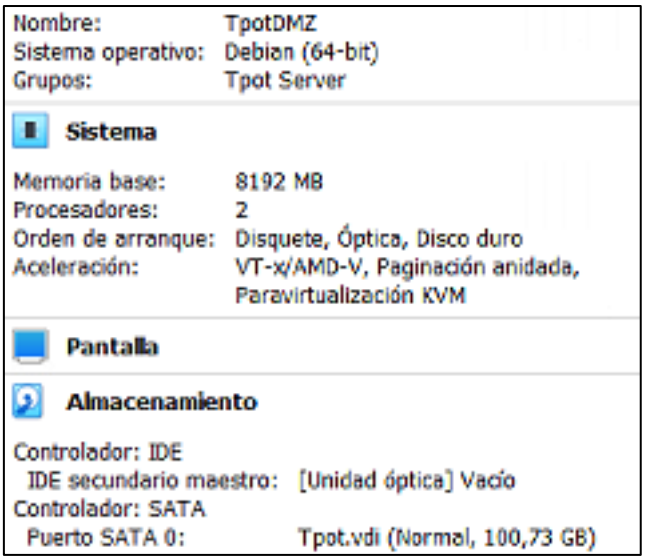


Figura 10. Características de la Máquina Virtual

c. Como siguiente paso, se cargó el archivo de disco en la máquina virtual y se realizó la instalación, de manera "Standard" con la respectiva información requerida en los pasos de instalación como se muestra en los **ANEXOS 6 y 7**.
d. Para la verificación de la realización del paso anterior, se llegó a obtener una pantalla similar a la *Figura 11* al momento de finalizar la instalación.

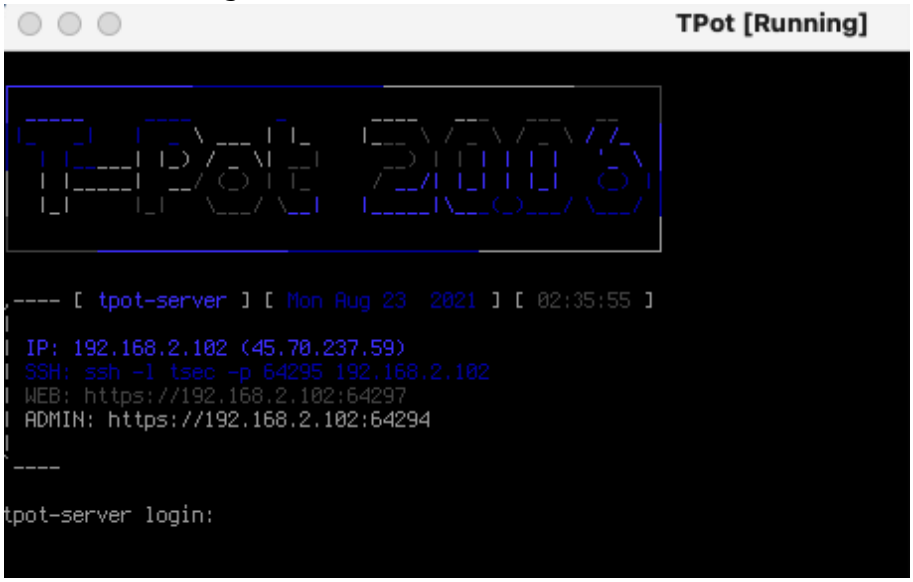


Figura 11. Pantalla de inicio de TPOT

Cabe recalcar que la información presentada en la *Figura 11*, es muy importante debido a que gracias a esos datos se logró administrar el servidor remotamente, ya sea por SSH o con una interfaz WEB. Además, para configurar los diferentes honeypot que contiene este servidor, se modificó el archivo *tpot.yml* que se encuentra en el directorio /opt/tpot/etc/ y luego reinició el servicio tpot con el comando *"sudo service tpot restart"*. Todo este proceso de instalación se encuentra detallado en el **ANEXO 7**, así como la configuración en el **ANEXO 8 sección 1**. También en el **ANEXO 8 sección 5,** se detalla una manera de administrar el servidor de manera remota y por último en el **ANEXO 8**

sección 2 se muestran maneras de revisar el estado de los honeypot que se encuentran activos en el servidor.

6.3.2 Implementación de servidores con sistemas reales de la empresa

Para realizar la Implementación de servidores con Sistemas reales de la empresa, se descargó un archivo de disco de Ubuntu Server 16.04 LTS desde su web oficial (https://ubuntu.com/download/server).

a. Luego se creó las máquinas virtuales en Virtualbox con características mínimas para su funcionamiento. Por ejemplo: 2GB RAM y 100GB Almacenamiento.

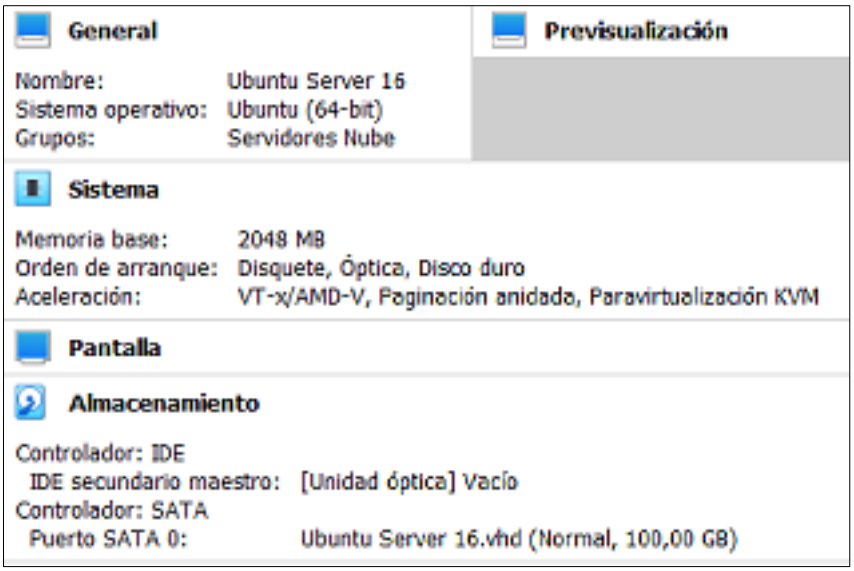


Figura 12. Características Máquina Virtual como servidor

Como siguiente paso, se procedió a cargar la imagen ISO en la máquina virtual, con la respectiva información requerida en los pasos de instalación. Además, se instaló los diferentes paquetes para los servicios que se hicieron uso en cada uno de ellos como: Openssh-Server, Postfix, Apache y Vsftp de los cuales su instalación se detalló en el **ANEXO 13**.

Para un mayor detalle del proceso de instalación se debe revisar el **ANEXO 10**, donde se expone paso a paso el procedimiento de instalación.

6.3.3 Implementación de Máquinas Atacantes

Para este apartado se procedió a instalar 5 máquinas virtuales con el SO Kali Linux, como se muestra en el ANEXO. Las características de las máquinas pueden ser mínimas (1GB RAM y 100GB Almacenamiento) debido a que no es necesario de muchos recursos en las maquinas atacantes para la ejecución de las pruebas.

6.3.4 Preparación de diccionarios para realizar el plan de pruebas

El uso de diccionarios de usuarios y contraseñas fue indispensable para el plan de pruebas puesto que se consistió en realizar ataques de fuerza bruta a cada uno de los puertos comunes para los protocolos HTTP, FTP, SSH y SMTP. Para proceder a armar los diccionarios se siguió los siguientes pasos:

a. Descargar archivos diccionario de páginas especializadas en recolectar información de usuarios y contraseñas como skullsecurity.org o packetstormsecurity.com.
b. Crear una estructura de directorios acorde a los casos de pruebas a realizar.

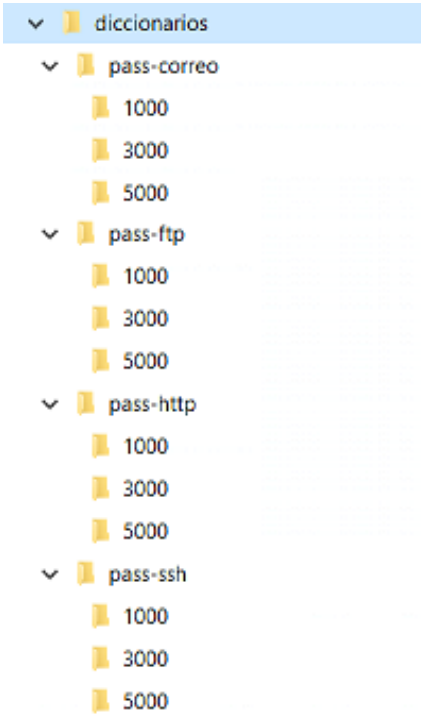


Figura 13. Archivos de Diccionario de Usuarios y Contraseñas

c. Cada directorio final (1000, 3000, 5000) tuvo sus correspondientes archivos de texto con usuarios, contraseñas y además la máquina en donde se usó, como se puede apreciar en la *Figura 14* y el contenido en la *Figura 15.*

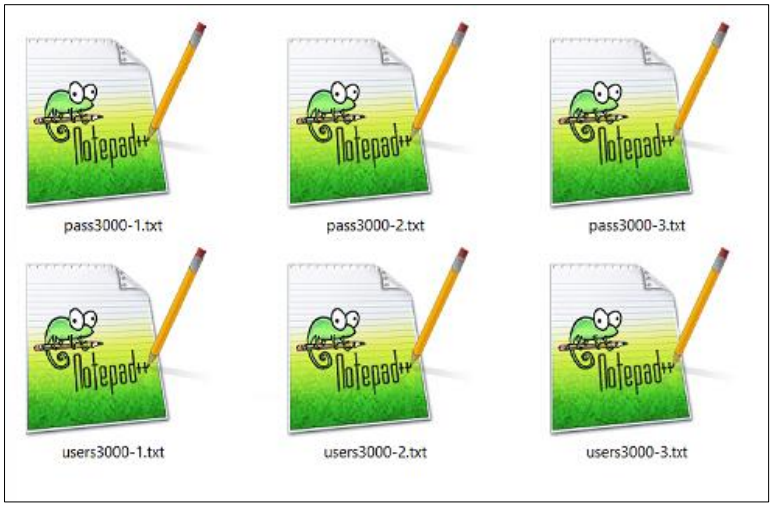


Figura 14. Diccionario de datos Contraseñas y Usuarios

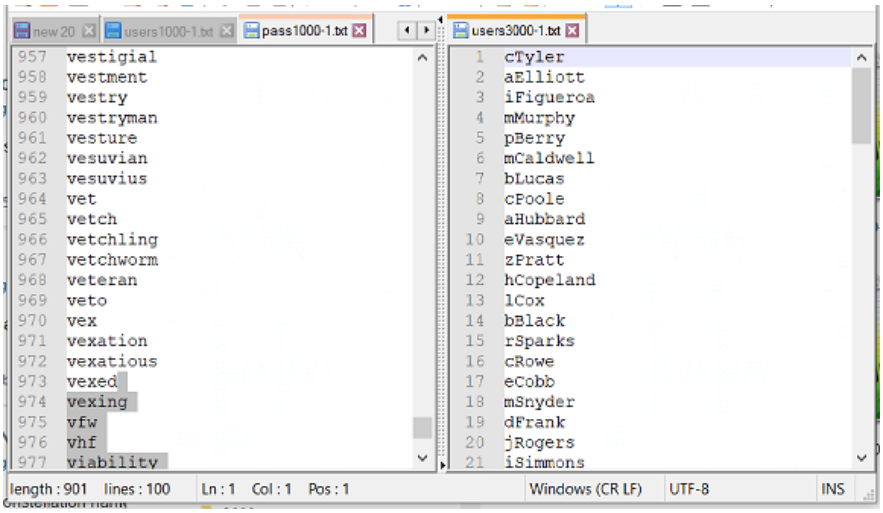


Figura 15. Contenido de los diccionarios

6.3.5 Líneas de comandos para pruebas

Para la ejecución del plan de pruebas fue indispensable realizar los comandos para cada caso de prueba, es decir escribir palabras reservadas que el SO hace uso para ejecutar una orden determinada desde una terminal, estas palabras reservadas se escribieron teniendo en cuenta cada uno de los parámetros que las herramientas requerían como se explica a continuación:

a. **Para los protocolos FTP y SSH:** En este caso se usó la herramienta Hydra, de la siguiente manera:

   hydra -L path_users.txt -P path_passwords.txt host protocolo -t 64

   -L *path_users.txt:* Ruta donde se encuentra el diccionario de los nombres de usuario a usar.
   -P *path_passwords.txt*: Ruta donde se encuentra el diccionario de las contraseñas a usar.

host: dirección ip o nombre de host a donde se va a realizar el ataque protocolo: El nombre del protocolo por donde se va a realizar el ataque. -t 64: Es el número de tareas paralelas. Por defecto es 16.

Ejemplo:
- hydra -L /ftpdir/pass-ssh/3000/users3000-1.txt -P /ftpdir/pass-ssh/3000/pass3000-1.txt tpot-dmz ssh -I -t 64

- hydra -L /ftpdir/pass-ssh/3000/users3000-1.txt -P /ftpdir/pass-ssh/3000/pass3000-1.txt tpot-dmz ftp -I -t 64

b. Para los protocolos SMTP y HTTP: Para los protocolos mencionados se hizo uso del módulo patator, incluido en Kali-Linux, así como sus herramientas smtp_login y http_fuzz de la siguiente manera:

smtp_login:
patator smtp_login user=FILE0 password=FILE1 0=/path_users.txt 1=/path_passwords.txt host=host

user=FILE0: se debe colocar un usuario o lista de usuarios.

password=FILE1: se debe colocar una contraseña o lista de contraseñas.

0,1: Hace referencia a la ruta donde se encuentran las listas de usuarios o contraseñas correspondientes.

host=host: Se debe colocar la dirección IP o hostname del objetivo.

Ejemplo:

patator smtp_login user=FILE0 password=FILE1 0=/ftpdir/pass-http/1000/users1000-1.txt 1=/ftpdir/pass-http/1000/pass1000-1.txt host=tpot-dmz

http_fuzz:

patator http_fuzz url=url user_pass=FILE0:FILE1 0=/path_users.txt 1=/path_passwords.txt -x ignore:code=401

url=url: se indica la página web a donde se requiere realizar el ataque.

user_pass=FILE0:FILE1: Se indica el/los usuarios(s)/contraseña(s).

0,1: se indica la ruta de la ubicación de las listas antes mencionadas.

-x ignore:code=number: se indica que error devuelto por el servidor, se ignorará, para que el proceso continúe normalmente.

Ejemplo

patator http_fuzz url=http://tpot-server.honeytechups.com/user/login user_pass=FILE0:FILE1 0=/ftpdir/pass-http/1000/users1000-1.txt 1=/ftpdir/pass-http/1000/pass1000-1.txt -x ignore:code=401

6.3.6 Implementación del Escenario Real

En este aparatado se expone el proceso que se realizó para armar el escenario virtualmente en el software GNS3 usando imágenes QEMU y máquinas virtuales de VirtualBox.

a. El primero paso fue activar la máquina virtual de GNS3 (GNS3 VM) en los ajustes de GNS3 como se muestra en la *Figura 16* o se puede revisar el **ANEXO 4** y **ANEXO 9** para un mayor detalle.

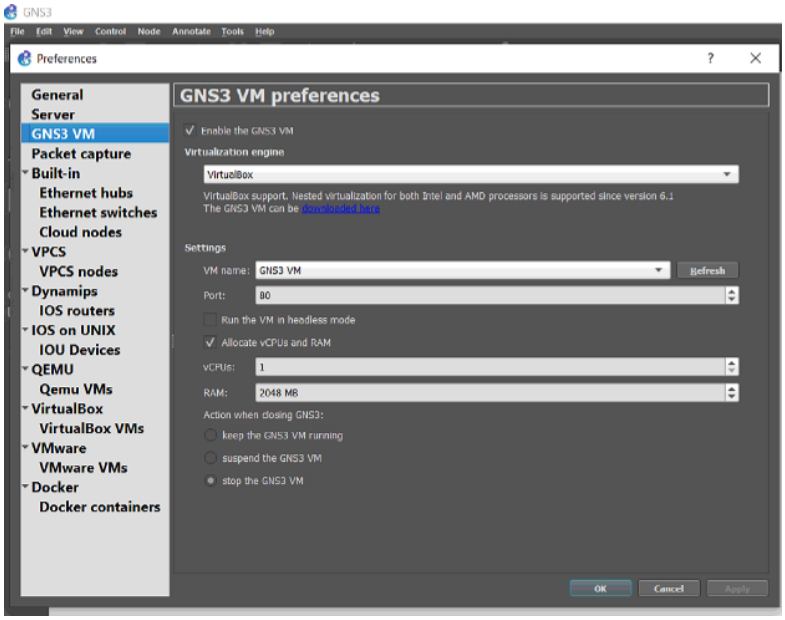


Figura 16. Habilitación de GNS3 VM

b. Después se debe agregar una imagen RouterOS, la cual se puede descargar desde su página oficial mikrotik.com, se agregó al apartado QEMU->Qemu VMs como se muestra en la *Figura 17* y también se detalla en el **ANEXO 9**.

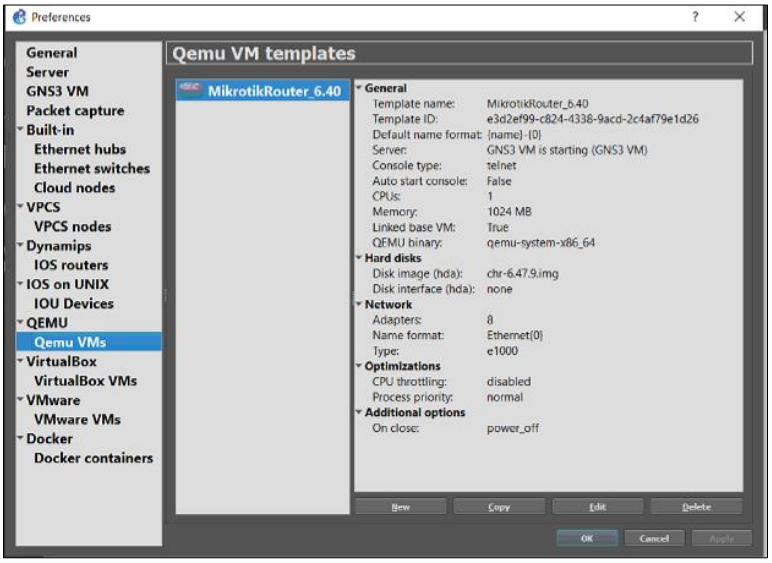


Figura 17. Importación de RouterOS en GNS3

c. Luego se añade las Virtualbox VMs necesarias para cumplir la topología propuesta, que se encuentran en el apartado Virtualbox como se aprecia en la *Figura 18.*

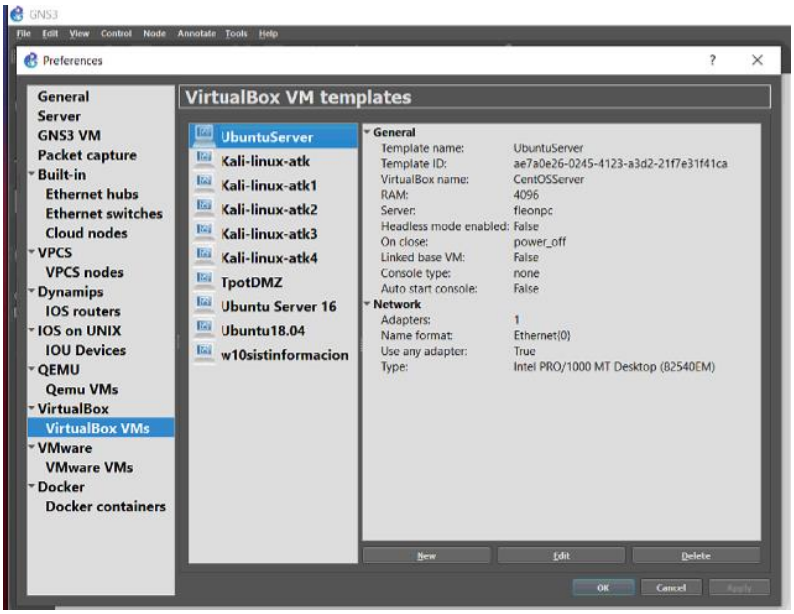


Figura 18. Importación de Máquinas Virtuales

d. Para colocar los elementos *como: máquinas virtuales, Router, switch*, ethernet, se arrastró uno por uno los elementos de la lista general a la izquierda de la zona de trabajo como se observa en la *Figura 19.*

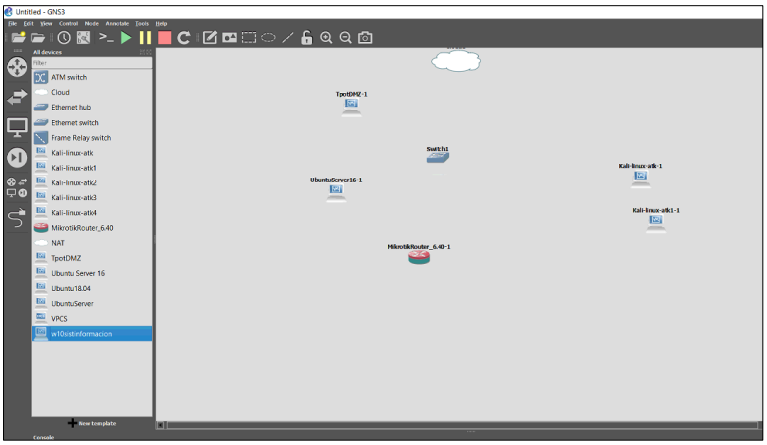


Figura 19. Elementos de GNS para plan de prueba

e. Para unir los dispositivos de la manera especificada**,** se usó el conector “Add a link” para unir uno a uno los puertos como se muestra la *Figura 20.*

Figura 20. Añadir enlace

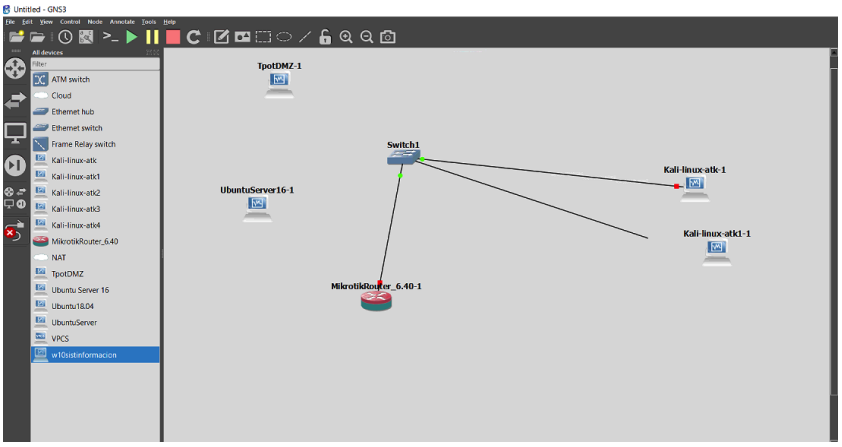


Figura 21. Equipos entrelazados

f. Finalmente se acomodó los dispositivos y se modificó los iconos, para diferenciar las maquinas entre sí, como en la *Figura 22*

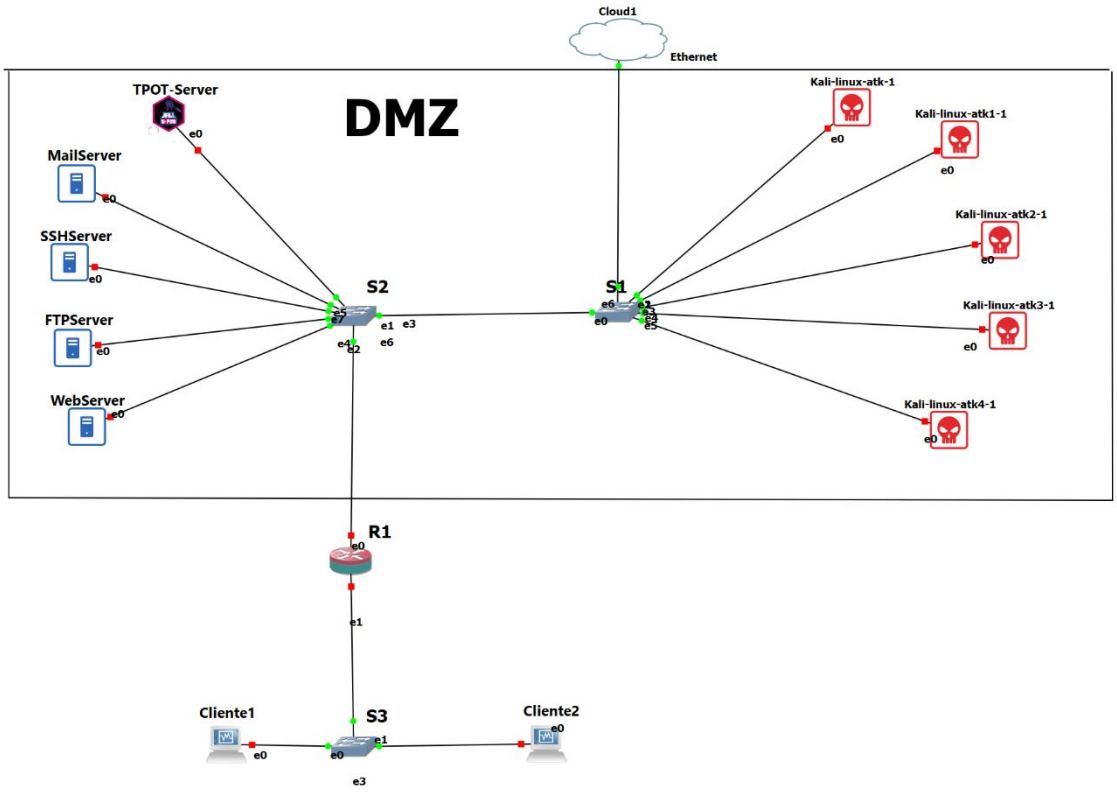


Figura 22. Topología o Escenario

6.3.7 Ejecución del plan de pruebas

Para la ejecución del plan de pruebas indicado anteriormente, se puso en funcionamiento la topología, en especial 5 máquinas virtuales con el SO Kali-Linux, así

como el servidor TPOT. Con los comandos ya explicados anteriormente, se fue ejecutando uno por uno desde la máquina correspondiente como se muestra en la *Figura 23 y Figura 24.* Además, se puede observar la ejecución de los comandos en cada maquina en el **ANEXO 11**.

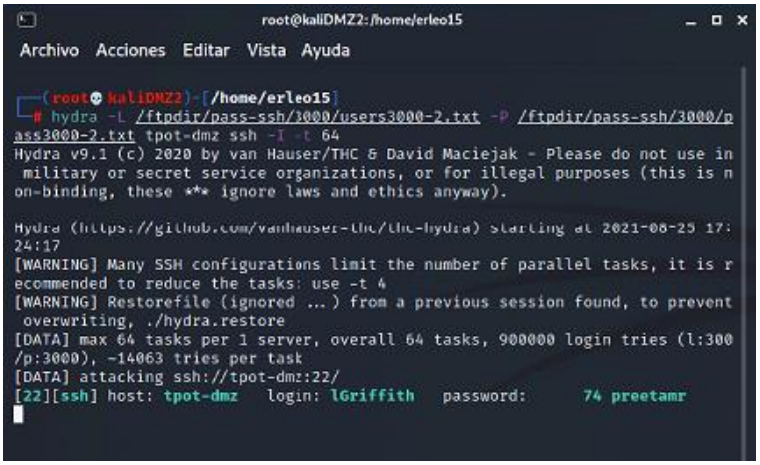


Figura 23. Ataque al servidor SSH

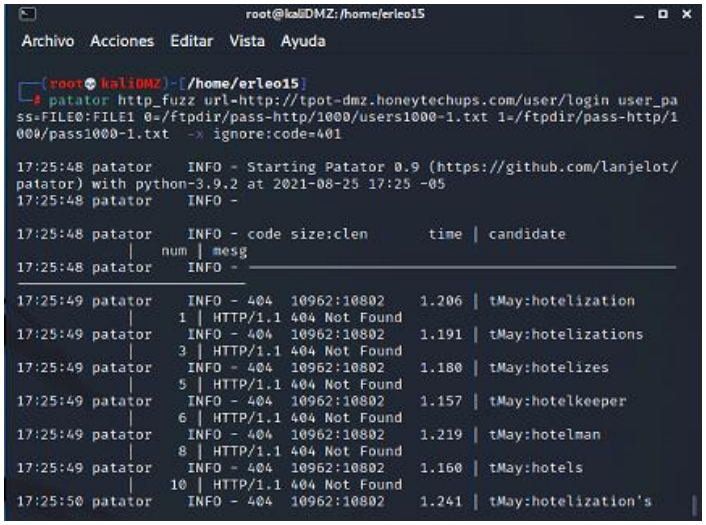


Figura 24. Ataque al servicio HTTP

Así se realizó caso por caso los ataques al servidor y este a su vez, gracias a los honeypots, se encargó de recolectar todo lo referente a los intentos de conexión, así como el número de ataques registrado, tal y como se puede ver en la *Figura 25.*

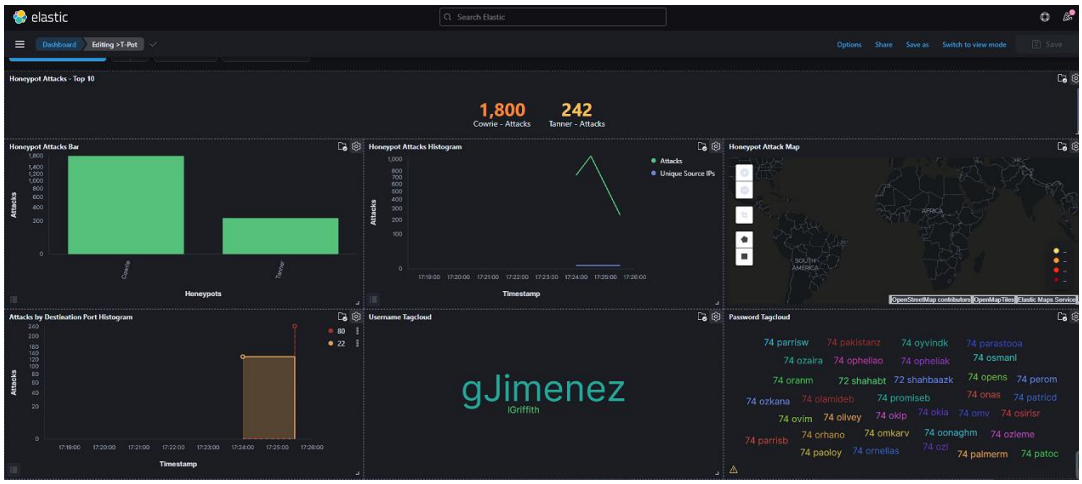


Figura 25. Ataques Registrados

Como se aprecia en la *Figura 25*, a cada ataque receptado se activó un honeypot correspondiente, así mismo va registrando los datos que ha usado para tratar de ingresar al su sistema entre otra información que puede resultar útil a la hora de analizar los ataques.

6.4 Resultados y Análisis

Este capítulo presenta el análisis y discusión de los resultados obtenidos de este estudio dando cumplimiento al cuarto objetivo *"Analizar el resultado y aportes de nuestra investigación"*. Los datos visualizados son de los cuatro honeypots mencionados en la *Tabla 2.*

Toda la información presentada a través de gráficos, representan los datos obtenidos durante el proceso de pruebas realizados con las herramientas de testing.

Los módulos que ayudaron con la interpretación de los datos se llaman Elasticsearch y Kibana, los cuales ya fueron definidos anteriormente. Para visualizar los datos ordenados por cada Honeypot, se accedió al Kibana dashboard (*Figura 26*) del servidor TPOT en la siguiente dirección:

https://hostname.domain:64297/kibana

En el **ANEXO 8 sección 4,** se detalla la forma de acceder al dashboard. También en el **ANEXO 8 sección 3** una forma de revisar los registros en archivos de texto, de la actividad de los honeypot.

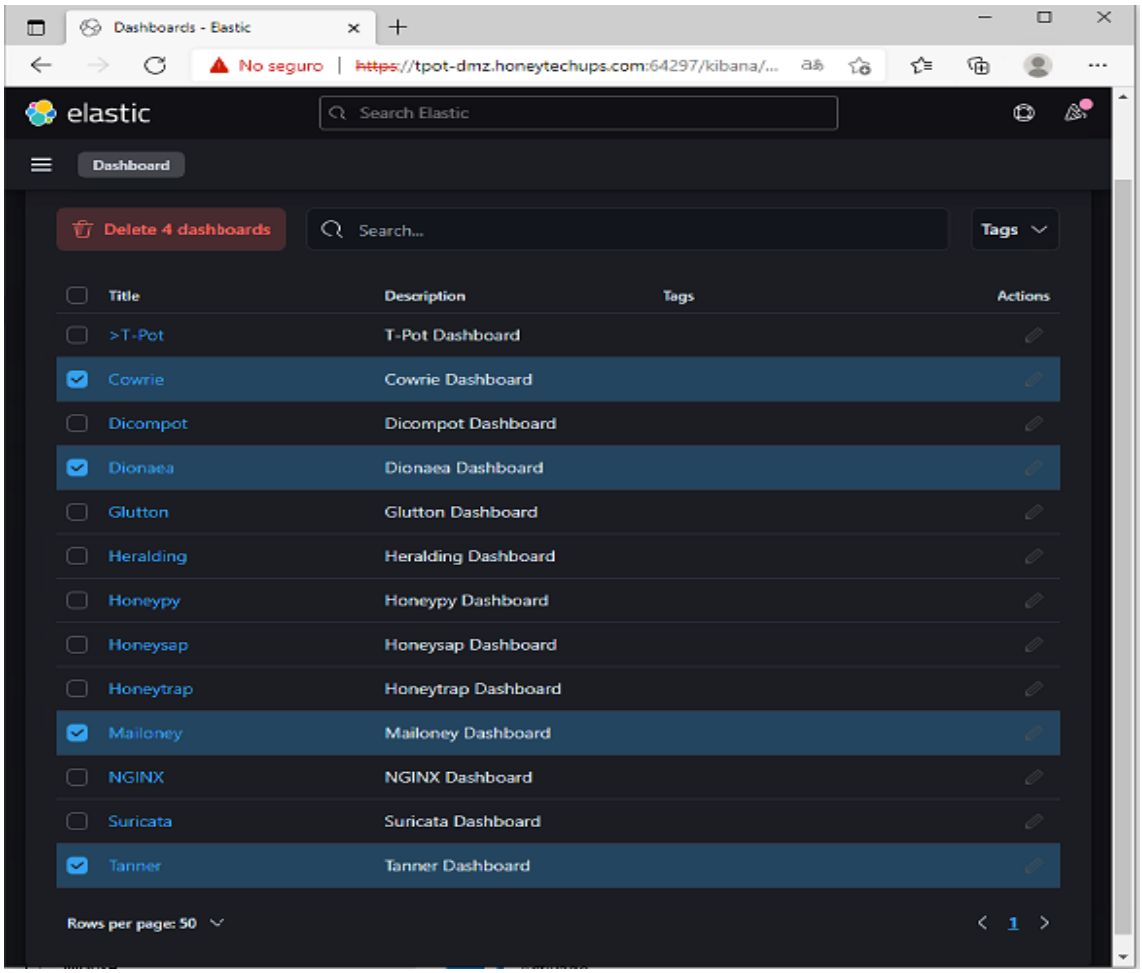


Figura 26. Kibana Dashboard

6.4.1 Análisis de resultados del honeypot Cowrie

En la *Tabla 3*, se muestra los resultados obtenidos luego de realizar las pruebas al protocolo SSH, simulado por el honeypot Cowrie. El tiempo empleado en este proceso varía en cada caso de prueba, debido al número de máquinas empleadas, así como el número de ataques realizado por cada una de ellas.

Caso	Numero de Maquinas	Numero usuarios por maquina	Numero de contrasenas por maquina	Numero de ataques por maquina atacante	Numero total de ataques	Tipo de ataque	Protocolo	Puerto	Honeypot	Cantidad detectada	Hora de Incio	Hora de finalizacion	Tiempo de ejecucion
1	1	100	1000	100.000,00	100.000,00	Fuerza Bruta	SSH	22	Cowrie	124.000,00	14/07/2021 17:00:54	14/07/2021 17:42:13	0:41:19
2	3	300	3000	900.000,00	2.700.000,00	Fuerza Bruta	SSH	22	Cowrie	2.406.116,00	15/07/2021 23:02:05	16/07/2021 11:07:05	12:05:00
3	5	500	5000	2.500.000,00	12.500.000,00	Fuerza Bruta	SSH	22	Cowrie	12.616.368,00	06/08/2021 0:16:58	07/08/2021 19:00:58	42:44:00

Tabla 3. Resultados obtenidos por el honeypot Cowrie

En la *Figura 26*, la herramienta indica el número de ataques que registró el honeypot Cowrie en distintas estancias del tiempo, así como el número de IPs que realizaron dichos ataques. Además, se tiene un histograma de ataques de acuerdo con las fechas en las cuales fueron recibidas.

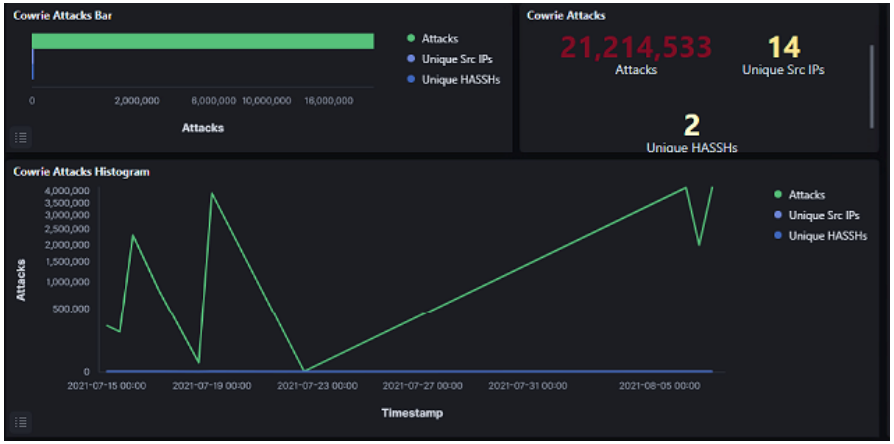


Figura 27. Ataques recibidos - Cowrie

Para saber la dirección IP de donde se realizaron los ataques se dispone del apartado “Attacker Src IP” -> “Source IP” donde hay un listado con las distintas direcciones IP, además en la columna “Count” se reflejó el número de ataque hechos por cada una de estas IP correspondientemente como se observa en la *Figura 28*.

Figura 28. Orígenes y cantidad de ataques

En la *Figura 29*, se observa una lista de comandos, los cuales fueron usados por los atacantes al momento de haber obtenido acceso con credenciales descifrados al momento de realizar los intentos de inicio de sesión. Además, se muestra una columna con el número de veces que fue usado ese comando.

Figura 29. Comandos usados en el honeypot por el atacante

6.4.2 Análisis de resultados del honeypot Dionaea

En la *Tabla 4*, se muestra los resultados obtenidos luego de realizar las pruebas al protocolo FTP, simulado por el honeypot Dionaea. El tiempo empleado en este proceso varía en cada caso de prueba, debido al número de máquinas empleadas, así como el número de ataques realizado por cada una de ellas.

Caso	Numero de Maquinas	Numero usuarios por maquina	Numero de contrasenas por maquina	Numero de ataques por maquina atacante	Numero total de ataques	Tipo de ataque	Protocolo	Puerto	Honeypot	Cantidad detectada	Hora de Incio	Hora de finalizacion	Tiempo de ejecucion
1	1	100	1000	100.000,00	100.000,00	Fuerza Bruta	FTP	21	Dionaea	5.598,00	18/07/2021 17:32:18	18/07/2021 17:34:42	12:02:24
2	3	300	3000	900.000,00	2.700.000,00	Fuerza Bruta	FTP	21	Dionaea	34.459,00	18/07/2021 18:17:51	18/07/2021 18:37:00	0:19:09
3	5	500	5000	2.500.000,00	12.500.000,00	Fuerza Bruta	FTP	21	Dionaea	102.871,00	18/07/2021 18:53:17	18/07/2021 19:44:52	0:51:35

Tabla 4. Resultados obtenidos por el honeypot Dionaea

En la *Figura 30*, la herramienta indica el número de ataques que registró el honeypot Dionaea en distintas estancias del tiempo, así como el número de IPs que realizaron dichos ataques. Gracias al histograma, se puede ver cómo se comporta los ataques a medida que transcurre el tiempo.

Figura 30. Ataques Registrados en Dionaea

Para saber la dirección IP de donde se realizaron los ataques se tiene el apartado “Attacker Src IP” -> “Source IP” donde hay un listado con las distintas direcciones IP, además en la columna “Count” presenta el número de ataque hechos por cada uno de estas IP correspondientemente como se refleja en la *Figura 31*.

Figura 31. Orígenes y cantidad de ataques

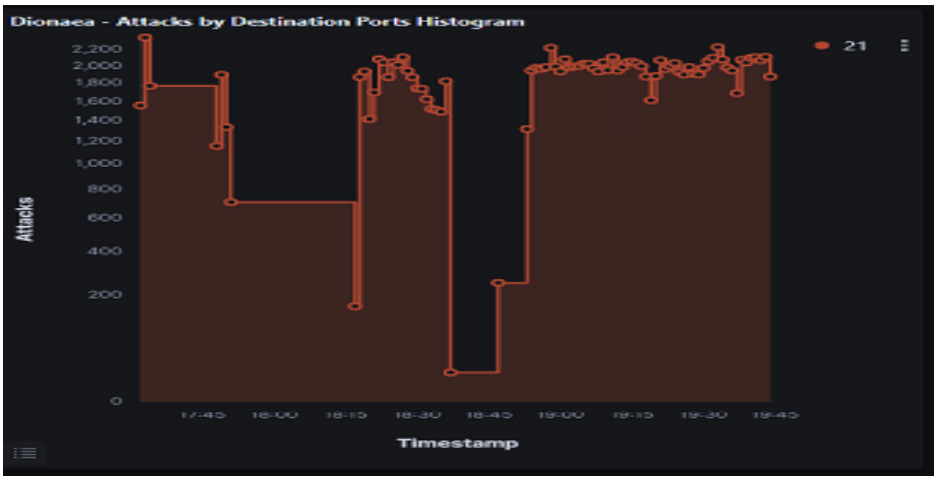


Figura 32. Puertos Atacados

6.4.3 Análisis de resultados del honeypot Mailoney

En la *Tabla 5*, se muestra los resultados obtenidos luego de realizar las pruebas al protocolo SMTP, simulado por el honeypot Mailoney. El tiempo empleado en este proceso varía en cada caso de prueba, debido al número de máquinas empleadas, así como el número de ataques realizado por cada una de ellas.

Caso	Numero de Maquinas	Numero usuarios por maquina	Numero de contrasenas por maquina	Numero de ataques por maquina atacante	Numero total de ataques	Tipo de ataque	Protocolo	Puerto	Honeypot	Cantidad detectada	Hora de Incio	Hora de finalizacion	Tiempo de ejecucion
1	1	100	1000	100.000,00	100.000,00	Fuerza Bruta	SMTP	25	Mailoney	199.961,00	25/07/2021 14:55:24	25/07/2021 15:05:07	0:09:43
2	3	300	3000	900.000,00	2.700.000,00	Fuerza Bruta	SMTP	25	Mailoney	220.991,00	25/07/2021 16:29:59	25/07/2021 19:35:43	3:05:44
3	5	500	5000	2.500.000,00	12.500.000,00	Fuerza Bruta	SMTP	25	Mailoney	2.693.846,00	25/07/2021 23:55:41	26/07/2021 13:04:41	13:09:00

Tabla 5. Resultados obtenidos por el honeypot Mailoney

En la *Figura 33*, la herramienta indica el número de ataques que registró el honeypot Mailoney en distintas estancias del tiempo, así como el número de IPs que realizaron dichos ataques. Los ataques recibidos aumentan cuando más máquinas realizan el ataque simultáneamente.

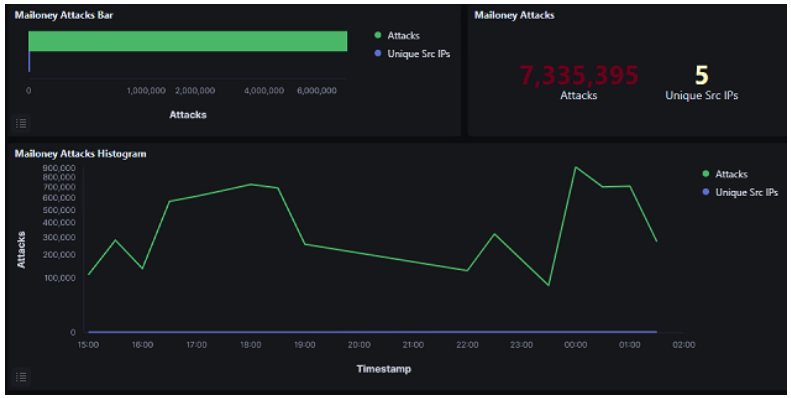


Figura 33. Ataques registrados en Mailoney

Para saber la dirección IP de donde se realizaron los ataques se tiene el apartado “Attacker Src IP” -> “Source IP” donde hay un listado con las distintas direcciones IP, además en la columna “Count” presenta el número de ataque hechos por cada uno de estas IP correspondientemente como se refleja en la *Figura 34*.

Figura 34. Orígenes y cantidad de ataques

6.4.4 Análisis de resultados del honeypot Tanner

En la *Tabla 6*, se muestra los resultados obtenidos luego de realizar las pruebas al protocolo HTTP, simulado por el honeypot Tanner. El tiempo empleado en este proceso varía en cada caso de prueba, debido al número de máquinas empleadas, así como el número de ataques realizado por cada una de ellas.

Caso	Numero de Maquinas	Numero usuarios por maquina	Numero de contrasenas por maquina	Numero de ataques por maquina atacante	Numero total de ataques	Tipo de ataque	Protocolo	Puerto	Honeypot	Cantidad detectada	Hora de Incio	Hora de finalizacion	Tiempo de ejecucion
1	1	100	1000	100.000,00	100.000,00	Fuerza Bruta	HTTP	80	Tanner	702,00	22/07/2021 17:35:49	22/07/2021 17:36:43	0:00:54
2	3	300	3000	900.000,00	2.700.000,00	Fuerza Bruta	HTTP	80	Tanner	14.999,00	24/07/2021 16:52:42	24/07/2021 17:17:35	0:24:53
3	5	500	5000	2.500.000,00	12.500.000,00	Fuerza Bruta	HTTP	80	Tanner	112.814,00	24/07/2021 19:23:17	25/07/2021 5:48:06	10:24:49

Tabla 6. Resultados obtenidos por el honeypot Tanner

En la *Figura 35*, la herramienta nos indica el número de ataques que registró el honeypot Tanner en distintas estancias del tiempo, así como el número de IPs que realizaron dichos ataques.

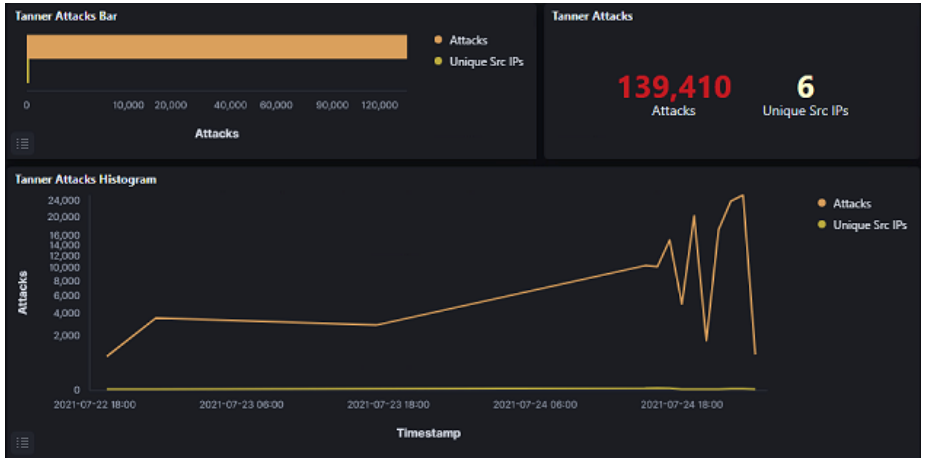


Figura 35. Ataques registrados en Tanner

Para saber la dirección IP de donde se realizaron los ataques se tiene el apartado “Attacker Src IP” -> “Source IP” donde hay un listado con las distintas direcciones IP, además en la columna “Count” presenta el número de ataque hechos por cada uno de estas IP correspondientemente como se refleja en la *Figura 36*.

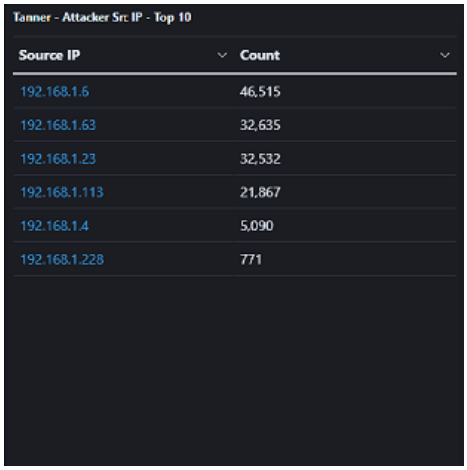


Figura 36. Orígenes y cantidad de ataques

En la *Figura 37*, se observa una lista de Identificadores Uniforme de Recursos (URIs) a donde se accedió, por parte del atacante, la cual será única e inequívoca. Además, se muestra una columna con el número de veces que fue usado ese comando.

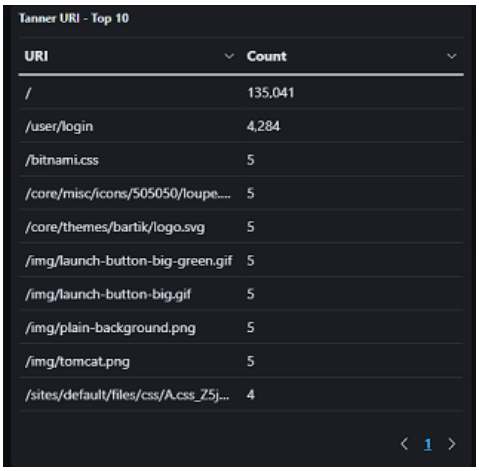


Figura 37. URIs y cantidad de veces accedida

6.4.5 Aportes de los honeypots

Los honeypot llegan a ser muy útiles cuando se requiere información de lo expuestos o comprometidos que pueden estar los servicios empresariales. Es muy conveniente saber qué tipo de honeypot requerimos para cada caso que se presente, además de la información que se requiere saber. El hecho de colocar un honeypot en un

lugar en la red donde su exposición sea mayor, lo hará mas interesante para un ente externo que trate de vulnerar la seguridad de alguno de los servidores presentes en la infraestructura de la empresa. Es por esto por lo que los servidores de la empresa deben de mantener unas estrictas políticas de seguridad, configuración y actualización, ya que esto hace que disminuya el nivel de interés del atacante por vulnerar esos servidores.

Por otra parte, y contrario a los servidores principales de la empresa, el honeypot debe de estar lo mas vulnerable posible por ejemplo la configuración por defecto, bajo nivel de defensa, entro otros. Esto hará que el nivel del interés de un atacante hacia este recurso de la red aumente considerablemente y opte por realizar ataques con técnicas que ya sean conocidas o incluso nuevas técnicas que el haya diseñado, lo que ayuda a la empresa a saber nuevas formas de ataque a sus servidores y reforzar sus políticas de seguridad en los servidores principales.

Los servidores que se usen para implementar los honeypot deben de tener características casi similares a los de los servidores principales debido a que un honeypot que no responda rápidamente las peticiones que realice el atacante, hará que pierda el interés por seguir probando sus ataques en ese servidor. También, los ataques simultáneos y numerosos hacen que el servidor sea incapaz de registrar a los mismos. Es por eso por lo que es indispensable un servidor honeypot con buenos recursos para recolectar la mayor cantidad de información posible ya que no es de gran utilidad un honeypot que no logre registrar toda la información de los ataques que ha recibido.

CONCLUSIONES

El propósito de esta tesis fue utilizar los honeypots como medio para registrar ataques de entes externos de esta manera se puede emplear medidas de ciberseguridad en la red, es decir implementar de sistemas de seguridad en redes de informáticas debido a que es primordial proporcionar información sobre la importancia de proteger las redes, dispositivos y los sistemas ya que el objetivo es mantener la integridad, confidencialidad y disponibilidad de la información. Mediante el proceso de este trabajo el honeypot se vuelve una herramienta fundamental para detectar y comprender las amenazas debido a que el honeypot registra las actividades maliciosas en la red.

La herramienta de Honeypot es muy eficiente e informativo no solo por su recopilación de datos de amenazas y ataques, sino como una detección de intrusos haciendo de este sistema personalizable para diferentes propósitos y entornos.

En cuanto a la metodología proporcionada sobre la implementación de registro de ataques se indica los pasos a seguir y los requerimientos mínimos de hardware e implementación de sistemas operativos virtualizados para llevar a cabo la simulación en tiempo real ya que se utilizó el honeypot de alta interacción haciéndolo creer al atacante que ingresó a un sistema real. Aunque los honeypot resultan ser muy útiles al momento de analizar cuán comprometida puede estar una red o los recursos que existen dentro de ella, en algunas ocasiones no siempre se podrán recoger los datos deseados, ya que los atacantes han desarrollado nuevas y sofisticadas maneras de eludir estas formas de mantener segura una red.

Por tanto, se realizó un análisis de resultados recogidos por los honeypots Cowrie, Mailoney, Dionaea y Tanner, los cuales simulan los diferentes protocolos SSH, SMTP, FTP y HTTP respectivamente. Donde se observó que el honeypot que requería de más tiempo para las pruebas es Cowrie con un tiempo aproximado de 55 horas, mientras que el de menor tiempo fue Dionaea con una marca aproximada de 1 hora.

Además, el honeypot que registró la mayor cantidad de ataques es Cowrie con al menos 14 millones de ataques registrados y el de menor registro de ataques fue Tanner con 126 mil registros. Es por esto por lo que, para recolectar el mayor número de ataques, es necesario tener una maquina con buenos recursos además de seleccionar un honeypot que aproveche esos recursos.

En conclusión, el objetivo central *"Desplegar un Honeypot como servicio, como primer perímetro de seguridad usando herramientas OpenSource."* de esta tesis, fue realizado satisfactoriamente ya que los honeypots llegan a ser un gran soporte de seguridad, gracias a sus sofisticados métodos de registro de eventos, además de la variedad de los datos que estos recopilen, permitirán en gran medida, realizar cambios en las políticas de seguridad dentro de una red empresarial, lo que reforzara la seguridad dentro de esta. En otras palabras, estos honeypots, al ser virtuales, se pueden configurar y ejecutarlos en cualquier red, en donde se desee realizar un estudio profundo del

comportamiento de dicha red, haciéndolo que se minimice el riesgo para los demás recursos que se encuentren conectados a la red.

Trabajos futuros, acorde con las actividades realizadas en esta tesis y los resultados obtenidos, se plantea una futura implementación de la tesis a nivel público, es decir, exponer la infraestructura actual a internet, lo que ayudará a un estudio de las formas de ataques que reciban los honeypots. También se puede realizar un estudio con un mayor numero de honeypots presentes en TPOT. Además, la información que reflejen los honeypots en una infraestructura expuesta a internet, ayudaran a que se implementen mecanismos de seguridad como el cifrado en las comunicaciones o diferentes tipos de autenticación.

REFERENCIAS

[1] B. Rababah, S. Zhou y M. Bader, «Evaluation the Performance of DMZ,» *Modern Education and Computer Science*, vol. 8, nº 1, p. 13, 2018.

[2] M. Keri, «DICOM Honeypot,» Github, 22 Octubre 2020. [En línea]. Available: https://github.com/nsmfoo/dicompot. [Último acceso: 19 Agosto 2021].

[3] J. Chirillo y Sons, Hack Attcks Revealed: A Complete Reference with Custom Security Hacking Toolkit, 2001, p. 837.

[4] L. Spitzner, Honeypots: Tracking Hackers, Boston: Pearson Education, Inc., 2002.

[5] F. Cocaro, M. García y M. J. Rouiller, «Diseño e Implementación de un Honeypot,» Facultad de Ingenieria de la UdelaR, Montevideo, 2008.

[6] A. A. Gómez, «Análisis de vulnerabilidades en IoT para el despliegue de Honeypots,» Lálaga, 2018.

[7] R. Campbell, K. Padayachee y T. Masombuka, «A survey of honeypot research: Trends and opportunities,» de *10th International Conference for Internet Technology and Secured Transactions*, 2015.

[8] K. Sadasivam, B. Samudrala y A. Yang, «Design of Network Secury Projects Using Honeypot,» 2016.

[9] J. G. Ruiz, «Cibercanario: un honeypot básico, tangible y usable para entornos IoT,» Málaga, 2020.

[10] P. Sokol, M. Zuzčák y T. Sochor, «Definition of Attack in Context of High Level Interaction Honeypots,» de *Software Engineering in Intelligent Systems Advances in Intelligent Systems and Computing*, 2015.

[11] J. Figueroa Suárez, R. Rodriguez Andrade, C. Bone Obando y J. Saltos Gomez, «La seguridad informática y la seguridad de la información,» *Polo del conocimiento*, p. 11, 2017.

[12] N. ISO, «REFERENCIAS NORMATIVAS ISO 27000,» [En línea]. Available: https://normaiso27001.es/referencias-normativas-iso-27000/. [Último acceso: 2 junio 2021].

[13] A. Rashid, «Data Center Architecture Overview,» *ReserchGate*, vol. 28, p. 11, 2019.

[14] W. P. Turner IV, J. Seader, V. Renaud, PE y K. Brilwwe, «Tier Classifications Define Site Infrastructure Performance,» UPTIME INSTITUTE, Santa Fe, 2008.

[15] E. Rafter, «The Data Center Tier Performance Standars and Their Importance to the Owner's Project Requirements,» Tier IV Consulting Group, Chicago, 2007.

[16] T. Daim, J. Justice, M. Krampits, M. Letts, G. Subramanian y M. Thirumalai, «An energy efficiency model for information technology managers,» emeraldinsight, Oregon, 2009.

[17] J. D. PEREZ R., «PLANEACIÓN DE DISEÑO DATACENTER TIER III,» Politecnico Grancolombiano, Bogotá, 2016.

[18] L. A. Barroso y U. Hölzle, «The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines,» Google Inc., Madison, 2009.

[19] J. Bolaños B., «Diseño de la arquitectura de seguridad perimetral de la red informatica en la industria de licores del Valle,» 28 Febrero 2018. [En línea]. Available: https://red.uao.edu.co/bitstream/handle/10614/10248/T07892.pdf?sequence=4&isAllowed=y. [Último acceso: 25 Agosto 2021].

[20] M. A. L. Parra, «DISEÑO DE PROCEDIMIENTOS DE SEGURIDAD BASADOS EN PRUEBAS DE PENTESTING APLICADAS A LA EMPRESA CJT&T INGENIERÍA DE SOFTWARE,» 2017.

[21] S. V. Dhavale, Constructing an Ethical Hacking Knowledge Base for Threat Awareness and Prevention, Unite Satate: IGI Global, 2018, p. 281.

[22] D. Narváez, C. Romero y M. Nuñez, «Evaluación de ataques de Denegación de servicio DoS y DDoS, y,» *DECC Report, Tendencias en Computación.*, vol. 1, nº 2, p. 13, 2010.

[23] T. Jama, P. Amaral, A. Kha, A. Zameer, K. Ullah y S. Aziz Butt, «Denial of Service Attack in Wireless LAN,» de *International Conference on Digital Society*, Rome, 2018.

[24] A. Lamba, S. Singh y B. Singh, «MITIGATING ZERO-DAY ATTACKS IN IOT USING A STRATEGIC FRAMEWORK,» *International Journal For Technological Research In Engineering* , vol. 4, nº 1, p. 4, 2016.

[25] U. Sarmah, D. Bhattacharyya y J. Kalita, «A survey of detection methods for XSS attacks,» *Journal of Network and Computer Applications*, vol. 118, p. 52, 2018.

[26] E. L. Montaño Rivas, *Evaluación de las Vulnerabilidades que Presentan los Firewalls en la Empresa DATASOLUTION S.A.*, Guayaquil: Universidad de Guayaquil. Facultad de Ciencias Matematicas y Fisicas. Carrera de Ingeniería en Sistemas Computacionales. Carrera de Ingeniería en Networking y Telecomunicaciones., 2011.

[27] D. G. Cortes, «FIREWALLS DE NUEVA GENERACIÓN: La Seguridad Informática Vanguardista,» de *Seminario de Investigación Aplicada*, Colombia, 2016.

[28] A. AlEroud y G. Karabatis, «SDN-GAN: Generative Adversarial Deep NNs for Synthesizing Cyber Attacks on Software Defined Networks,» p. 11, 2020.

[29] J. S. Perez y F. Mora Gimeno, «IDS de red para la detección de ataques sobre SSH y FTP,» 2020.

[30] T. Secury, «Introduction into T-Pot: A Multi-Honeypot Platform,» T-Pot , 17 Marzo 2015. [En línea]. Available: https://github.security.telekom.com/2015/03/honeypot-tpot-concept.html#t-pot. [Último acceso: 30 junio 2021].

[31] . I. Zymberi, «Honeypots: A Means of Sensitizing Awareness of Cybersecurity Concerns,» 1 Mayo 2021. [En línea]. Available: https://www.theseus.fi/bitstream/handle/10024/496070/Zymberi_Ilirjana.pdf?sequence=2&isAllowed=y. [Último acceso: 10 Agosto 2021].

[32] A. Higgins, «Adaptive Containerised Honeypots for Cyber-Incident Monitoring,» 2018.

[33] C. Kelly, N. Pitropakis, A. Mylonas, S. McKeown y W. Buchanan, «A Comparative Analysis of Honeypots on Different Cloud Platforms,» *Sensors*, vol. |, nº 1, p. 6, 2021.

[34] W. H. Chong y C. K. Koh, «Learning cyberattack patterns with active honeypots,» DTIC, Monterey, 2018.

[35] M. Hänninen, «Organisaation sisäverkon tilannekuvan parantaminen hunajapurkkituotteita hyödyntäen,» *Theseus*, vol. |, nº 1, p. 41, 2020.

[36] A. Bachelet y A. Moussard, 1 Enero 2020. [En línea]. Available: https://arthurbachelet.me/assets/Project/Files/Honeypots.pdf. [Último acceso: 19 Agosto 2021].

[37] J. Banfi Vázquez, «POC: CAPTURA DE MALWARE CON EL HONEYPOT DIONAEA - PARTE I,» UNAM, 1 Enero 2020. [En línea]. Available: https://revista.seguridad.unam.mx/numero23/poc-captura-de-malware-con-el-honeypot-dionaea-parte-i. [Último acceso: 19 Agosto 2021].

[38] J. Vestergaard, «GitHub - johnnykv/heralding: Credentials catching honeypot,» 27 Diciembre 2020. [En línea]. Available: https://github.com/johnnykv/heralding. [Último acceso: 18 Agosto 2021].

[39] T. Trajanovski, An Automated and Comprehensive Framework for IoT Botnet Detection and Analysis (IoT-BDA), Manchester: School of Computer Science, The University of Manchester, UK, 2021.

[40] Cymmetria, «CiscoASA Honeypot,» Github, 16 Agosto 2018. [En línea]. Available: https://github.com/Cymmetria/ciscoasa_honeypot. [Último acceso: 10 Agosto 2021].

[41] C. D. Serrano Ávila y M. Rúiz Castaño, «Diseño e implementación de un honeypot en la línea de negocio Facturación electrónica en la empresa jaime torres c y cia,» Bogotá, 2021.

[42] SECUREAUTH LABS, «HoneySAP: SAP Low-interaction honeypot,» SecureAuth Corporation, 2021 Junio 2021. [En línea]. Available: https://honeysap.readthedocs.io/en/latest/. [Último acceso: 19 Agosto 2021].

[43] E. E. V. Fernández, «Virtualización de servidores,» Almería, 2010.

[44] C. O. J. Morales, «Compilación Unidad Témática: Sistemas Operativos,» Florencia, 2015.

[45] K. L. O. Documentation, «¿Qué es kali linux?,» 2015.

[46] A. Kouka, Ubuntu Server Essentials, Abdelmonam Kouka, 2015.

[47] Windows, Windows 10: preparación para la certificación MCSA, ENI.

[48] D. H. Castro, «DISEÑO E IMPLEMENTACIÓN DE LA INTERCONEXIÓN DE SUCURSALES DE HP-STORE EN LAS CIUDADES DE AREQUIPA Y CUSCO MEDIANTE VPN CON MIKROTIK ROUTER,» Arequipa, 2019.

[49] Galaxy Technologies LLC, «GNS3,» 2021. [En línea]. Available: https://docs.gns3.com/docs/. [Último acceso: 29 7 2021].

[50] QEMU, «Qemu,» Virtualizacion Full-System emulation, 2020. [En línea]. Available: https://www.qemu.org/. [Último acceso: 14 junio 2021].

[51] Elasticsearch, «¿Qué es Elasticsearch?,» [En línea]. Available: https://www.elastic.co/es/what-is/elasticsearch. [Último acceso: 2 julio 2021].

[52] P. Kleindienst, «Building a real-world logging infrastructure with Logstash, Elasticsearch an Kibana,» Logstash, [En línea]. Available: https://www.elastic.co/es/logstash/. [Último acceso: 1 julio 2021].

[53] P. Kleindienst, «Building a real-world logging infrastructure with Logstash, Elasticsearch and Kibana».

[54] NMAP, «NMAP,» [En línea]. Available: https://nmap.org/. [Último acceso: 11 julio 2021].

[55] G. Yimin y Z. Zhang, «LPSE: Lightweight password-strength,» *Computers & Security*, vol. 73, p. 517, 2017.

[56] A. Álvarez Vilchez, «SNARE es un honeypot de aplicaciones web y es el sucesor de Glastopf, que tiene muchas de las mismas características que Glastopf, así como la capacidad de convertir páginas web existentes en superficies de ataque con TANNER.,» 01 Marzo 2017. [En línea]. Available: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/59528/8/aalvarezvilTFG0117memm%C3%B2ria.pdf. [Último acceso: 2021 Agosto 10].

[57] Telekom Security, «Digitalisierung braucht Security - Magenta Security | Telekom Geschäftskunden,» 20 Agosto 2020. [En línea]. Available: https://github.security.telekom.com/2020/08/honeypot-tpot-20.06-released.html. [Último acceso: 10 Agosto 2021].

[58] M. T. González, «Desarrollo de un entorno MPLS basado en GNS3,» Valencia, 2019.

[59] L. D. Cervantes, «Evaluación de la herramienta GNS3 con,» 2014.

[60] G. V. Alvarez, «SEGURIDAD EN REDES IP: Honeypots y Honeynet,» vol. IV, p. 37.

[61] F. Pouget y D. Hervé, «Honeypot, Hooneytoken: Terminological issues,» Eurecom, Francia, 2003.

[62] E. J. S. Cabrera, *"Análisis basado en teoría de juegos de modelos de negocio de operadores móviles virtuales en redes 4G y 5G"*. Valencia: Universitat Politècnica de València, 2021.

[63] E. J. S. Cabrera, L. Guijarro, y J. R.Vidal, “Economic feasibility of virtual operators in 5G via network slicing”, *Futur. Gener. Comput. Syst. vol. 109*, pp. 172–187, 2020.

[64] E. J. S. Cabrera, L. Guijarro, y P. Maillé, “Game theoretical analysis of a multi-MNO MVNO business model in 5G networks”, *Electron.*, vol. 9, núm. 6, pp. 1–26, 2020, doi: 10.3390/electronics9060933.

[65] E. J. Sacoto-Cabrera, A. Sanchis-Cano, L. Guijarro, J. R. Vidal, y V. Pla, “Strategic Interaction between Operators in the Context of Spectrum Sharing for 5G Networks”, *Wirel. Commun. Mob. Comput.*, vol. 2018, 2018, doi: 10.1155/2018/4308913.

[66] A. Sanchis-Cano, J. Romero, E. J. Sacoto-Cabrera, y L. Guijarro, “Economic feasibility of wireless sensor network-based service provision in a duopoly setting with a monopolist operator”, *Sensors (Switzerland)*, vol. 17, núm. 12, pp. 1–22, 2017, doi: 10.3390/s17122727.

[67] V. Vimos y E. J. Sacoto Cabrera, “Results of the implementation of a sensor network based on Arduino devices and multiplatform applications using the standard OPC UA”, *IEEE Lat. Am. Trans.*, vol. 16, núm. 9, pp. 2496–2502, 2018, doi: 10.1109/TLA.2018.8789574.

[68] E. Sacoto-Cabrera, J. Rodriguez-Bustamante, P. Gallegos-Segovia, G. Arevalo-Quishpi, y G. León-Paredes, “Internet of things: Informatic system for metering with communications MQTT over GPRS for smart meters”, *2017 Chil. Conf. Electr. Electron. Eng. Inf. Commun. Technol. CHILECON 2017 - Proc.*, vol. 2017-Janua, núm. October 2017, pp. 1–6, 2017, doi: 10.1109/CHILECON.2017.8229598.

[69] V. Vimos y E. J. Sacoto Cabrera, “Results of the implementation of a sensor network based on Arduino devices and multiplatform applications using the standard OPC UA”, *IEEE Int. Conf. Autom.*, vol. 16, núm. 9, pp. 2496–2502, 2016, doi: 10.1109/TLA.2018.8789574.

[70] E. J. S. Cabrera, S. Plaguachi, G. Leon, P. G. Segovia, y G. B. Quezada, “‘Industrial Communication Based on MQTT and Modbus Communication Applied in a Meteorological Network’”, *Int. Conf. Adv. Emerg. Trends Technol.*, 2020.

ANEXOS

ANEXO 1: Instalación y configuración Windows Education:

La instalación de Windows 10 Education se instalará en una máquina física:

La instalación se requiere Windows 10 Education ISO, y una USB:

Primero cargamos la ISO Windows 10 Education en la USB haciendo uso de la herramienta de Rufus.

Segundo punto al instalar Windows 10 Education aparecerá en pantalla la elección del:

- **Idioma que va a instalar,** seleccionamos el idioma **Español (Español, España)**
- **Formato de hora y moneda:** Español (Ecuador, Latinoamerica).
- **Teclado o método de entrada:** Español
- Damos clic en siguiente y presionamos en el botón **Instalar ahora.**

Después de unos minutos nos aparecerá una pantalla **Programa de instalación de Windows**, Activar Windows, en caso de no tener la clave del producto seleccionamos en el **botón No tengo clave del producto.**

- Aceptamos los términos de licencia y clic en siguiente
- Hacemos clic en **Personalizada: Instalar solo Windows**
    - Hacemos clic en la unidad del disco duro y clic en **nuevo** y asignamos la cantidad de disco duro a particionar y presionamos en **aplicar y aceptar** para crear una nueva partición del disco, y clic en **siguiente**.
    - En el proceso de instalación se demora de acuerdo con su velocidad de internet
- Una vez instalada nos aparecerá en la pantalla **Comenzar rápidamente** hacemos clic en **Personalizar configuración.**
- **Personalizar configuración** clic en **siguiente**.
- La instalación se demorará debido a que Windows contienen actualizaciones.
- Aparecerá en la pantalla **Elija como se conectará** elegimos la opción **Unirse a un Dominio** clic en s**iguiente**.
- Aparecerá en pantalla **Crear una cuenta para este equipo,** en esta sección se pondrá en nombre del Usuario y contraseña y damos clic en **siguiente**.

Figura 38. Escritorio de Windows 10 Education

ANEXO 2: Instalación y configuración VirtualBox

Para la instalación de VirtualBox se descargó del siguiente link: https://www.virtualbox.org/

Seguimos los siguientes pasos:

1. Damos clic derecho sobre VirtualBox, clic en la opción **Ejecutar como Administrador.**
2. Nos aparecerá la ventana de Oracle VM VirtualBox, damos clic en **Next.**
3. Nuevamente hacemos clic en **Next** y dejamos por defecto las opciones enmarcadas y damos clic en **Next,** posteriormente damos clic en **YES.**

4. Finalmente damos clic en **Install** una vez terminada la instalación damos clic en **Finish**

Figura 39. Ventana inicial de VirtualBox

ANEXO 3: Instalación y Configuración de GNS3

La instalación y configuración se requieren de:

GNS https://gns3.com/software/download

Pasos a seguir:

1. Damos clic derecho sobre GNS3, clic en Ejecutar como Administrador y se abrirá GNS3.
2. Damos clic continuamente en **Next** hasta que se llegue a visualizar como se ve en la *Figura 40*.

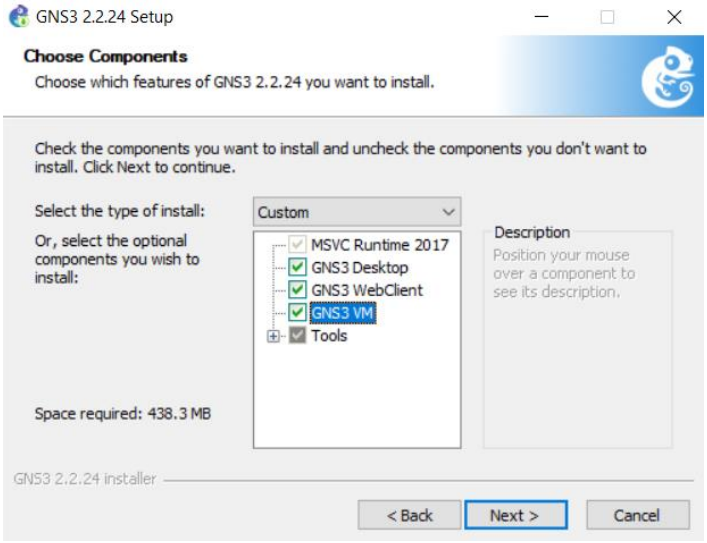


Figura 40. Enmarcamos la Opción GNS3 Desktop y VM

3. Damos clic en **next** después de enmarcar GNS3 Y VM.
4. Seleccionamos la ruta o ubicación donde se requiera instalar GNS3 como se ve en la *Figura 41 y clic en **next.***

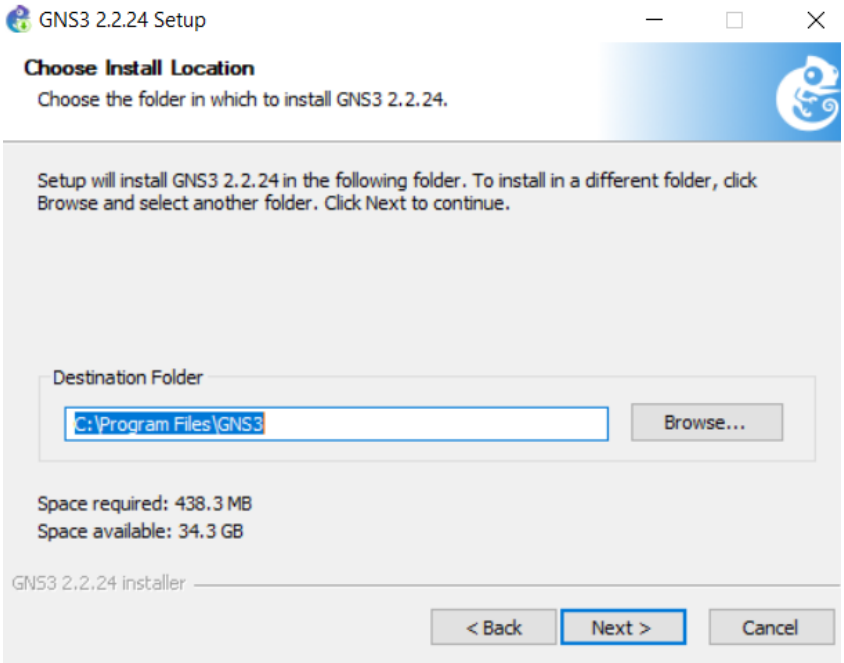


Figura 41. Ruta de instalación GNS3

5. Al instalarse GNS3 te pedirá instalar algunos programas como Npcap como se ve en la *Figura 42*, es recomendable instalar todos si es la primera vez que se instala GNS3 y damos clic en **next.**

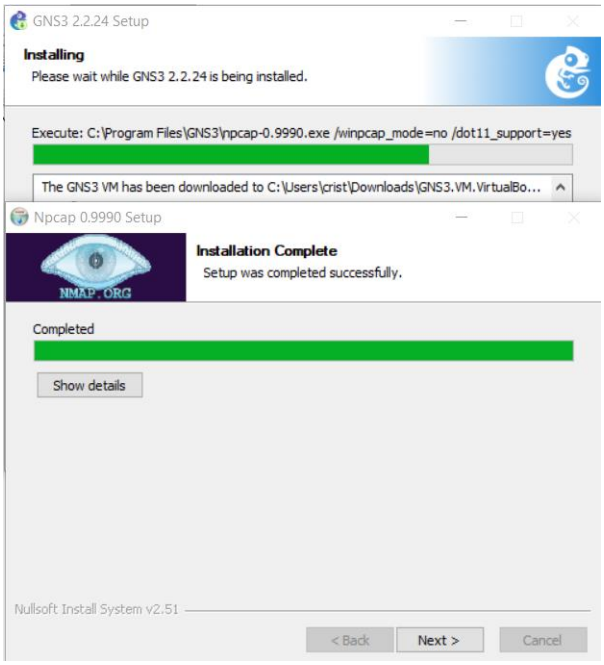


Figura 42. Instalación de programas adicionales como NCAP

6. Nos aparecerá una ventana, en la cual aceptamos el acuerdo de licencia y uso de privacidad para el uso de GNS3 y clic en **Accept,** como se ve en la *Figura 43.*

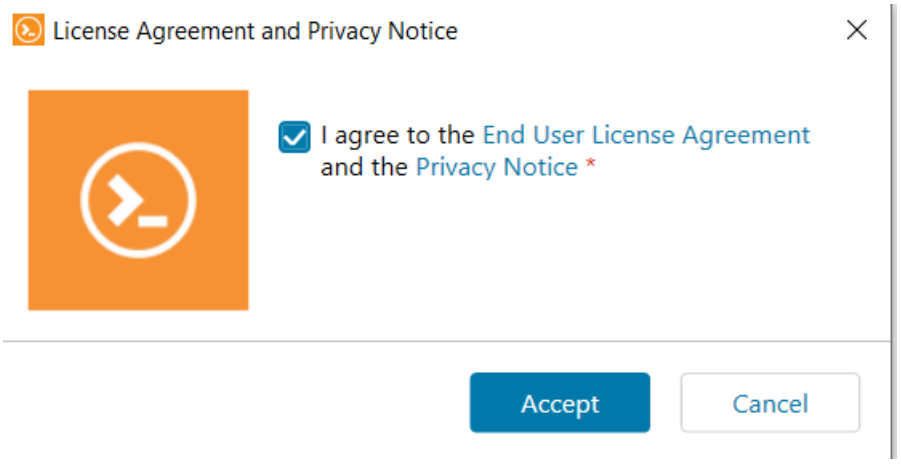


Figura 43. Aceptar el acuerdo de licencia y uso de privacidad

7. GNS3 ofrece una serie de herramientas por la cual tiene un costo, esto no es necesario, por la cual, damos un clic en **NO y** hacemos clic en **Next,** como se observa en la *Figura 44.*

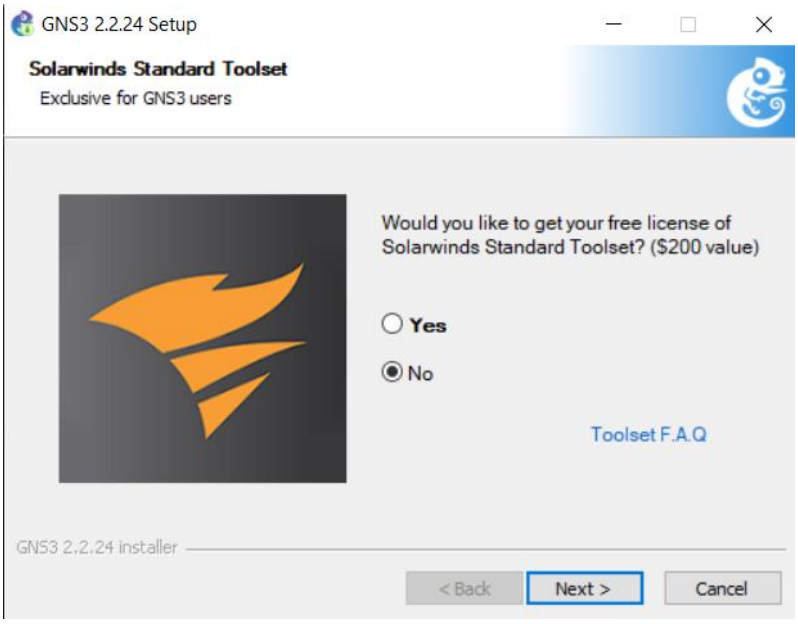


Figura 44. Solaris Standard Toolset

8. Finalmente damos clic en **finish.**

ANEXO 4: Instalación Configuración GNS3_VM

La instalación y configuración se requieren de:

GN3 VM: https://www.gns3.com/software/download-vm

Pasos a seguir:

1. Una vez descargado GNS3 VM, precedemos a cargar en VirtualBox como en la *Figura 45.*

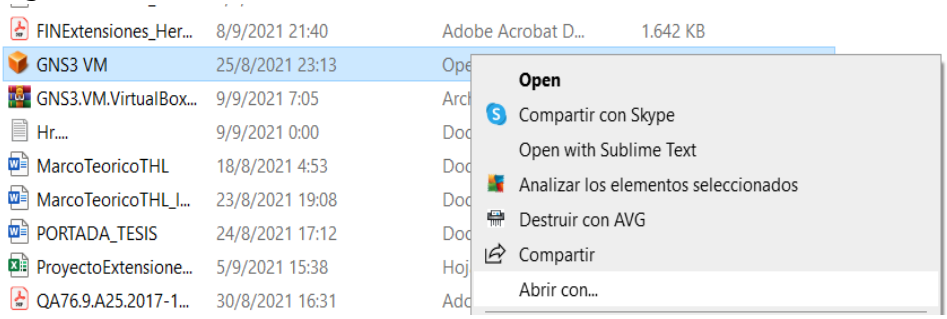


Figura 45. Abrimos GNS3 en VirtualBox

2. Abrir con VirtualBox Manager clic en **Aceptar**

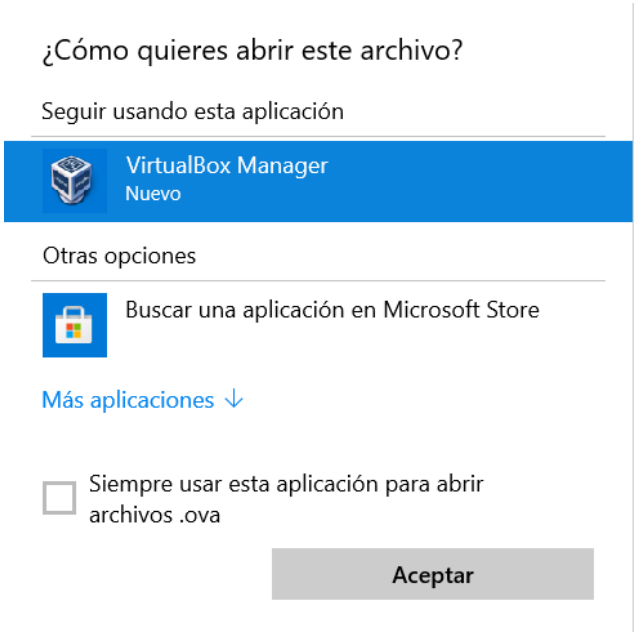


Figura 46. Abrir en VirtualBox

3. En la *Figura 47 se visualizan los detalles de GNS3 VM y damos* clic en **Importar**

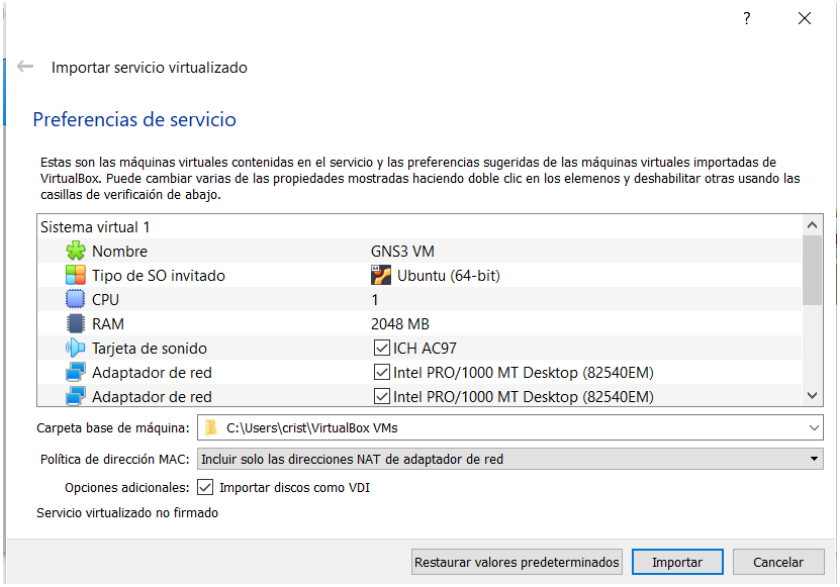


Figura 47. Importación de GNS3

4. Seleccionamos GNS3, damos clic en **Configuraciones,** nos situamos en la sección **Red** donde se describe **Habilitar adaptador de red** seleccionamos **Red NAT** y damos clic en **Aceptar** como de ejemplifica en la *Figura 48.*

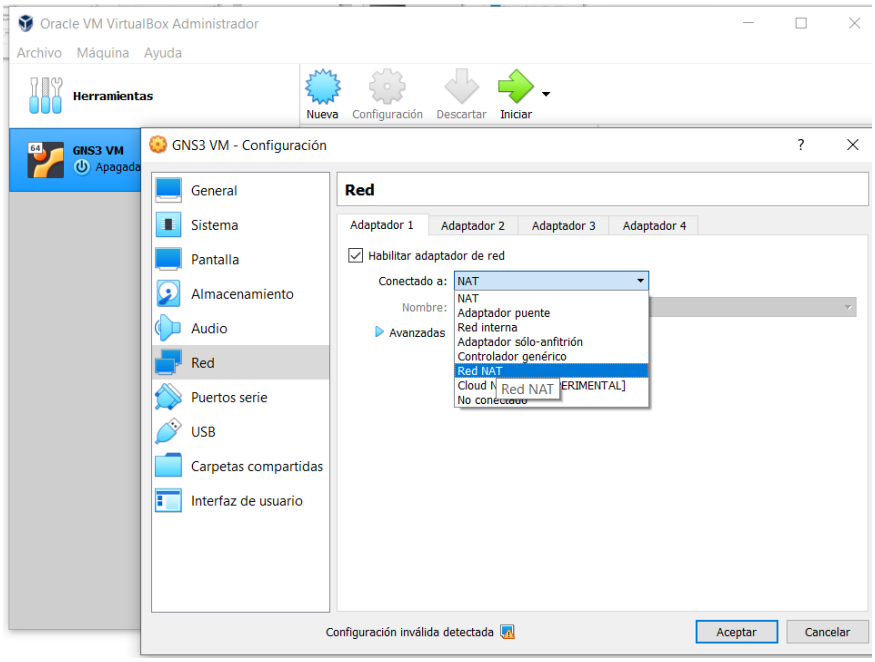


Figura 48. Habilitación de Red NAT

5. Antes de seguir con las configuraciones primero se debe verificar que el software GNS y el OVA GNS3 VM sean las mismas versiones, debido que, en el proceso, se presentan problemas de compatibilidad por ello deben ser las mismas versiones, como se observa en la *Figura 49.*

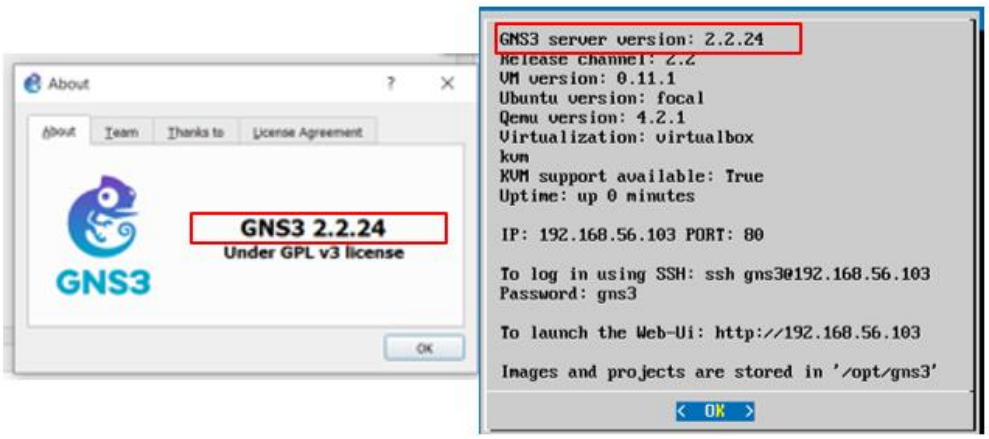


Figura 49. Versiones de GNS3 & GNS3 VM

ANEXO 5: Crear una máquina virtual en VirtualBox

1. Abrir VirtualBox, y crear una nueva máquina en la opción “Nueva”, en la *Figura 50*, se ilustra el ejemplo.

Figura 50. Ventana principal de VirtualBox

2. Se debe de añadir un nombre, la ubicación donde se guardará el archivo de máquina, como se observa en la *Figura 51*, el tipo de máquina que será (Linux, Solaris, Windows, etc.) y su versión (para Linux: Debian, Oracle, Ubuntu, etc) y damos a siguiente.

Figura 51. Opciones del tipo de maquina

3. Asignamos una cantidad de memoria RAM a la máquina, la cual representa el espacio de trabajo para las operaciones de esta, *Figura 52.*

Figura 52. Asignación de memoria RAM

4. Luego se elige un disco existente (si se tiene) o se crea uno nuevo, *Figura 53.*

Figura 53. Opciones de Disco duro

5. Después seleccionamos el tipo de archivo de disco, como se ve a continuación en la *Figura 54*.

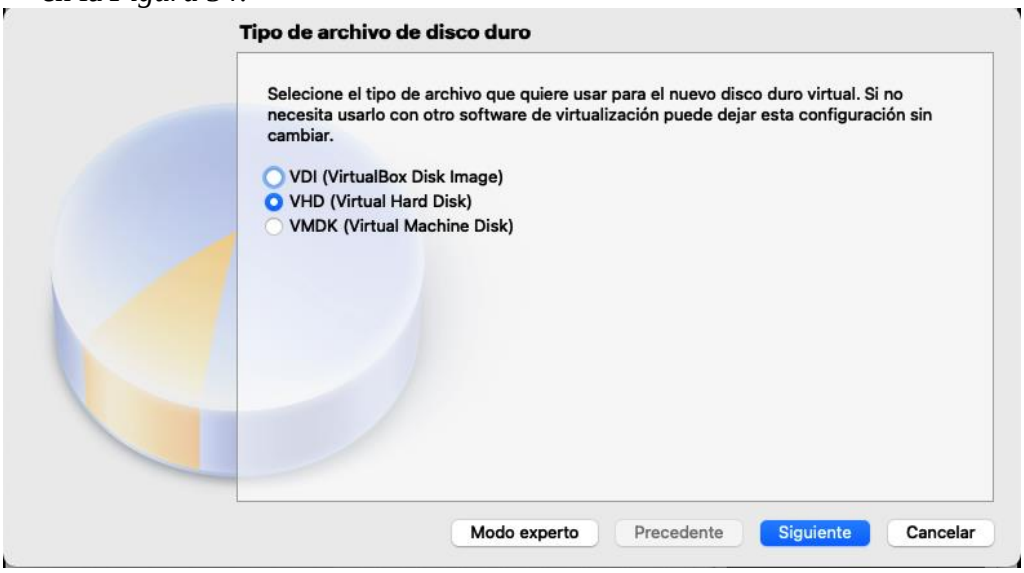


Figura 54. Tipos de archivos de disco duro

6. En esta parte seleccionamos el tipo de ocupación de espacio que tendrá el archivo de disco, *Figura 55.*

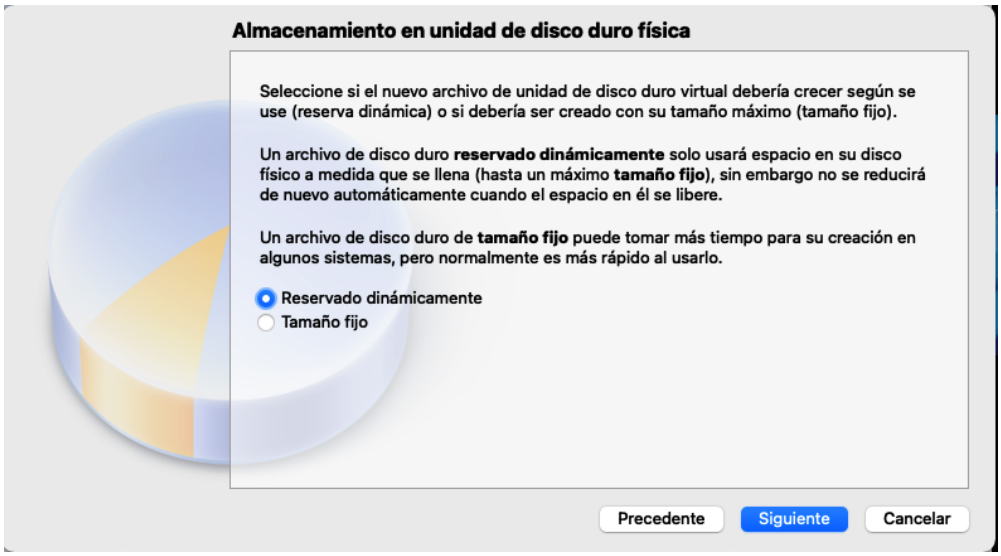


Figura 55. Tipo de almacenamiento en disco

7. Finalizamos esta parte colocando la ubicación donde se almacenará el archivo de disco, así como su tamaño, *Figura 56.*

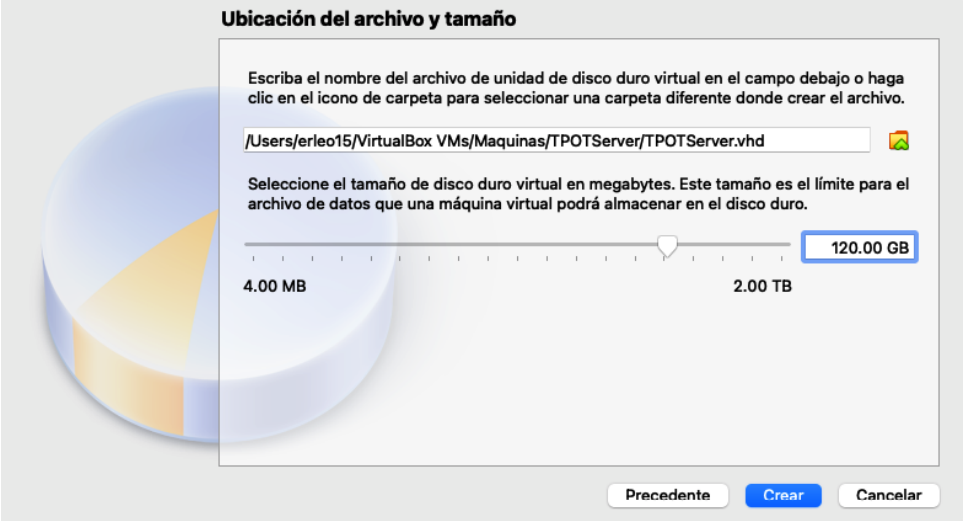


Figura 56. Tamaño de disco duro virtual

8. Luego de configurar los parámetros de la nueva máquina, esta se creará y la veremos reflejada en la lista de máquinas presentes en VirtualBox, como se muestra en la *Figura 57.*

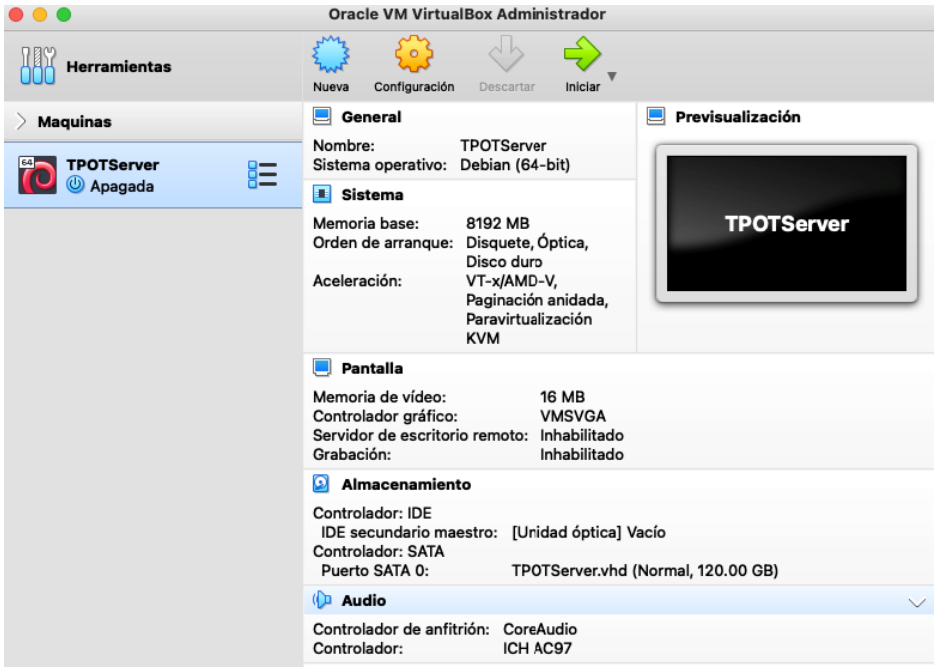


Figura 57. Ventana de VirtualBox con datos generales de la máquina virtual

ANEXO 6: Iniciar una máquina virtual en VirtualBox

1. Para iniciar la máquina creada o una que ya dispongamos, debemos de seleccionar la maquina en cuestión y hacer clic en el botón **Iniciar**, como se indica en la *Figura 58*.

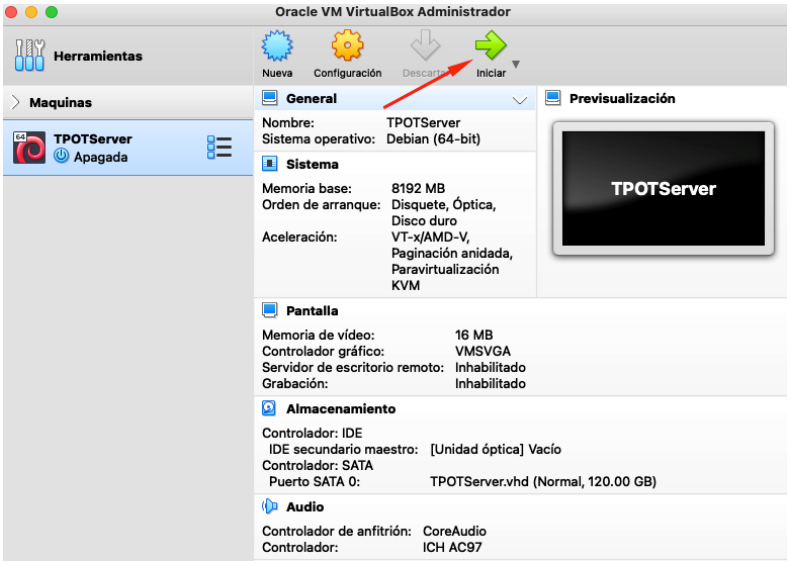


Figura 58. Selección del botón "Iniciar"

2. Si la maquina ya está lista para su uso nos arrancará en el SO que se tenga instalada, caso contrario nos aparecerá una ventana similar a la figura. Para este caso debemos de seleccionar una imagen de disco (ISO), en caso de no tener uno se deberá agregar dando clic en el icono indicado en la *Figura* 59.

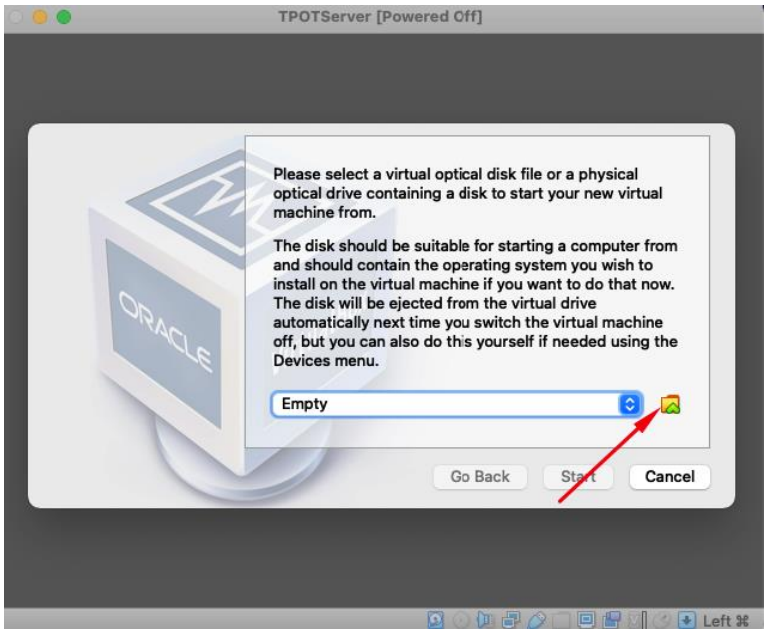


Figura 59. Ubicación donde se encuentra la ISO

3. Luego daremos clic en la opción "**Añadir**" ejemplo de la *Figura 60*.

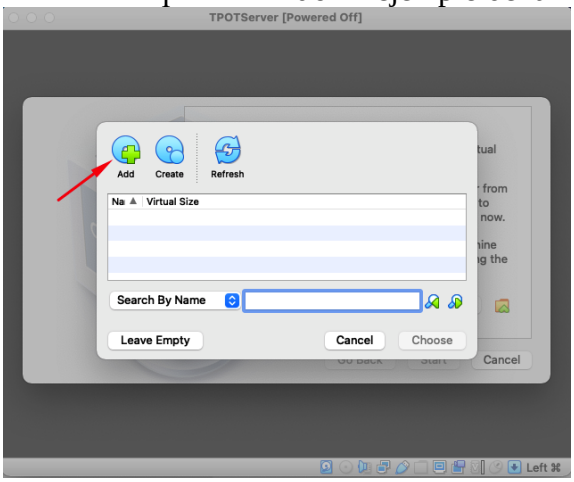


Figura 60. Botón "Añadir ISO"

4. Seleccionaremos el ISO y clic en **Abrir**, como se ve en la *Figura 61*.

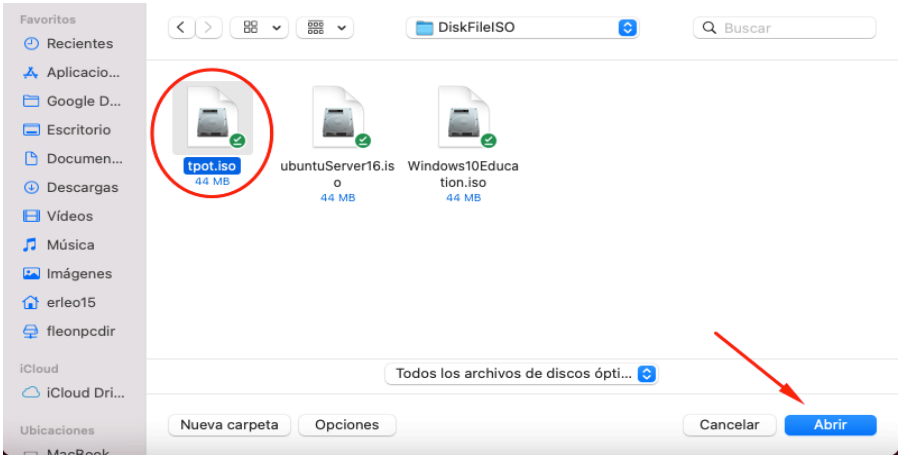


Figura 61. Selección de la ISO TPOT

5. Seleccionamos el ISO agregado y damos clic en **"Start".**

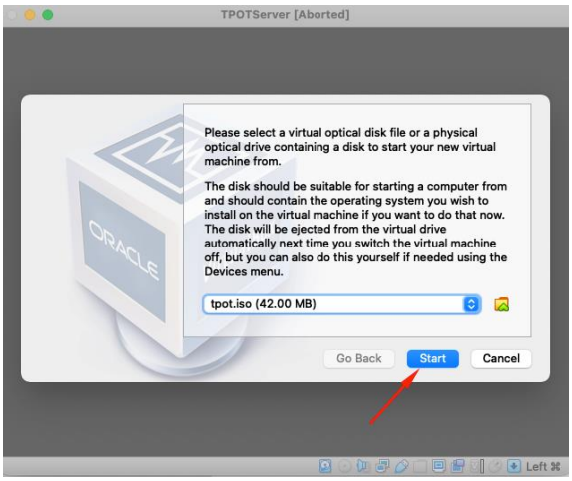


Figura 62. Inicialización de la ISO con el sistema TPOT

Con este procedimiento, se ha explicado los pasos para la configuración de una nueva máquina virtual para la instalación de su SO. Luego de finalizar la instalación del SO, se podrá apagar, guardar el estado de la máquina para luego volver a encenderla como se explicó en el paso 1 de esta sección.

ANEXO 7: Instalación de servidor TPOT

1. Para instalar el servidor multi-honeypot iniciamos la máquina virtual preparada con anterioridad (es necesario que la máquina virtual disponga de una conexión a internet) con el ISO de TPOT y nos aparecerá la siguiente ventana en donde seleccionamos "**Advanced Options**", como se ejemplifica en la *Figura 63*.

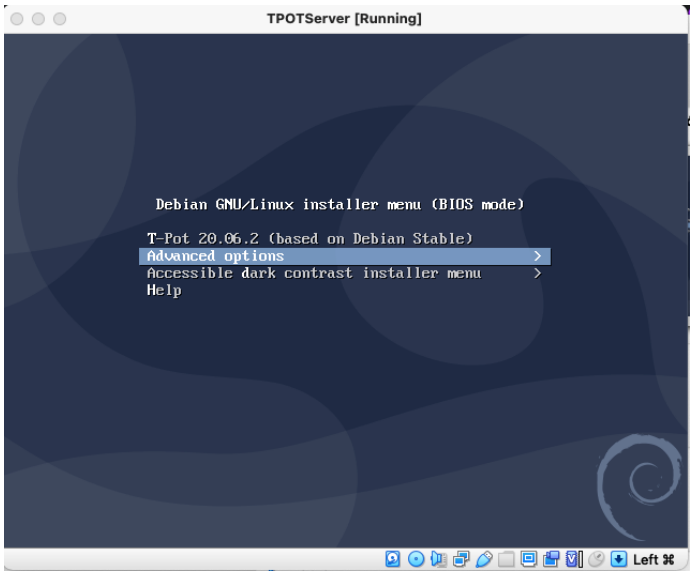


Figura 63. Ventana inicial al arrancar la ISO de TPOT

2. Luego seleccionamos la opción "**Automated Install**" y se iniciara con la instalación del SO en la máquina, ejemplo en la *Figura 64.*

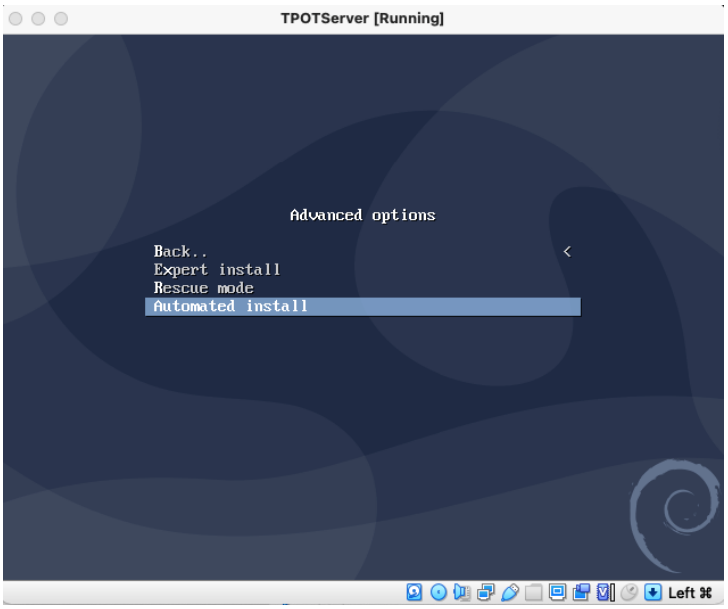


Figura 64. Opción de Instalación Automática de TPOT

3. Cuando la instalación del SO se haya finalizado la máquina se reiniciará y empezará la instalación de los componentes de TPOT como se muestra en la *Figura 65*, *Figura 66* para lo cual nos pedirá una contraseña para el usuario por defecto de la maquina "tsec" como se observa en la Figura 67. También necesitaremos un usuario web (*Figura 68*, *Figura 69*) el cual no tiene relación alguna con el usuario por defecto de la máquina.

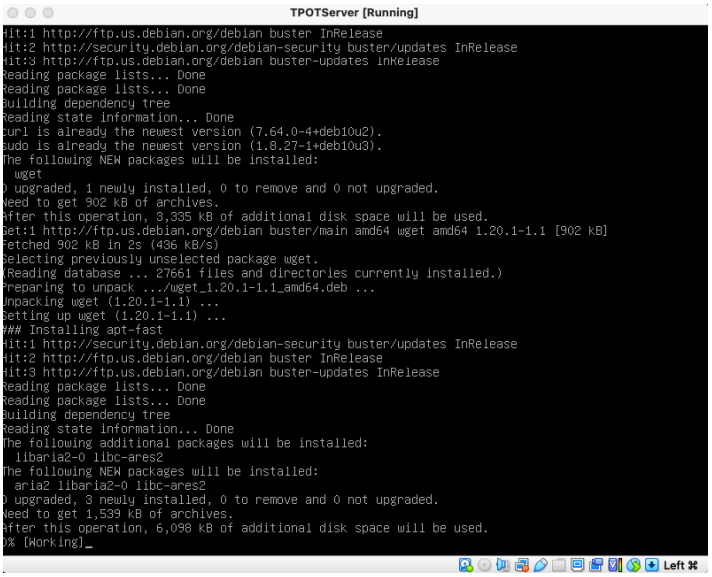


Figura 65. Instalación de los componentes TPOT

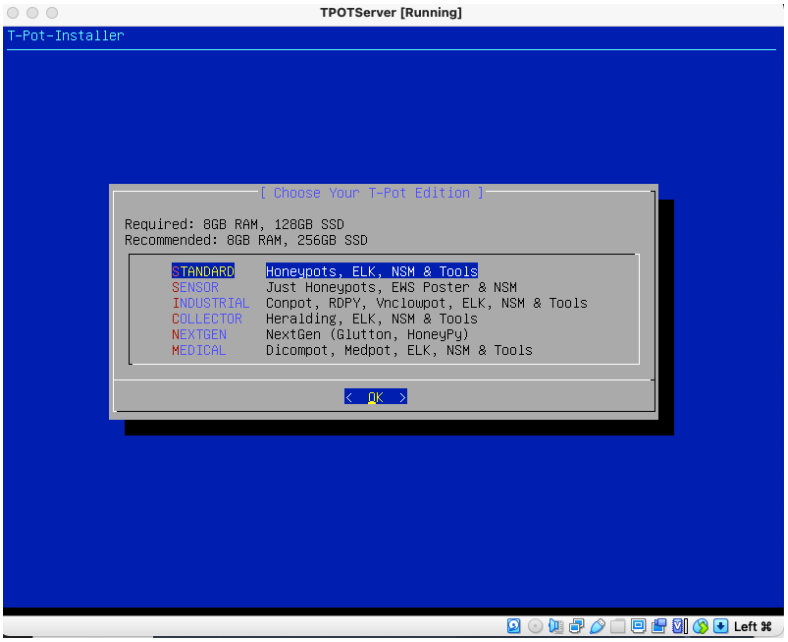


Figura 66. Opciones para el tipo de instalación de TPOT

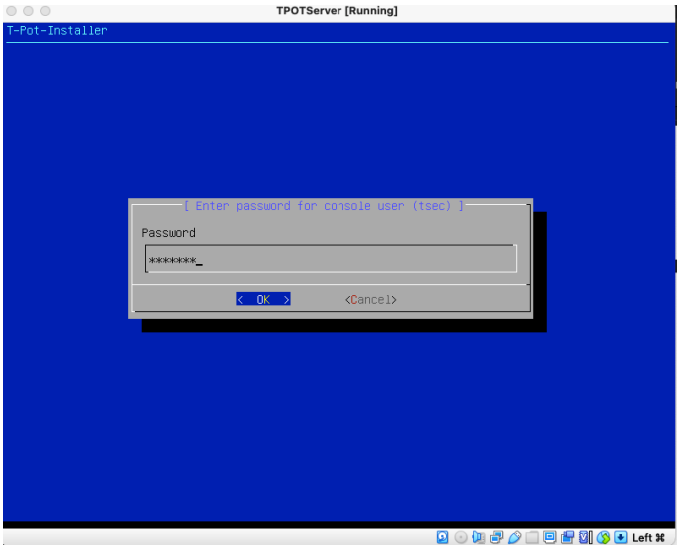


Figura 67. Contraseña del usuario principal

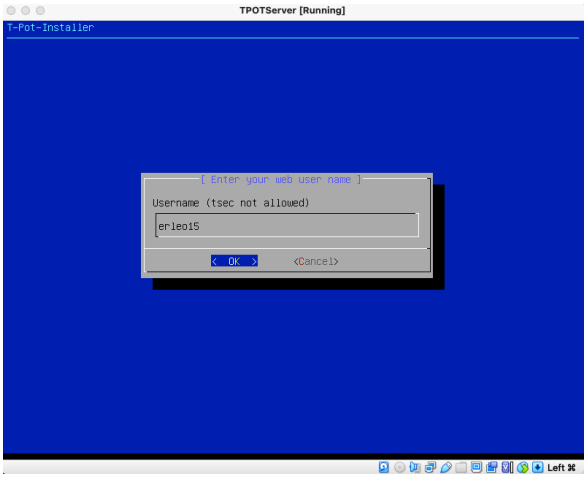


Figura 68. Nombre de usuario WEB

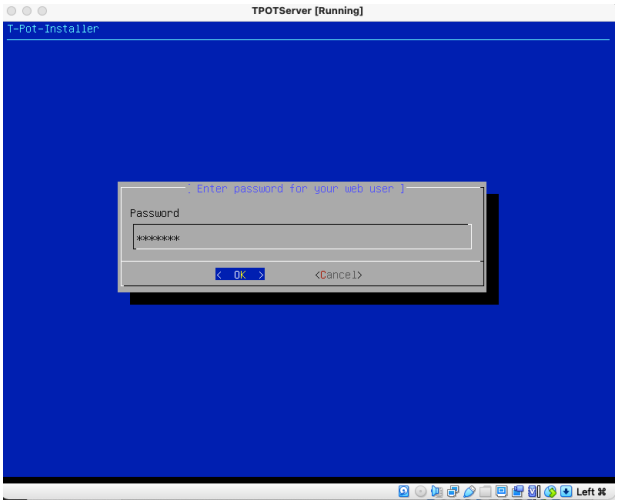


Figura 69. Contraseña para el usuario WEB

4. Cuando se haya configurado los usuarios tanto de la maquina como de la web, iniciará la instalación de los componentes de TPOT como se observa en la *Figura 70.*

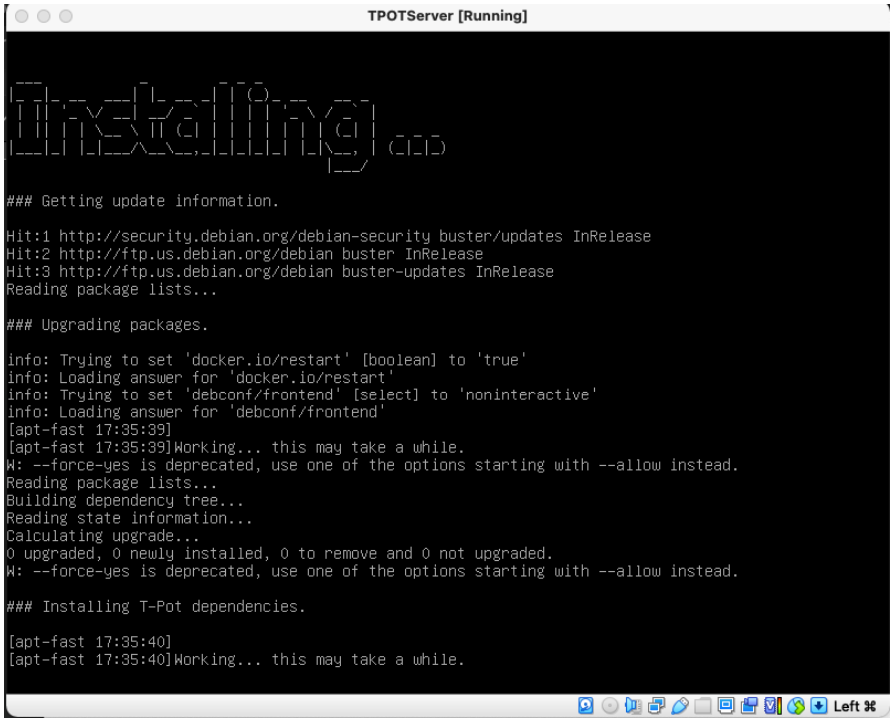


Figura 70. Instalación de los componentes de TPOT

5. Cuando el proceso de instalación haya finalizado tendremos una ventana con información similar al de la *Figura 71.*

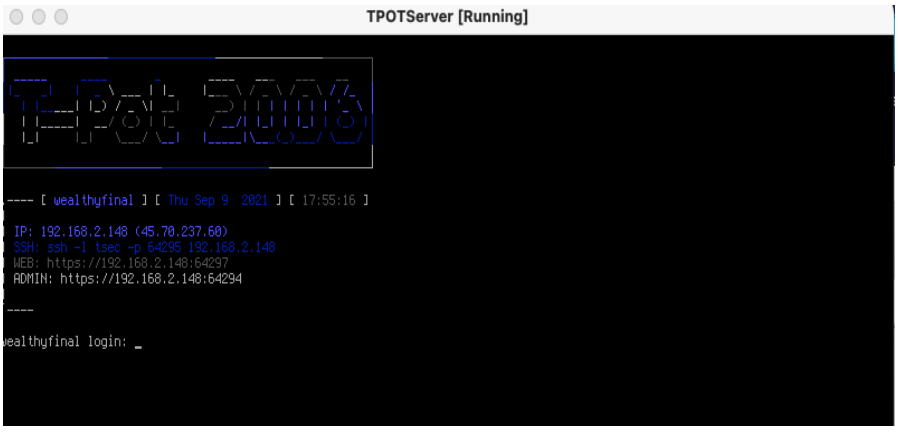


Figura 71. Ventana principal, finalizada la instalación de TPOT

Para revisar más opciones de instalación de TPOT puede visitar su sitio oficial en GitHub.com.

ANEXO 8: Configuración de TPOT y visualización de datos

Luego de instalar TPOT y sus componentes tendremos a disposición una máquina virtual con múltiples honeypot dockerizados es decir honeypots empaquetados en un binario donde se incluyen todas las dependencias de este, de forma aislada al resto de programas o aplicaciones de la máquina de manera que puedan convivir en un mismo host. Podremos revisar los siguientes aspectos básicos:

1. **Configuración de servidor Tpot**
    a. Debemos de abrir el tpot.yml ubicado en el directorio /opt/tpot/etc
    b. Se modificó el contenido acorde a las necesidades en cada una de las secciones que corresponde a la configuración de los honeypot. Al finalizar los cambios, se guardó y reinició el servicio tpot con el comando: **"service restart tpot"**
    c. Consulta la *Figura 72.*

Figura 72. Ejemplo de configuración del servidor TPOT

2. **Imágenes o paquetes en Docker**
    a. Debemos usar el comando "docker ps", para listar los paquetes activos que se tienen en docker además de su estado actual y nos aparecerá un resultado similar al de la *Figura 73.*

```
root@wealthyfinal:/home/tsec # docker ps
CONTAINER ID        IMAGE                           COMMAND                  CREATED             STATUS                    PORTS
                    NAMES
428cab9d684f        dtagdevsec/logstash:2006        "/bin/sh -c 'update.…"   About an hour ago   Up About an hour (healthy)
                    logstash
ff13ba879d0d        dtagdevsec/head:2006            "node_modules/http-s…"   About an hour ago   Up About an hour (healthy)   127.0.0.1:64302->9100/tcp
                    head
561670a5a8b6        dtagdevsec/kibana:2006          "docker-entrypoint.s…"   About an hour ago   Up About an hour (healthy)   127.0.0.1:64296->5601/tcp
                    kibana
e0122b584caf        dtagdevsec/snare:2006           "/bin/sh -c 'snare -…"   About an hour ago   Up About an hour             0.0.0.0:80->80/tcp
                    snare
a2acaf825d37        dtagdevsec/tanner:2006          "tanner"                 About an hour ago   Up About an hour
                    tanner
51bfe10b7b7a        dtagdevsec/tanner:2006          "tannerapi"              About an hour ago   Up About an hour
                    tanner_api
b2b87093edb6        dtagdevsec/nginx:2006           "/bin/sh -c '/start.…"   About an hour ago   Up About an hour
                    nginx
515d8091b022        dtagdevsec/citrixhoneypot:2006  "/bin/sh -c 'nohup /…"   About an hour ago   Up About an hour             0.0.0.0:443->443/tcp
                    citrixhoneypot
b15361a20f7d        dtagdevsec/redis:2006           "docker-entrypoint.s…"   About an hour ago   Up About an hour             6379/tcp
```

Figura 73. Resultado de paquetes activos en Docker

b. Si se requiere saber el estado especifico de un paquete se usará una combinación de dos comandos (docker y grep) de la siguiente manera "docker ps | grep -i 'nombre_paquete'", teniendo un resultado similar al de la *Figura 74*.

```
root@wealthyfinal:/opt/tpot/etc # docker ps | grep -i kibana
561670a5a8b6        dtagdevsec/kibana:2006          "docker-entrypoint.s…"   About an hour ago   Up About an hour (healthy)   127.0.0.1:64296->5601/tcp
                    kibana
root@wealthyfinal:/opt/tpot/etc #
```

Figura 74. Estado actual del paquete

3. Logs de honeypots

a. Para saber la actividad que ha realizado un honeypot usaremos el comando "docker logs 'nombre_honeypot'", el cual nos arrojara un resultado similar al de la *Figura 75*.

```
root@wealthyfinal:/home/tsec # docker logs kibana
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"canvas\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"ml\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"uptime\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"apm\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"cases\" has been disabled since the following direct or transitive dependencies are missing or disabled: [securitySolution]"}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"fleet\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"infra\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"logstash\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"osquery\" has been disabled since the following direct or transitive dependencies are missing or disabled: [fleet]"}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"security\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"securitySolution\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"spaces\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:44+00:00","tags":["info","plugins-service"],"pid":18,"message":"Plugin \"timelines\" is disabled."}
{"type":"log","@timestamp":"2021-09-09T18:09:46+00:00","tags":["warning","config","deprecation"],"pid":18,"message":"plugins.scanDirs is deprecated and is no longer used"}
{"type":"log","@timestamp":"2021-09-09T18:09:46+00:00","tags":["warning","config","deprecation"],"pid":18,"message":"kibana.defaultAppId is deprecated and will be removed in 8.0. Please use the `defaultRoute` advanced setting instead"}
{"type":"log","@timestamp":"2021-09-09T18:09:46+00:00","tags":["warning","config","deprecation"],"pid":18,"message":"Config key [monitoring.cluster_alerts.email_notifications.email_address] will be required for email notifications to work in 8.0.\""}
{"type":"log","@timestamp":"2021-09-09T18:09:46+00:00","tags":["warning","config","deprecation"],"pid":18,"message":"Disabling the security plugin (`xpack.security.enabled`) will not be supported in the next major version (8.0). To turn off security features, disable them in Elasticsearch instead."}
{"type":"log","@timestamp":"2021-09-09T18:09:46+00:00","tags":["warning","config","deprecation"],"pid":18,"message":"Disabling the spaces plugin (xpack.spaces.enabled) will not be supported in the next major version (8.0)"}
{"type":"log","@timestamp":"2021-09-09T18:09:49+00:00","tags":["info","plugins-system"],"pid":18,"message":"Setting up [93] plugins: [taskManager,lists,licensing,globalSearch,globalSearchProviders,banners,eventLog,encryptedSavedObjects,code,usageCollection,xpackLegacy,telemetryCollectionManager,telemetry,telemetryCollectionXpack,kibanaUsageCollection,securityOss,share,newsfeed,mapsEms,mapsLegacy,kibanaLegacy,translations,licenseApiGuard,legacyExport,embeddable,uiActionsEnhanced,expressions,charts,esUiShared,bfetch,data,home,cloud,console,consoleExtensions,apmOss,searchprofiler,painlessLab,grokdebugger,management,advancedSettings,fileUpload,savedObjects,visualizations,visTypeVega,visTypeVislib,visTypeTimelion,features,upgradeAssistant,snapshotRestore,reporting,licenseManagement,indexManagement,remoteClusters,crossClusterReplication,indexLifecycleManagement,dataEnhanced,beatsManagement,actions,alerting,triggersActionsUi,stackAlerts,ruleRegistry,watcher,transform,ingestPipelines,visTypeTagcloud,visTypeTable,visTypeMetric,visTypeMarkdown,visTypeXy,tileMap,regionMap,savedObjectsTagging,presentationUtil,graph,timelion,dashboard,maps,dashboardMode,dashboardEnhanced,visualize,visTypeTimeseries,inputControlVis,lens,observability,monitoring,indexPatternManagement,rollup,discover,discoverEnhanced,fileDataVisualizer,savedObjectsManagement]"}
{"type":"log","@timestamp":"2021-09-09T18:09:49+00:00","tags":["info","plugins","taskManager"],"pid":18,"message":"TaskManager is identified by
```

Figura 75. Actividad realizada por el Honeypot

Cabe recalcar que de esta manera se visualizaran los valores registrados desde el ultimo encendido, es decir cuando se reinicie el servidor los valores actuales se borraran del archivo de texto y se almacenaran en archivos comprimidos.

b. Existe otra manera de saber la actividad de un honeypot, siendo esta, la más recomendada, debido a que luego de un reinicio del servidor el comando anterior no arrojara resultados de la anterior sesión. Es por esto por lo que debemos de:

   i. En el directorio "/data" se tendrán subdirectorios similares al de la *Figura 76.*

```
root@wealthyfinal:/ # cd /data
root@wealthyfinal:/data # ls
adbhoney        ddospot     emobility  hellpot    ipphoney  rdpy
ciscoasa        dicompot    endlessh   heralding  nailoney  redishoneypot
citrixhoneypot  dionaea     ews        honeypy    nedpot    spiderfoot
conpot          elasticpot  fatt       honeysap   nginx     suricata
cowrie          elk         glutton    honeytrap  p0f       tanner
```

Figura 76. Carpetas en el directorio /data

   ii. Luego se deberá entrar en una carpeta especifica, en este caso será "tanner/log" que en donde se encuentra el archivo log y lo visualizaremos con el comando "cat" como se muestra en la *Figura 77.*

```
root@wealthyfinal:/data # cd tanner/log
root@wealthyfinal:/data/tanner/log # ls
tanner_report.json
root@wealthyfinal:/data/tanner/log # cat tanner_report.json
{"method": "GET", "path": "/", "headers": {"host": "192.168.2.148", "connection":
"keep-alive", "upgrade-insecure-requests": "1", "user-agent": "Mozilla/5.0 (Macint
osh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.45
77.63 Safari/537.36 Edg/93.0.961.38", "accept": "text/html,application/xhtml+xml,a
pplication/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v
=b3;q=0.9", "accept-encoding": "gzip, deflate", "accept-language": "es-419,es;q=0.
9,en-US;q=0.8,en;q=0.7"}, "uuid": "c168a290-4ca4-4b7d-854c-41ba7af92e01", "peer":
{"ip": "192.168.2.145", "port": 51852}, "status": 200, "cookies": {"sess_uuid": nu
ll}, "response_msg": {"version": "0.6.0", "response": {"message": {"detection": {"
name": "index", "order": 1, "type": 1, "version": "0.6.0"}, "sess_uuid": "a95538c2
-ffaa-476e-8cde-1b3171a67a8f"}}}, "timestamp": "2021-09-09T18:35:57.582447"}
{"method": "GET", "path": "/bitnami.css", "headers": {"host": "192.168.2.148", "co
nnection": "keep-alive", "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_
15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36 Edg
/93.0.961.38", "accept": "text/css,*/*;q=0.1", "referer": "http://192.168.2.148/",
 "accept-encoding": "gzip, deflate", "accept-language": "es-419,es;q=0.9,en-US;q=0
.8,en;q=0.7", "cookie": "sess_uuid=a95538c2-ffaa-476e-8cde-1b3171a67a8f"}, "uuid":
 "c168a290-4ca4-4b7d-854c-41ba7af92e01", "peer": {"ip": "192.168.2.145", "port": 5
1852}, "status": 200, "cookies": {"sess_uuid": "a95538c2-ffaa-476e-8cde-1b3171a67a
8f"}, "response_msg": {"version": "0.6.0", "response": {"message": {"detection": {
"name": "index", "order": 1, "type": 1, "version": "0.6.0"}, "sess_uuid": "237244f
9-4fa7-4f04-b931-9ca52b969dfd"}}}, "timestamp": "2021-09-09T18:35:58.527975"}
{"method": "GET", "path": "/img/tomcat.png", "headers": {"host": "192.168.2.148",
"connection": "keep-alive", "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X
```

Figura 77. Contenido del archivo tanner/log

4. Visualización de datos en una interfaz Web (Dashboard)

Para la visualización de los datos recolectados se tiene un módulo implementado en TPOT llamado Kibana, el cual nos permite la visualización en gráficas, de los registros que tienen cada uno de los honeypot de manera separada o todo en un dashboard, para lo cual debemos de:

a. Usar un navegador web en donde colocaremos la siguiente dirección url "https://ip_servidor:64297/" con su puerto por defecto 64297, donde nos pedirá el nombre de usuario web y contraseña configurados en la instalación de la máquina virtual) y nos mostrará la ventana (*Figura 78*) y daremos clic en Kibana.

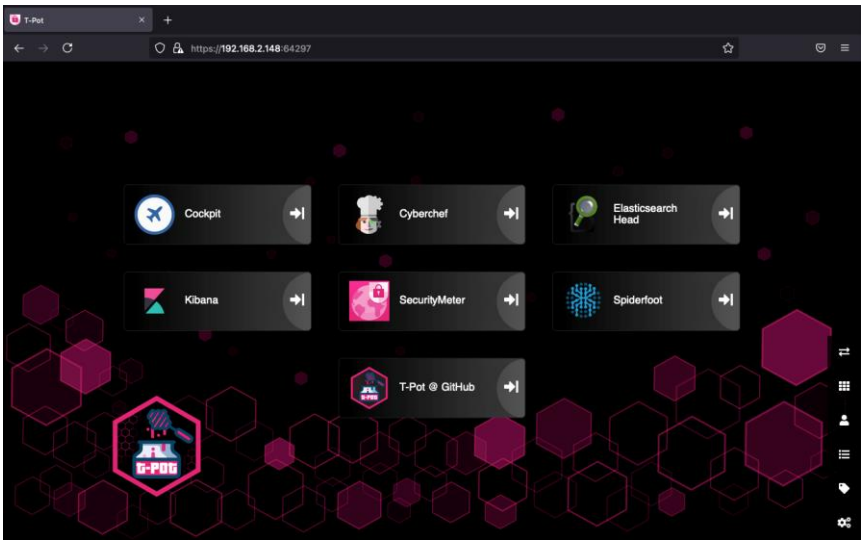


Figura 78. Pagina WEB de administración del servidor TPOT

b. Luego tendremos una ventana similar a la *Figura 79*, en donde se podrá visualizar las estadísticas de cada uno de los honeypot o un resumen general de todos los honeypot en el apartado “>T-Pot”.

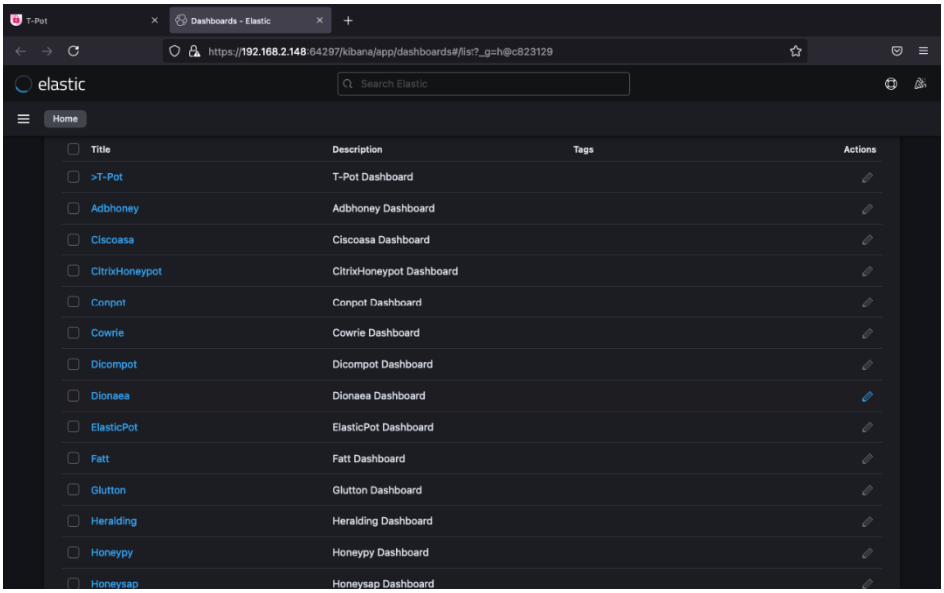


Figura 79. Resumen general de los Honeypot

5. Administrar remotamente el servidor TPOT

Para la administración de TPOT se tiene un módulo llamado Cockpit que nos permitirá administrar de forma remota el servidor, además de otras opciones. Para acceder al módulo de cockpit debemos de:

a. Ingresar a la opción “Cockpit” presente en la *Figura 80*. donde nos redirigirá a la ventana de inicio de sesión o directamente desde el navegador web se accede a la siguiente url “https://ip_servidor:64294/” con su puerto por defecto 64294.

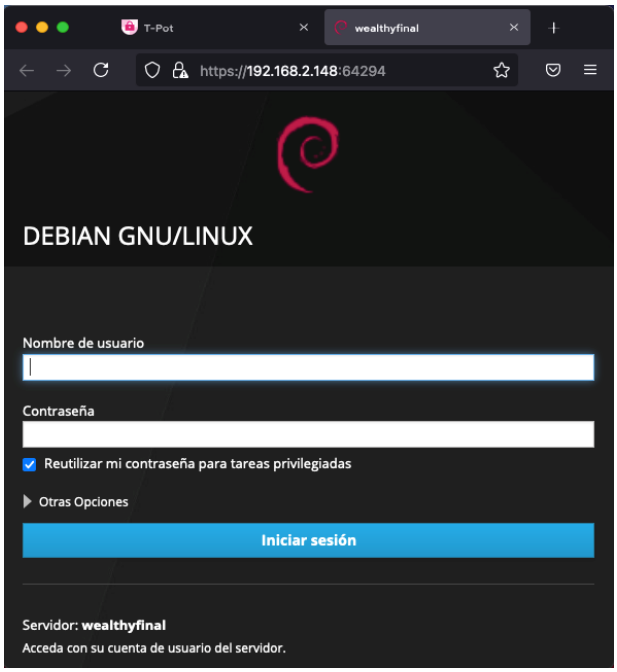


Figura 80. Autenticación del usuario en Cockpit

En esta ventana ingresaremos el nombre de usuario y contraseña del servidor (por defecto tsec) y activaremos (de ser necesario) la casilla para reutilizar la contraseña y acceder en modo root al servidor.

b. Luego de iniciar sesión correctamente nos aparecerá una página similar a la *Figura 81*, en donde nos muestra las características y estado del servidor, así como la posibilidad de revisar las actividades que ha realizado el servidor, configuraciones de red, entre otras. Además, nos da la posibilidad de interactuar con el servidor desde una consola "Terminal".

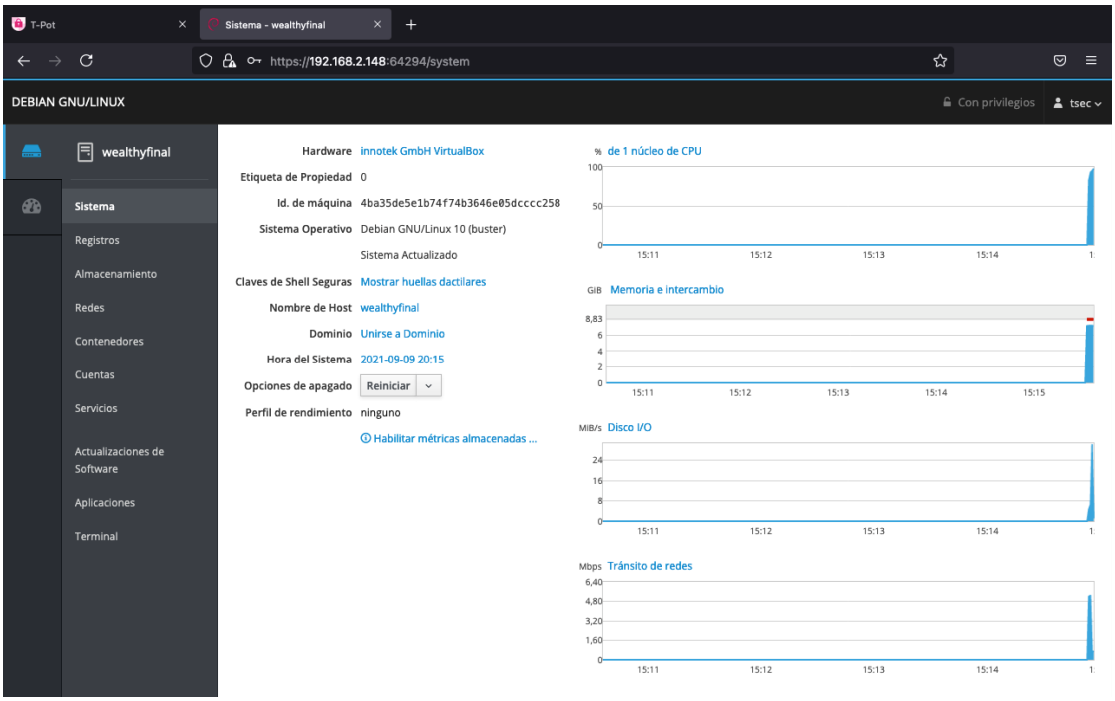


Figura 81. Características y estado actual del servidor

Gracias a este módulo es posible acceder y administrar fácilmente el servidor TPOT de manera remota, debido a que la información que presenta es muy intuitiva. Cabe recalcar que una configuración errónea o manipulación incorrecta pueden ocasionar graves daños en la máquina, debido a que se está actuando en modo root (de ser el caso) directamente sobre el sistema de archivos raíz del servidor.

ANEXO 9: Importación de ROUTER OS en GNS3

Para realizar la simulación propuesta se deberán importa todas las máquinas virtualizadas incluidos ROUTER Mickrotic, GNS3 VM, switch, etc.

Para ello se deberá seguir los siguientes pasos:

1. Abrir GNS3
2. Hacer clik en el panel de **Edit ->Preferences ->GNS3 VM,** como se observa en la *Figura 82.*
    - *Habilitamos en **Enable the GNS3 VM***
    - *Seleccionamos **VirtualBox***
    - *En VM name seleccionamos **GNS3 VM***
    - *Finalmente, clic en **ok***

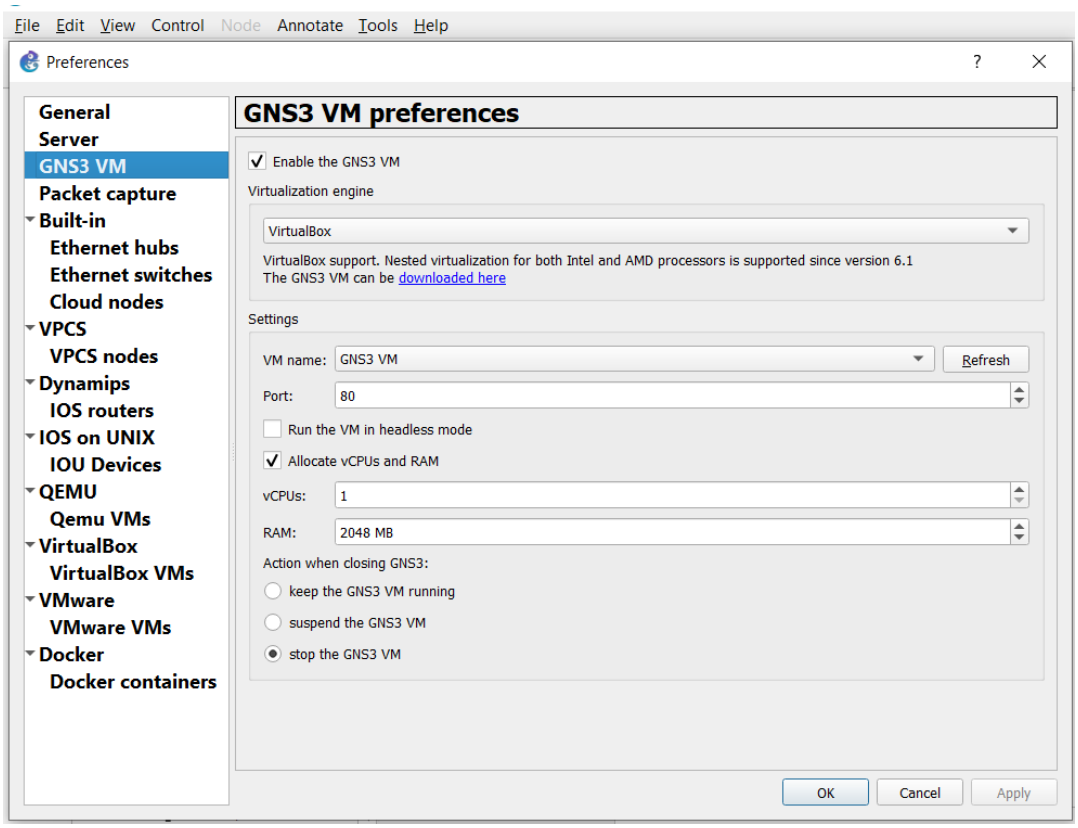


Figura 82. Importación de GNS3 VM

3. Ahora se importará RouterMikrotik una vez descargado en el siguiente enlace https://mikrotik.com/download
4. Hacer clic en el panel de **Edit ->Preferences ->QEMU,** como se observa en la *Figura 83.*

    - Precionamos en **New**
    - Se abrirá una ventana y marcamos **Run this Qemu VM on my local computer.**

- Hacemos clic en **Next**

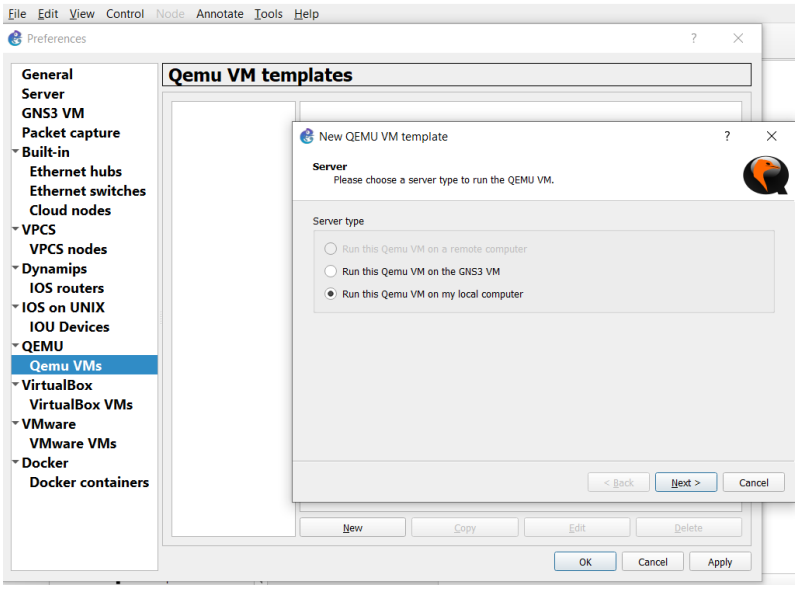


Figura 83. Importación de Router Mikrotik.

5. Una vez realizado el paso 4, nos saldrá una nueva ventana como se muestra en la *Figura 84*, le damos un nombre al Router que se descargó previamente y presionamos **next.**

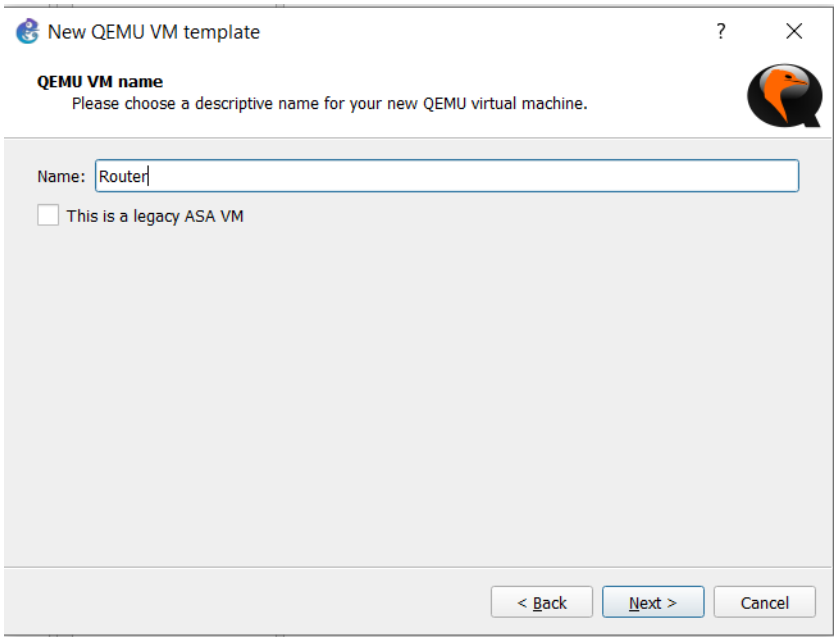


Figura 84. Asignación de un nombre al router

6. Dejamos los valores por defecto y damos clic en **next,** Figura *85*.

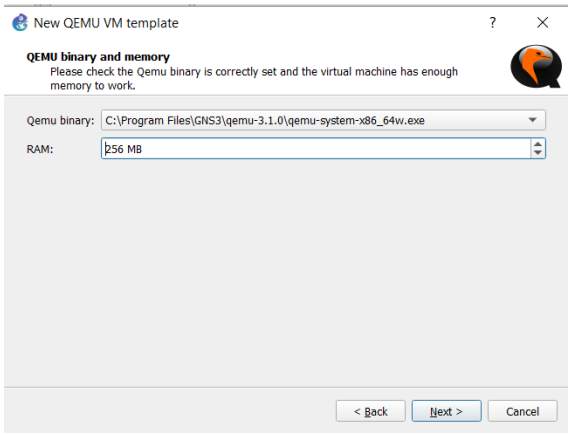


Figura 85. Ubicación de QemuSystem y asignación de RAM

7. Ahora seleccionamos la opción **Telnet,** damos clic en **next** como se observa en la *Figura 86.*

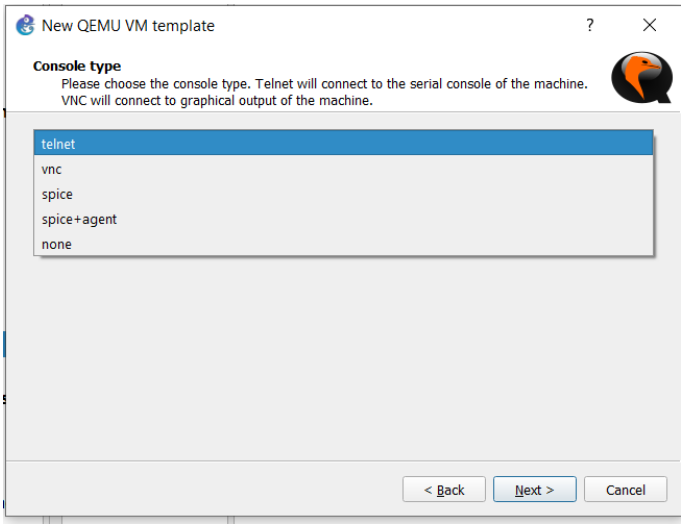


Figura 86. Tipos de consolas para la plantilla

8. Una vez realizado el paso anterior, en la *Figura 87 nos* saldrá una ventana en la cual seleccionamos la opción **New Image** y en **Browse** buscamos la ruta donde ese encuentra el Router Mikrotik descargado, realizado esto damos clic en **Finish.**

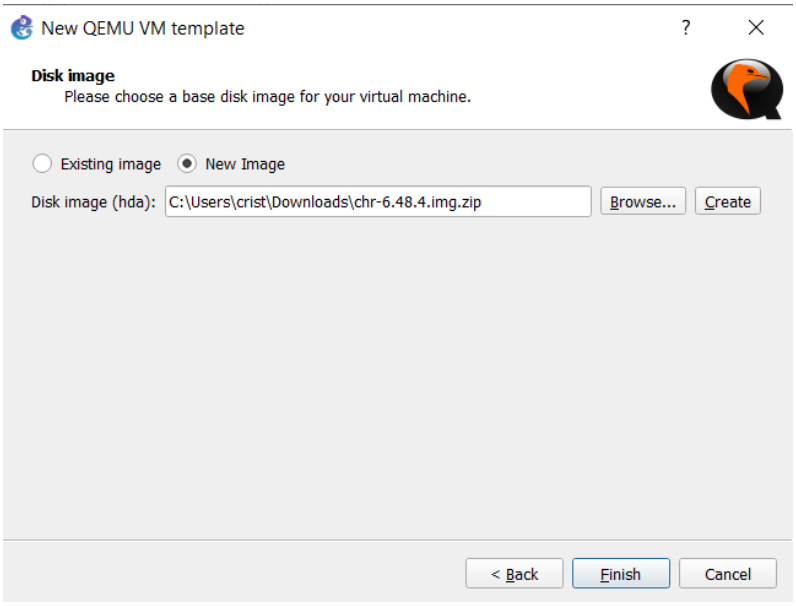


Figura 87. Ruta de ubicación del router

9. Finalmente, importado correctamente nos mostrara datos generales, hacemos clic en **Apply** y **Ok,** como esta en la *Figura 88.*

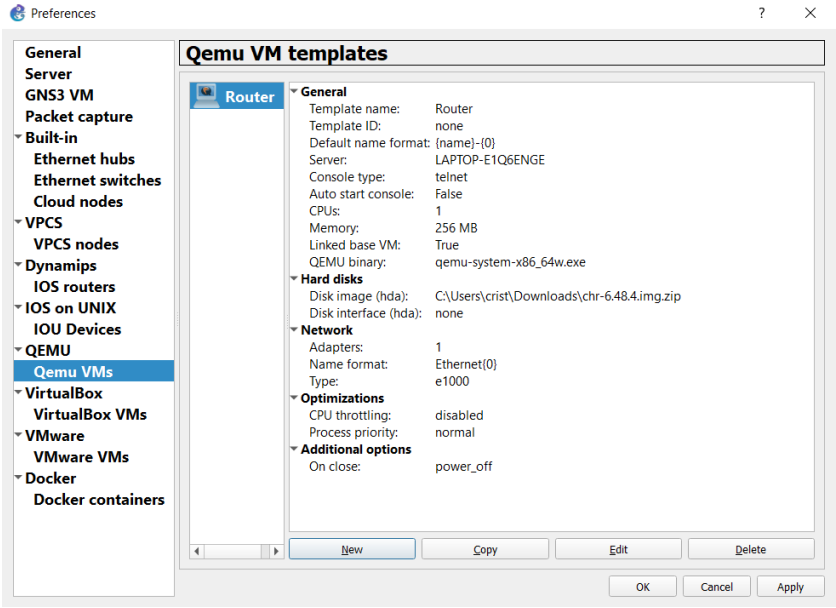


Figura 88. Datos Generales de router

10. Hacemos clic en **All device,** clic derecho y damos clic en **Configure template,** ejemplo en la *Figura 89.*

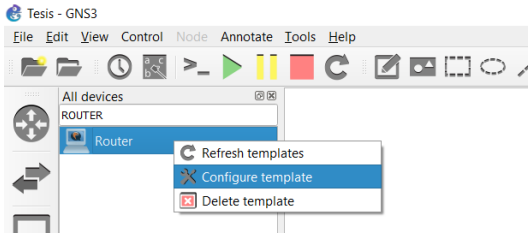


Figura 89. Representación del router importado

11. En la *Figura 90* se indica como elegir la simbología que represente a un router, de esta manera tendremos mayor orden y se evitara confusiones. Primero vamos a ubicarnos en la pestaña **General setting ->**simbología ->**Browse.**
    Se abrirá una nueva ventana en la cual presionamos en >**Classic** y seleccionamos **router_irewall** y finalmente damos **OK.**

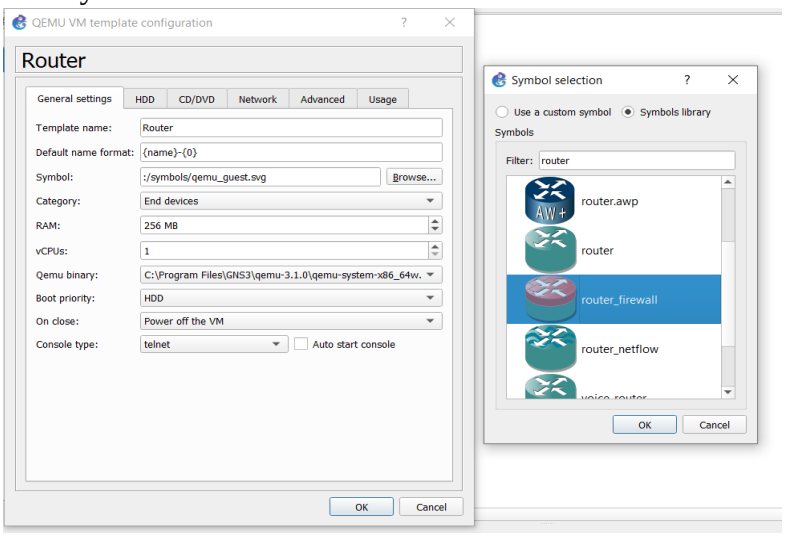


Figura 90. Asignación de simbología del router

ANEXO 10: Instalación de Ubuntu

Descargamos Ubuntu server desde el sitio oficial: http://releases.ubuntu.com/16.04/

Pasos por seguir en la instalación:

1. Elección del Idioma **Español**, observe la *Figura 91*.

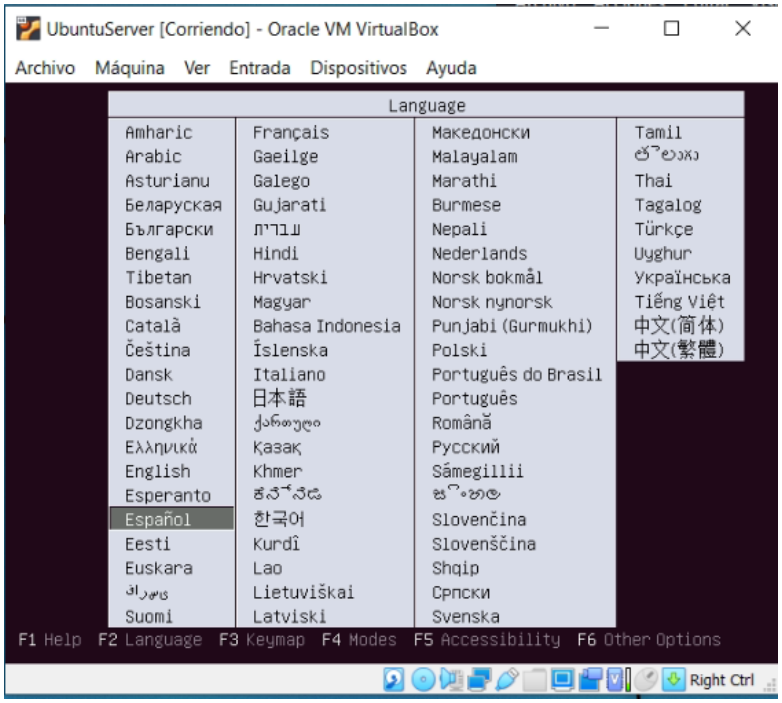


Figura 91. Idiomas disponibles para la instalacion

2. Elección del Idioma **Spanish—español**, revisa la *Figura 92*.

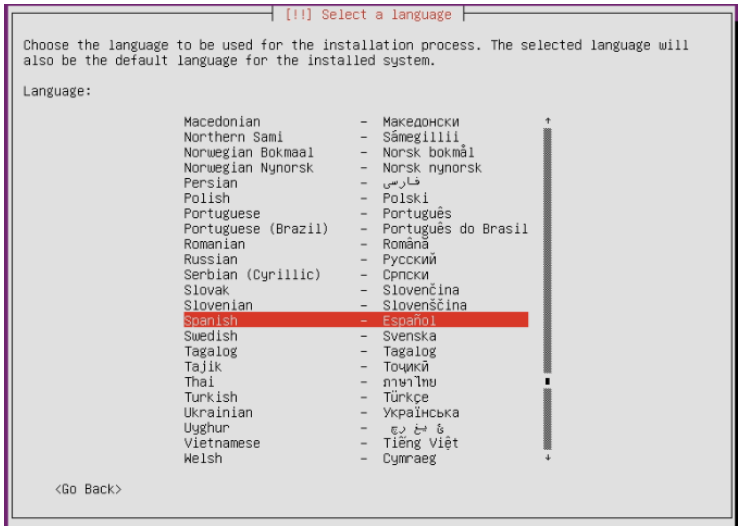


Figura 92. Selección del idioma para el proceso de instalación

3. En la *Figura 93*, se muestra la selección del país donde nos encontramos ubicados actualmente, para poder configurar más adelante la zona horaria, presionamos enter en **Ecuador.**

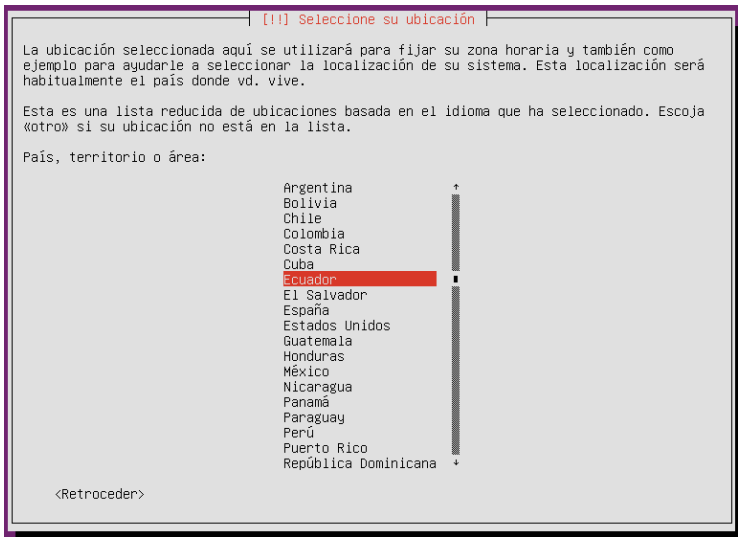


Figura 93. Selección de ubicación

4. En la *Figura 94*, se configura el teclado, realizar enter en la opción **<Si>** para detectar el teclado.

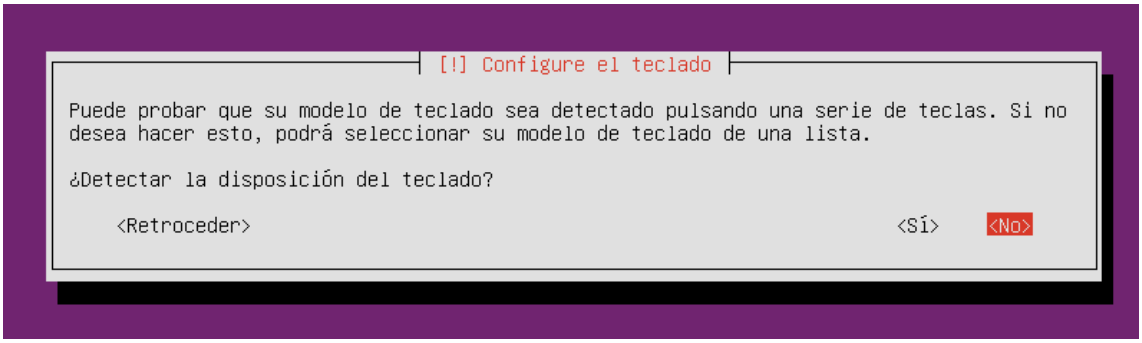


Figura 94. Configuración de teclado

5. Nos pedirá el nombre de la máquina y nombre del usuario, es recomendable anotar los datos del nombre del usuario, se detalla en la *Figura 95*, ya que más delante el sistema te pedir para poder instalar programas a futuro e ingresar de manera segura.

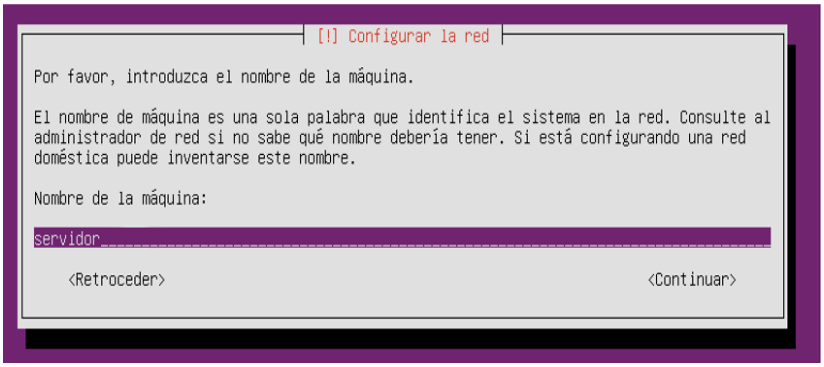


Figura 95. Configurar la red

6. En la *Figura 96.* El sistema pedirá crear una contraseña máximo escribir 8 caracteres, caso contrario las contraseñas serán rechazadas porque son fáciles de descifrar, una vez escrita la contraseña, damos enter en **<Continuar>**.

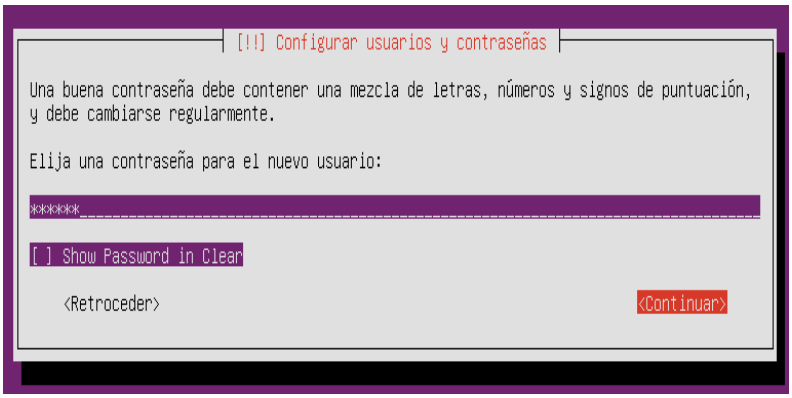


Figura 96. Configuración de Contraseña

7. Por seguridad es recomendable cifrar la carpeta personal, ya que se obtiene almacenamiento de manera privada, hacer enter en **<Si>.** Como se observa en la *Figura 97.*

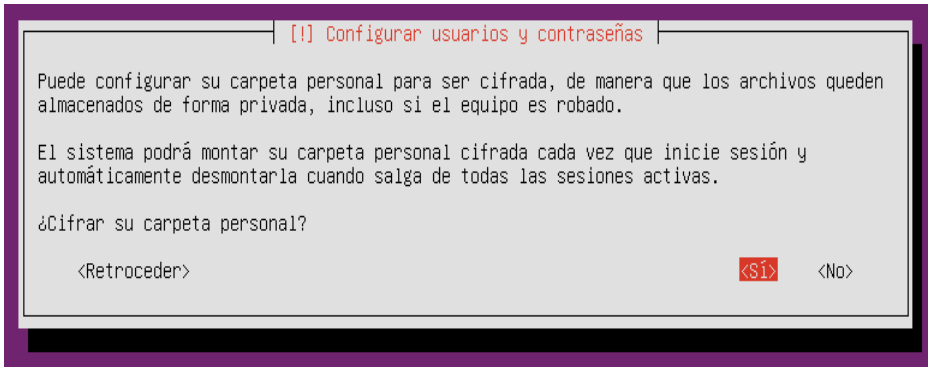


Figura 97. Cifrar carpeta personal

8. El SO, detecta la zona horaria por ello verificamos que la zona horaria (*Figura 98*) sea correspondiente a la ubicación actual, si la zona horaria es correcta, presionamos enter en **<Si>.**

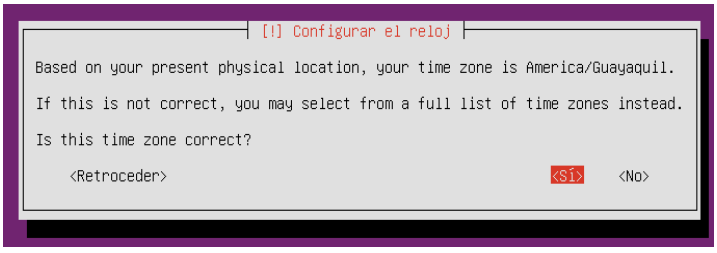


Figura 98. Configuración de reloj

9. Seleccionamos la opción **Guiado – utilizar el disco duro completo y configurar LVM** como se observa en la *Figura 99.*

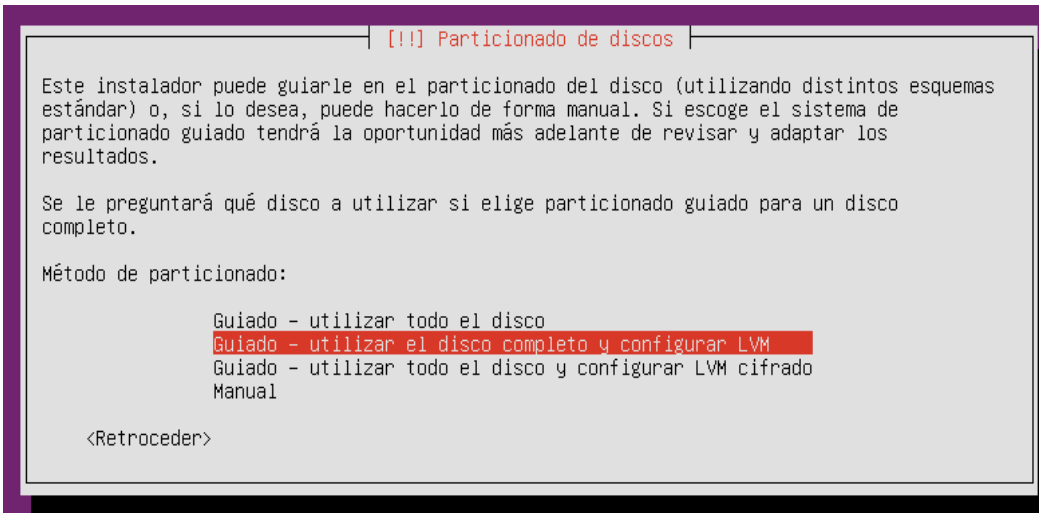


Figura 99. Partición del disco

10. Damos enter en **<Si>** para guardar los cambios realizados en la partición de discos, *Figura 100.*

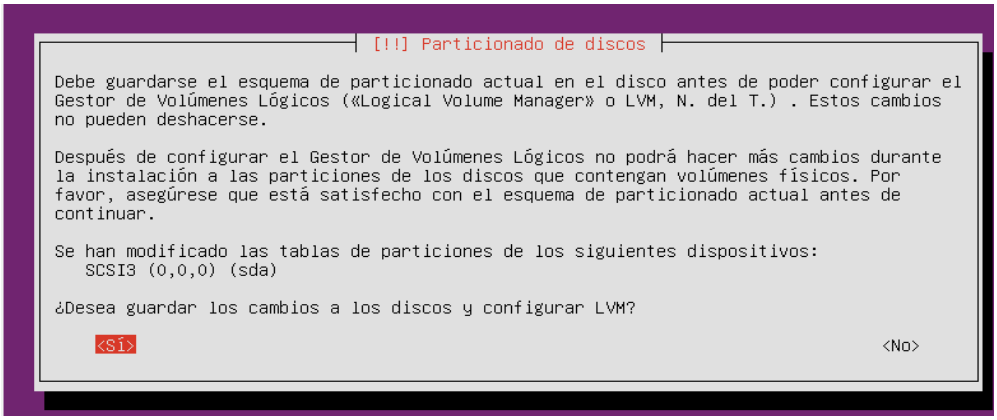


Figura 100. Aceptar cambios de partición del disco

11. Damos enter en **<Si>** para escribir los cambios en los discos, como se muestra en la *Figura 101*.

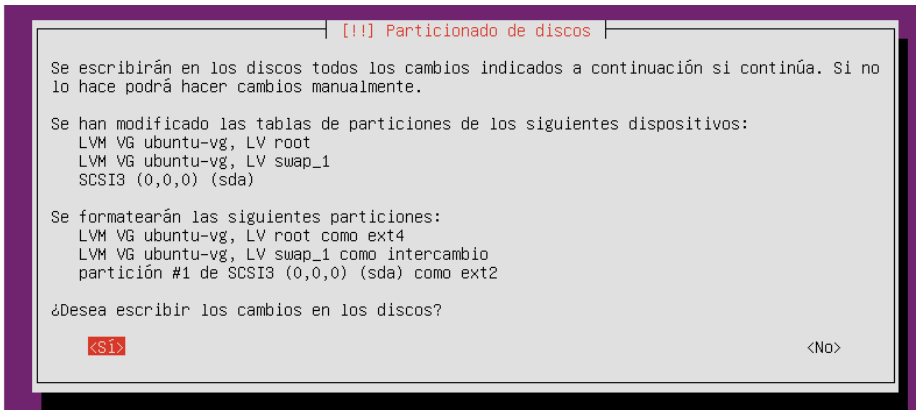


Figura 101. Confirmación de cambios en el disco

12. En este paso esperaremos que termine de instalarse el sistema de Ubuntu y una vez dejamos en blanco la sección de Proxy, debido que no es necesario. posteriormente damos enter en **<Continuar>**, revisar la *Figura 102.*

Figura 102. Gestor de Paquetes

13. Esperamos que los paquetes terminen de descargarse, no hacer ninguna acción en este punto, se mostrara una pantalla igual a la *Figura 101*, cuando termine de configurarse los paquetes se mostrara la opción **Configuración de tasksel** como en la *Figura 104*, damos enter en la opción **sin actualizaciones automáticas.**

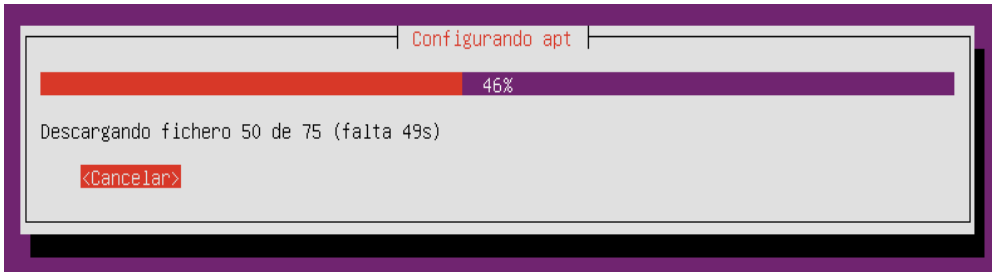


Figura 103. Descarga de ficheros

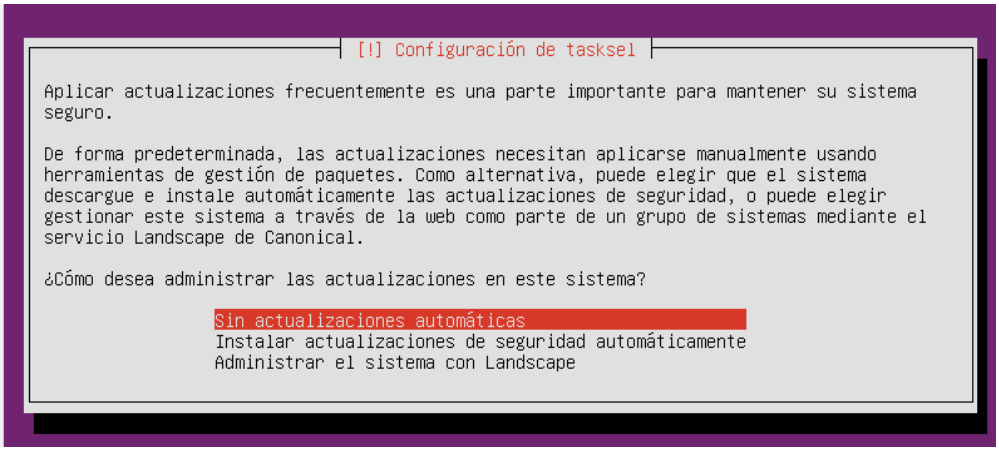


Figura 104. Administrador de actualizaciones

14. En este apartado como se observa en la *Figura 105*. Dejamos por defecto **Standard system utilities**, ya que se instalarán manualmente otros servicios. Una vez que se instale completamente Ubuntu Server.

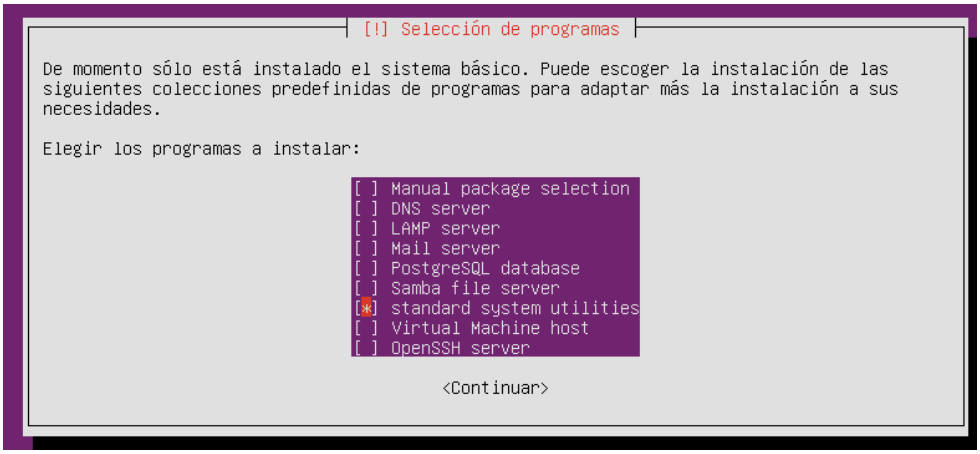


Figura 105. Selección de programas a instalarse

15. Instalamos el cargador GRUD de arranque para ello damos clic en **<Si>**, como se muestra en la *Figura 106.*

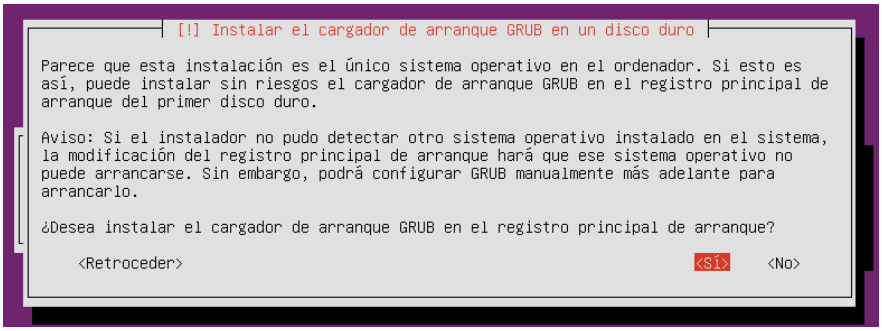


Figura 106. Instalación de GRUB de arranque

16. Damos enter en **<Continuar>** para realizar la instalación completa, como en la *Figura 107*, terminado este procedimiento, se abrirá la consola de Ubuntu, en la cual solicita nombre de usuario y contraseña para acceder a los servicios de Ubuntu y demás herramientas

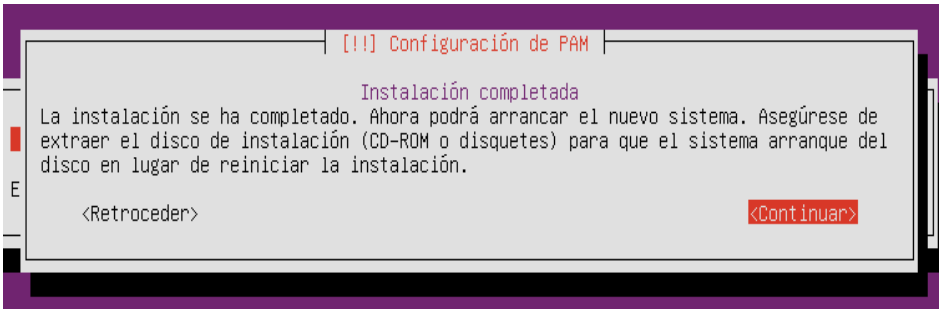


Figura 107. Configuración de PAM

ANEXO 11: Resultado de la ejecución en consola del plan de pruebas

Para la ejecución sobre el protocolo SSH (Honeypot Cowrie) se tienen los siguientes resultados.

1. Para el primer caso que consiste en 1 máquina se tiene:

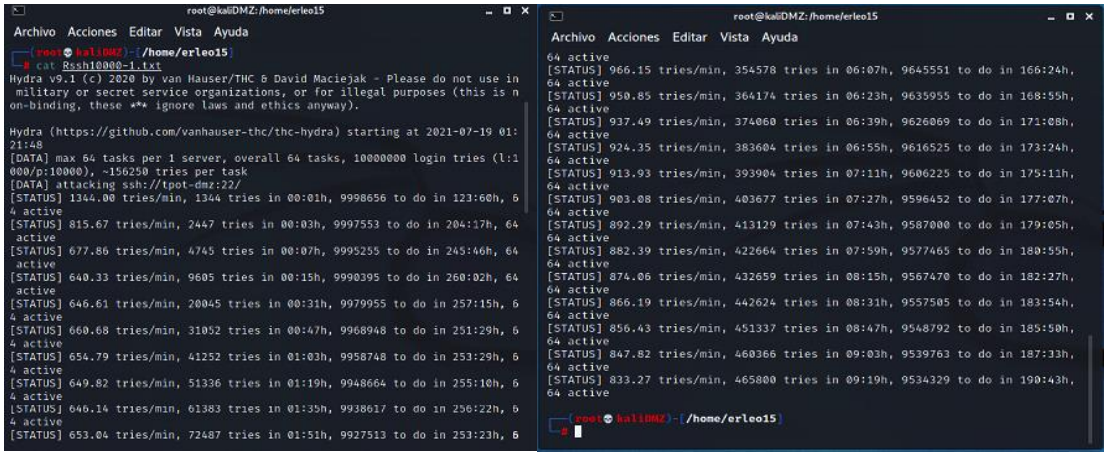


Figura 108. Resultado del ataque en SSH

2. Para el segundo caso, en donde se tienen 3 máquinas atacantes se tiene:

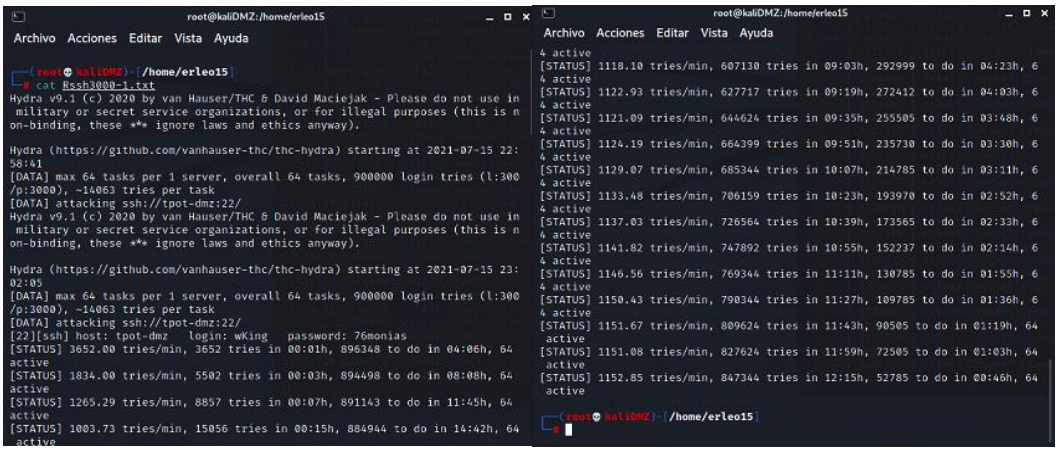


Figura 109. Máquina 1 Atacante

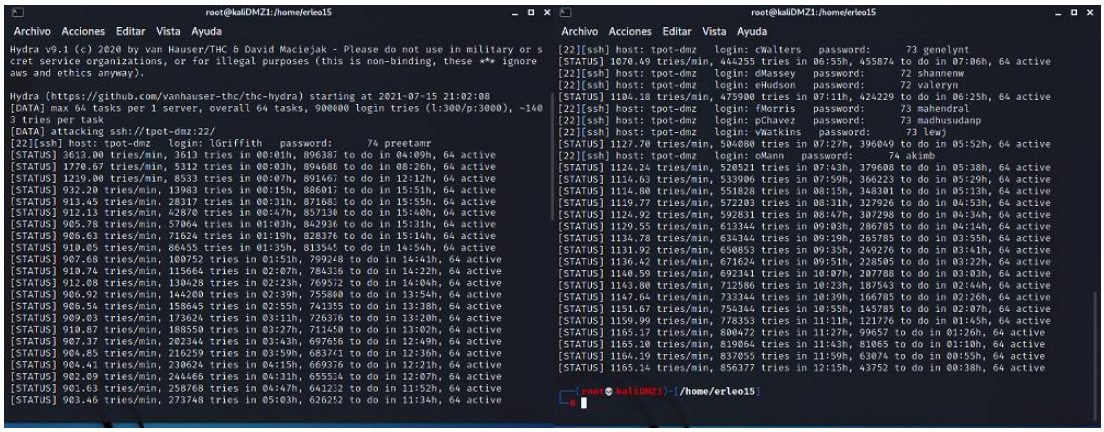


Figura 110. Máquina 2 Atacante

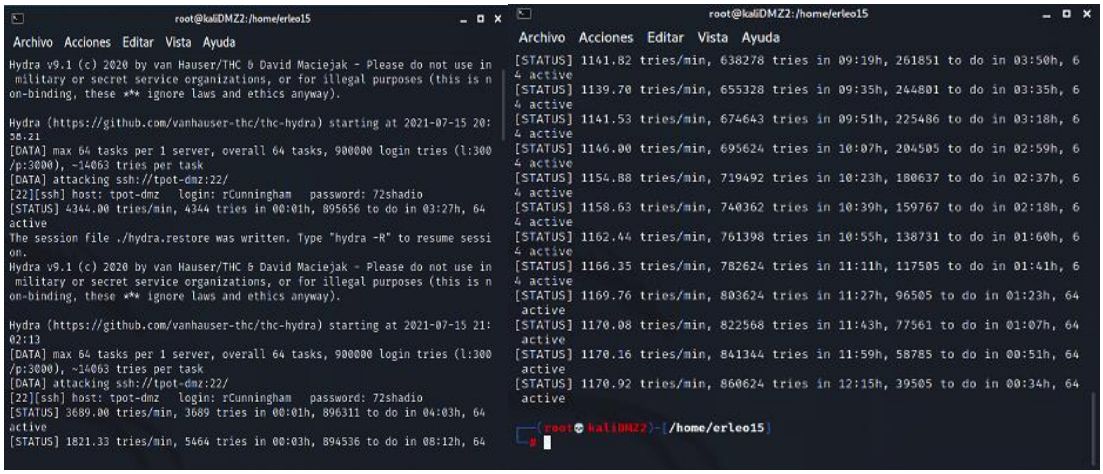


Figura 111. Máquina 3 Atacante

3. En el tercer caso, donde se tienen 5 máquinas, el resultado es:

   Maquina 1

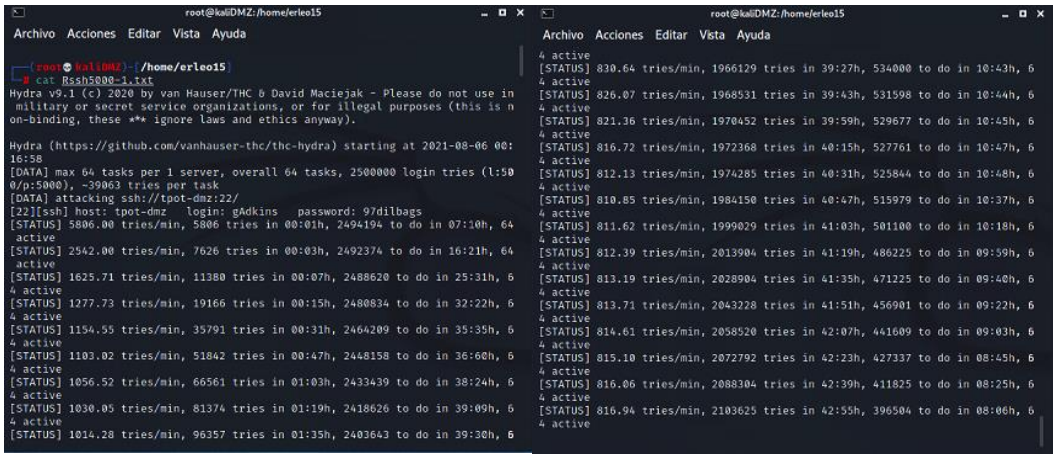


Figura 112. Máquina 1 Resultado del ataque en SSH

Máquina 2

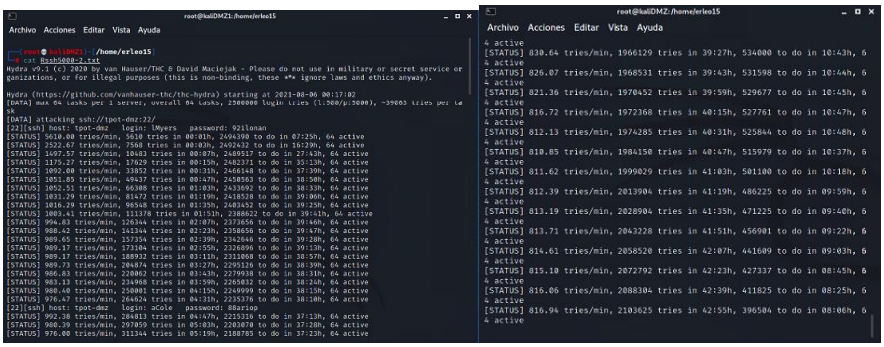


Figura 113. Máquina 2 resultado del ataque en SSH

Maquina 3

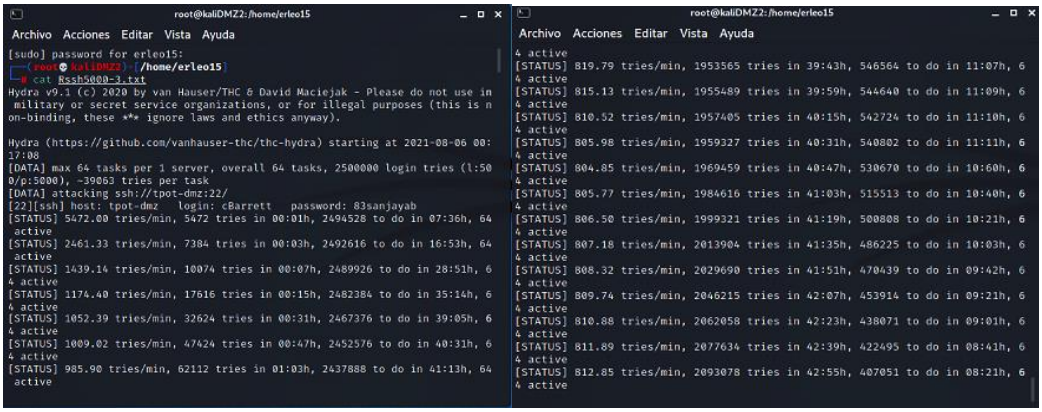


Figura 114. Máquina 3 resultado del ataque en SSH

Maquina 4

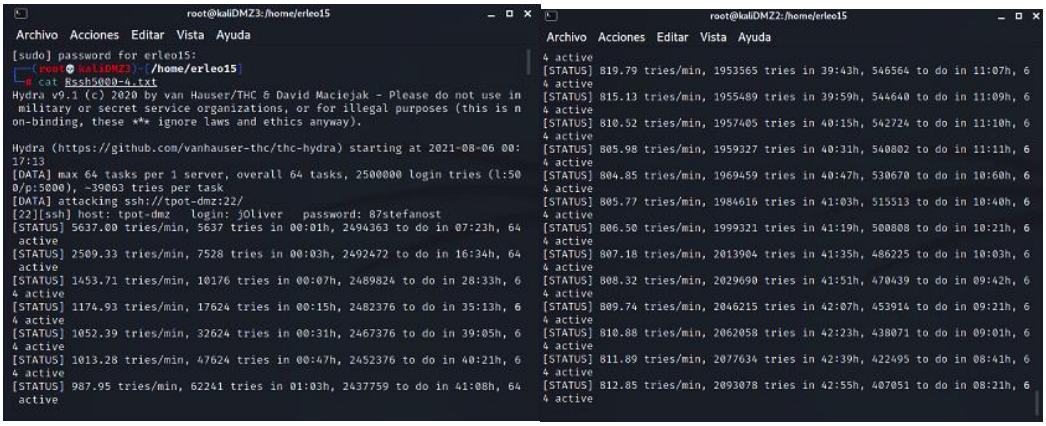


Figura 115. Máquina 4 resultado del ataque en SSH

Maquina 5

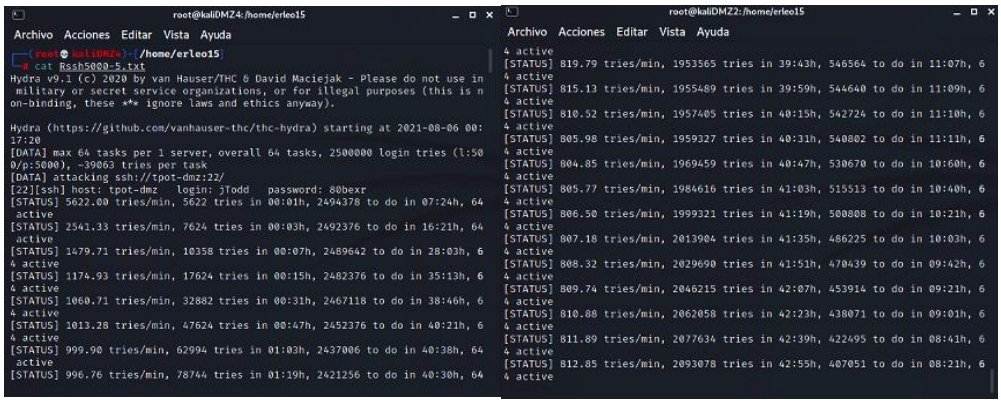


Figura 116. Máquina 5 resultado del ataque en SSH

Realización de ataques al honeypot Dionaea (FTP)

1. Para el primer caso de prueba se usó 1 máquina donde se corrió el comando correspondiente dando como resultado lo siguiente.

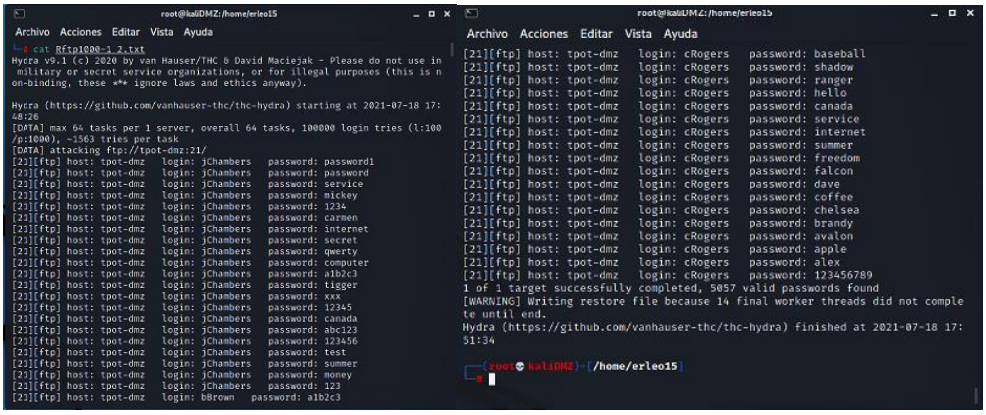


Figura 117. Ataque en FTP

2. En el segundo caso de prueba se tuvieron 3 máquinas atacantes de las cuales se obtuvo las siguientes salidas.

Máquina 1

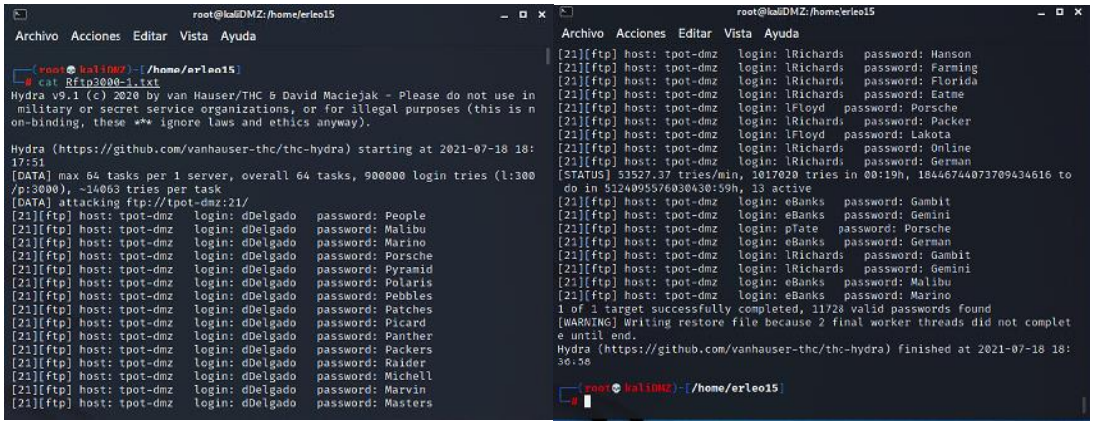


Figura 118. Máquina 1 Atacante

Máquina 2

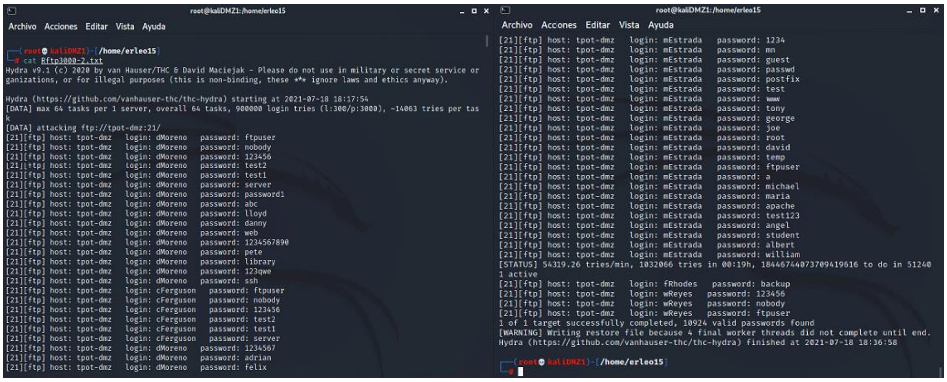


Figura 119. Máquina 2 Atacante

Máquina 3

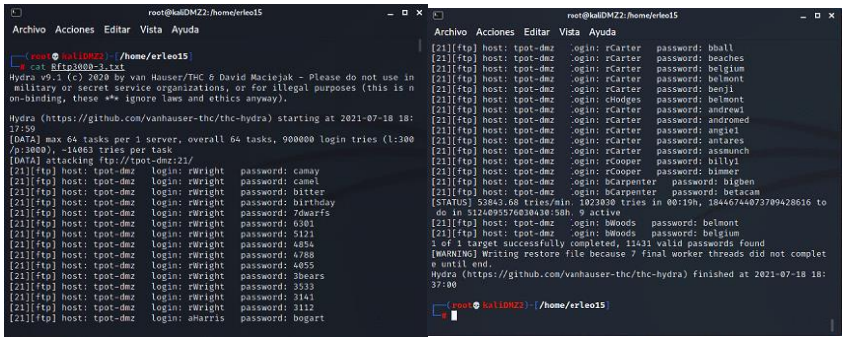


Figura 120. Máquina 3 Atacante

3. En el tercer caso tenemos 5 máquinas atacantes las cuales ejecutaron sus respectivos comandos con las siguientes salidas.

Máquina 1

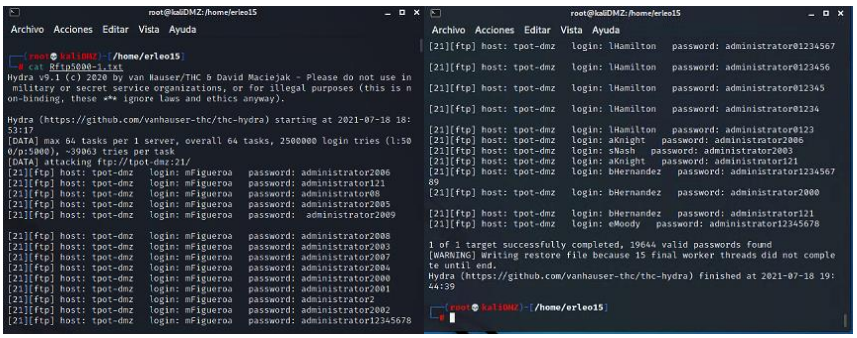


Figura 121. Máquina 1 atacante - salida de resultados

Máquina 2

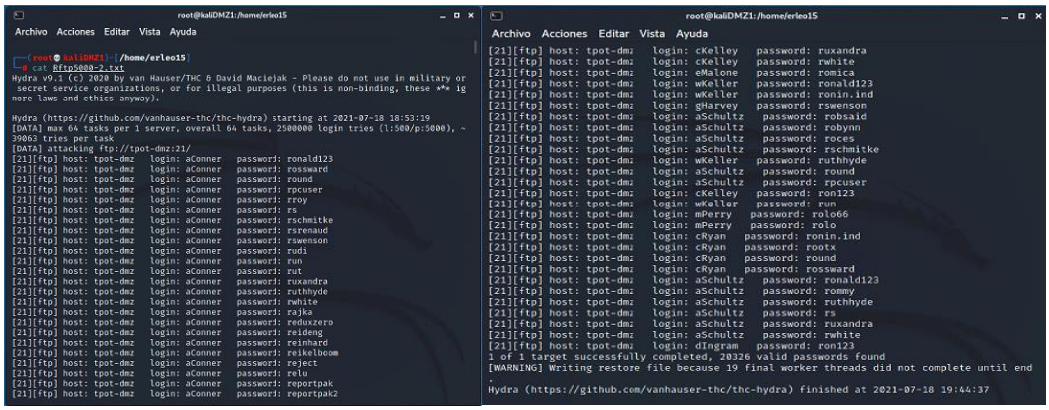


Figura 122. Máquina 2 atacante - salida de resultados

Maquina 3

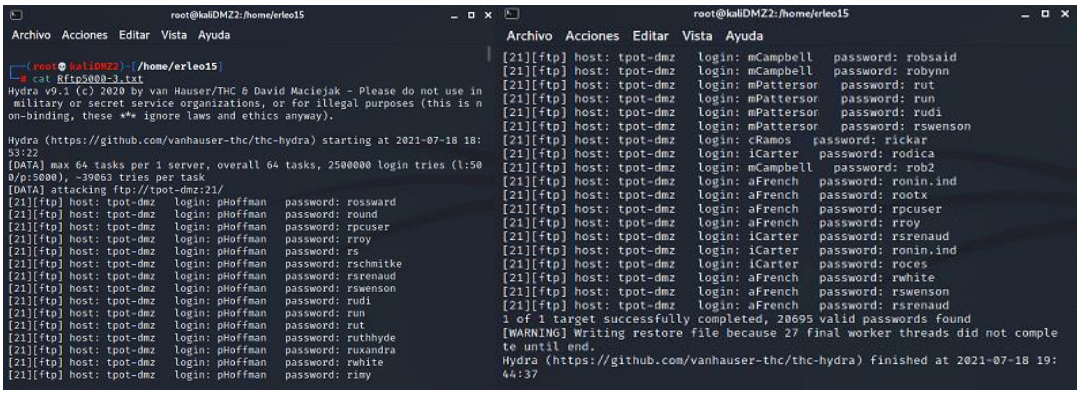


Figura 123. Máquina 3 atacante - salida de resultados

Maquina 4

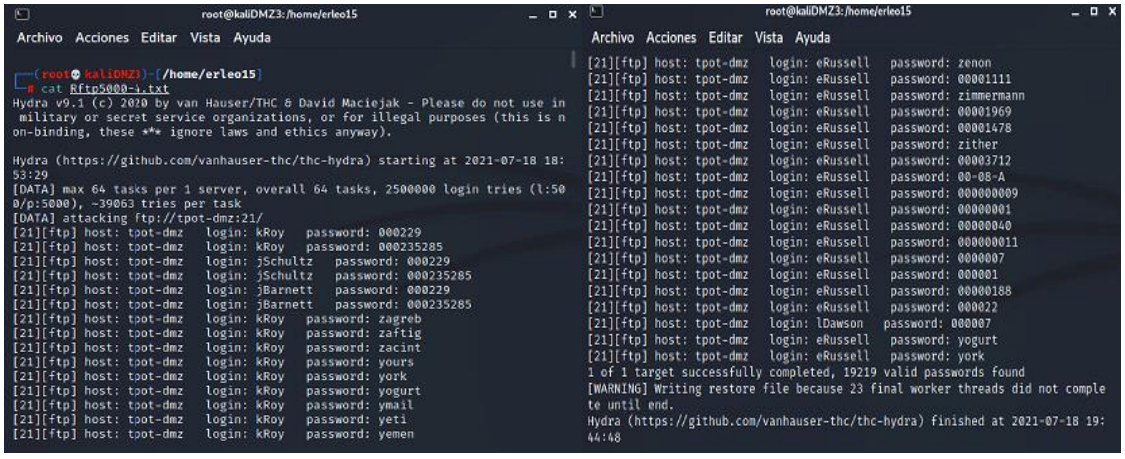


Figura 124. Máquina 4 atacante - salida de resultados

Máquina 5

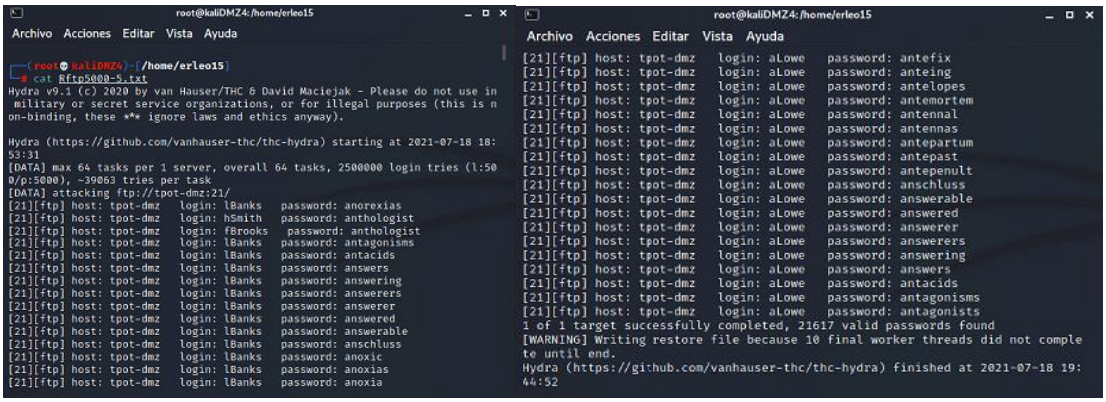


Figura 125. Máquina 5 atacante - salida de resultados

Realización de ataques al honeypot Mailoney (SMTP)

1. En el primer caso de prueba comprende una sola máquina con la siguiente salida en consola.

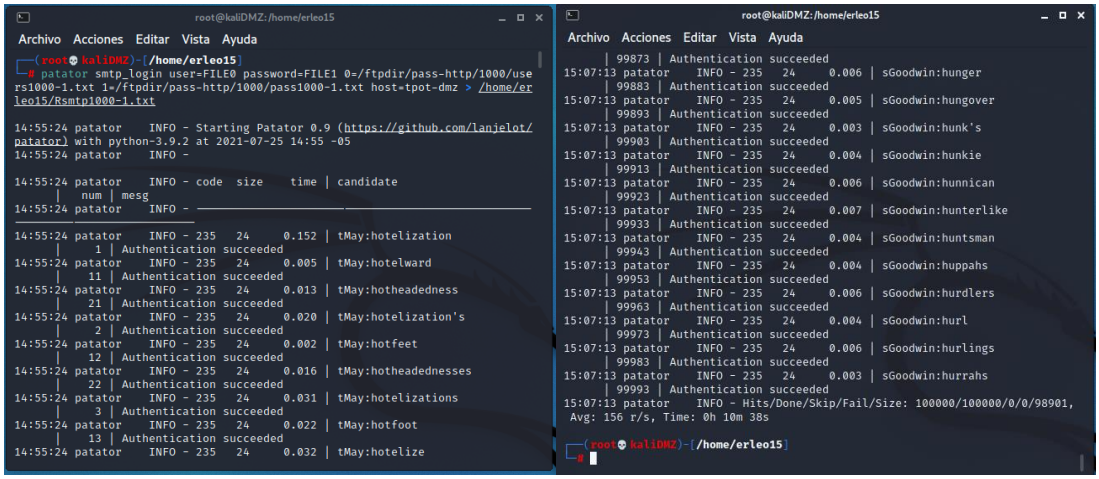


Figura 126. Ataque en SMTP

2. En el segundo caso de prueba se agregan 2 computadoras en las cuales se usan los comandos correspondientes con las siguientes salidas en consola de cada una.

Máquina 1

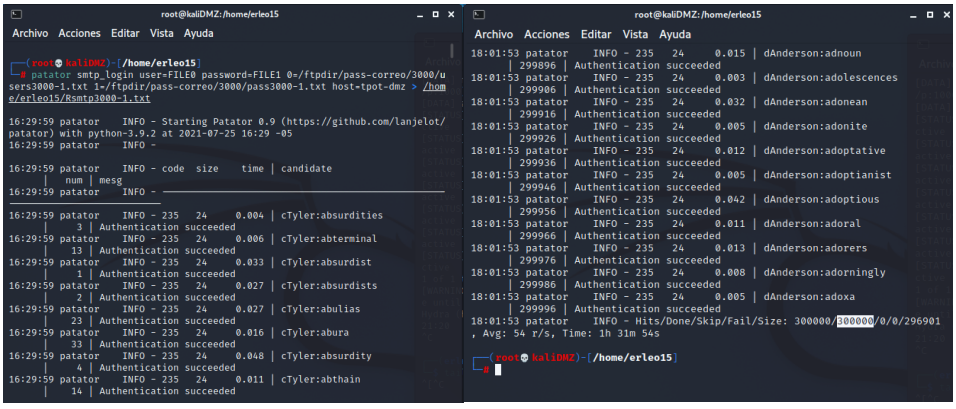


Figura 127. Máquina Resultado del ataque en SMTP

Máquina 2

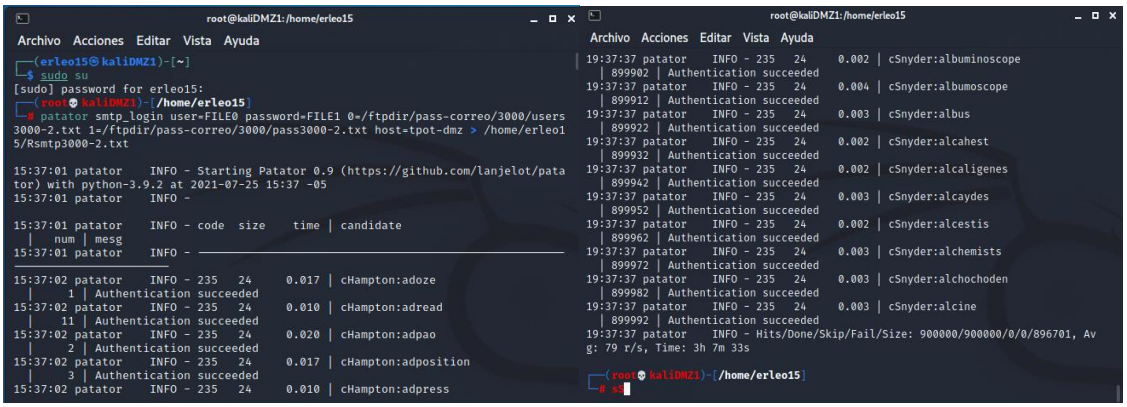


Figura 128. Máquina 2 Resultado del ataque en SMTP

Máquina 3

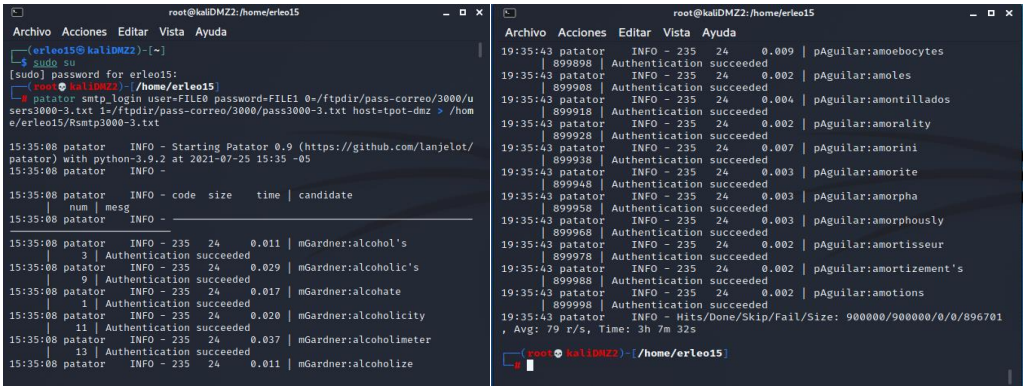


Figura 129. Máquina 3 resultado del ataque en SMTP

3. En el tercer caso de pruebas se usan 5 máquinas, donde se corrieron los comandos correspondientes y nos dieron las siguientes salidas en consola.

Máquina 1

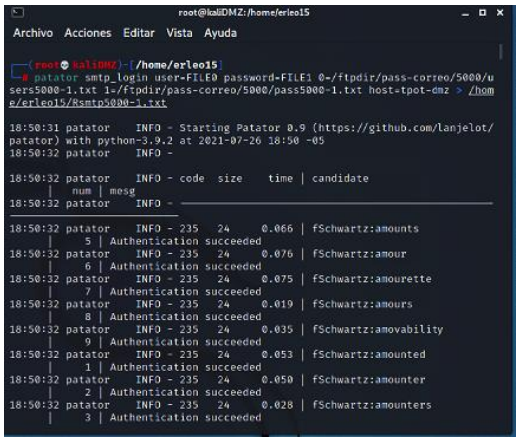


Figura 130. Resultado de la máquina 1 ataque en SMTP

Máquina 2

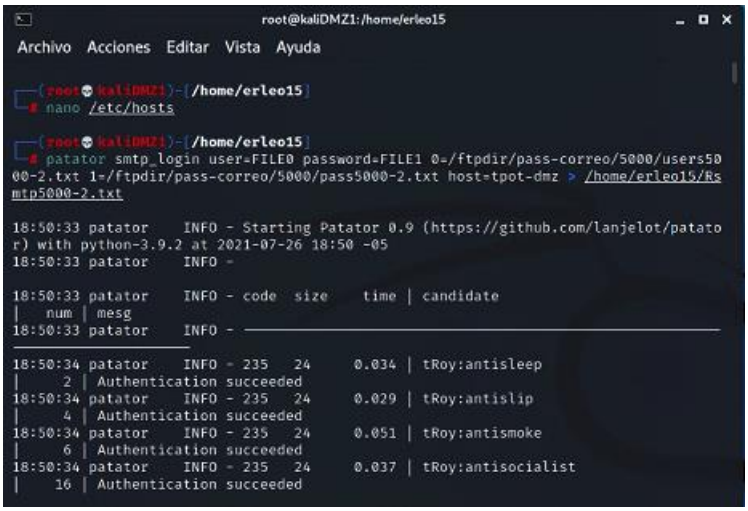


Figura 131. Resultado de la máquina 2 ataque en SMTP

Máquina 3

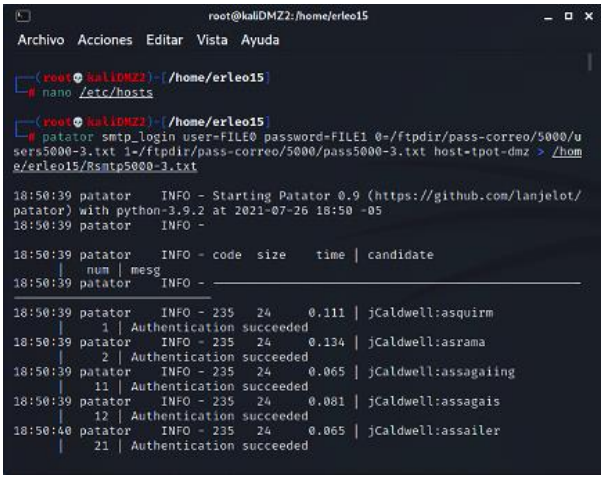


Figura 132. Resultado de la máquina 3 ataque en SMTP

Máquina 4

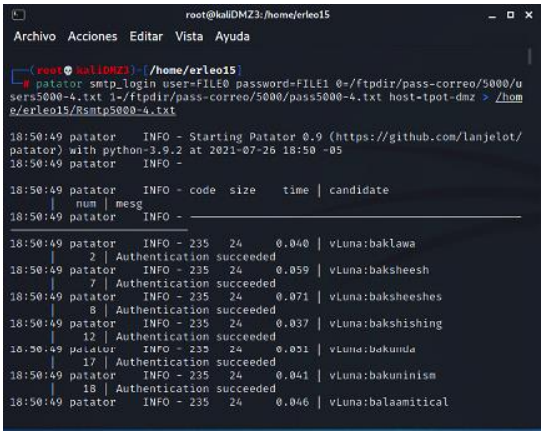


Figura 133. Resultado de la máquina 4 ataque en SMTP

Máquina 5

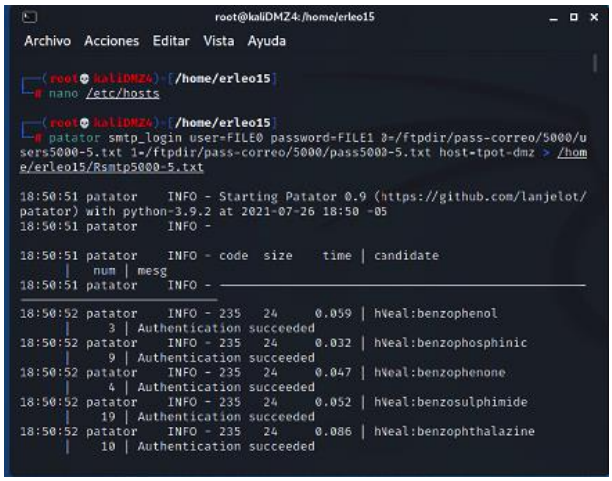


Figura 134. Resultado de la máquina 5 ataque en SMTP

Realización de ataques al honeypot Tanner (HTTP)

1. En el primer caso de prueba comprende una sola máquina con la siguiente salida en consola.

Máquina 1

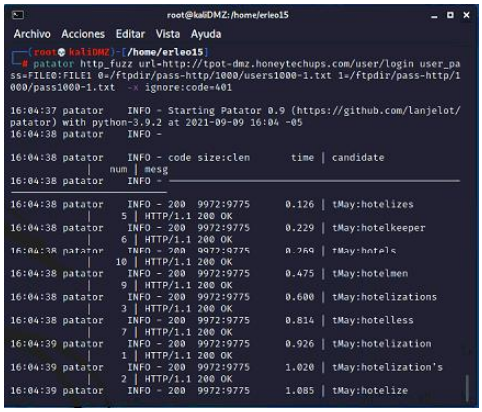


Figura 135. Resultado del ataque en HTTP

2. Para el segundo caso se tienen 3 computadoras atacantes las cuales, al ejecutar los comandos pertinentes nos arrojaron la siguiente salida en consola.

Máquina 1

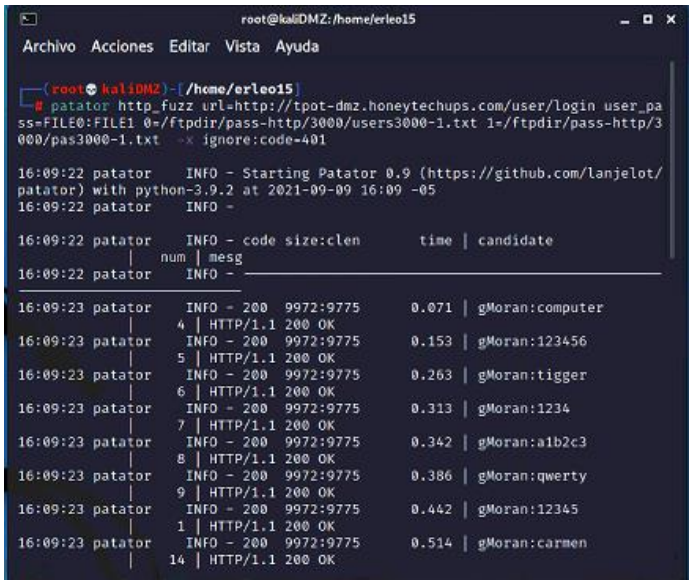


Figura 136 Resultado de la máquina 1 ataque en HTTP

Máquina 2

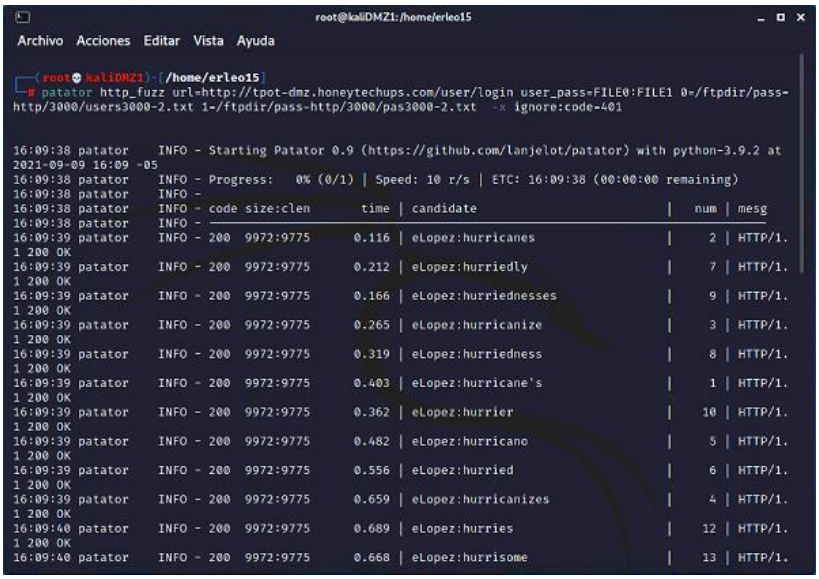


Figura 137. Resultado de la máquina 2 ataque en HTTP

Máquina 3

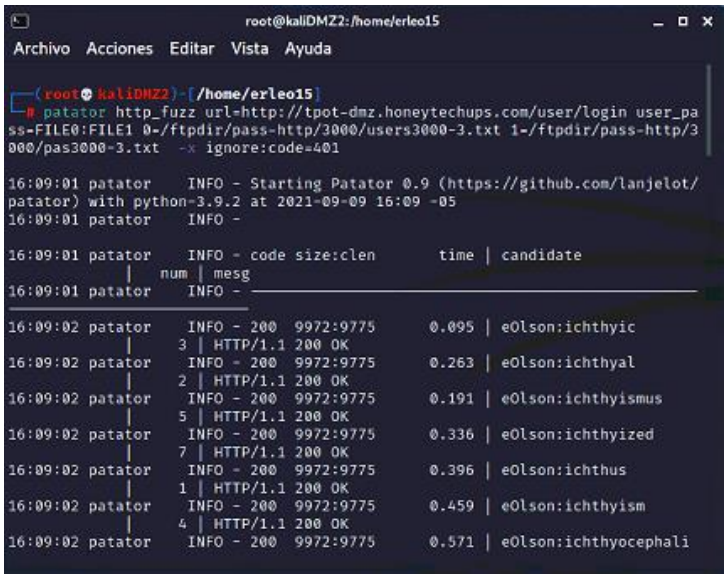


Figura 138. Resultado de la máquina 3 ataque en HTTP

3. Para el tercer caso de pruebas, se usa 5 máquinas las cuales realizaron los ataques desde la línea de comandos, con las siguientes salidas en consola.

Máquina 1

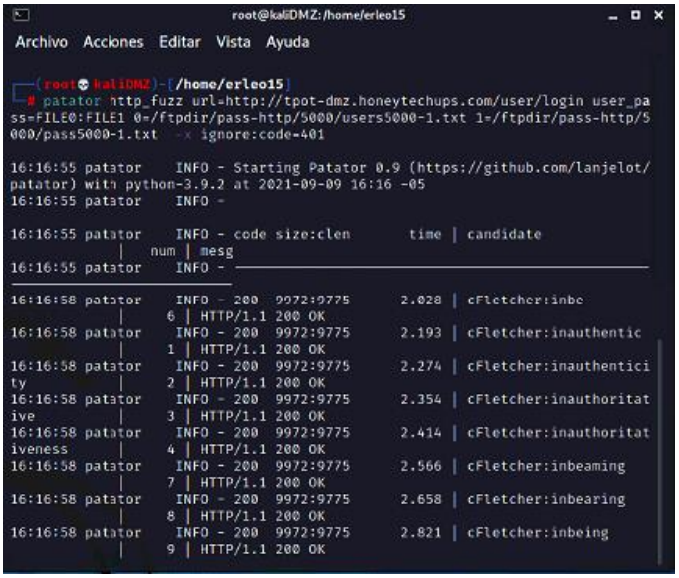


Figura 139. Resultado en consola en la máquina 1 en HTTP

Máquina 2

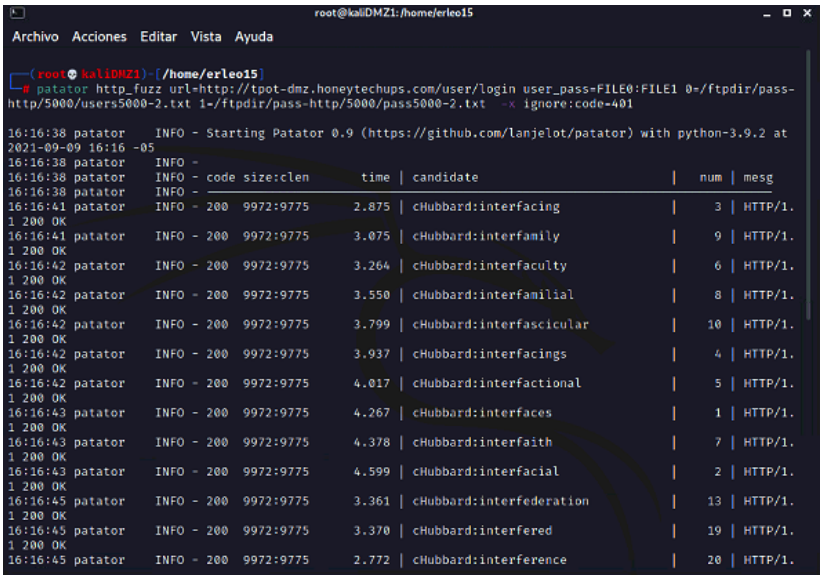


Figura 140. Resultado en consola en la máquina 2 en HTTP

Máquina 3

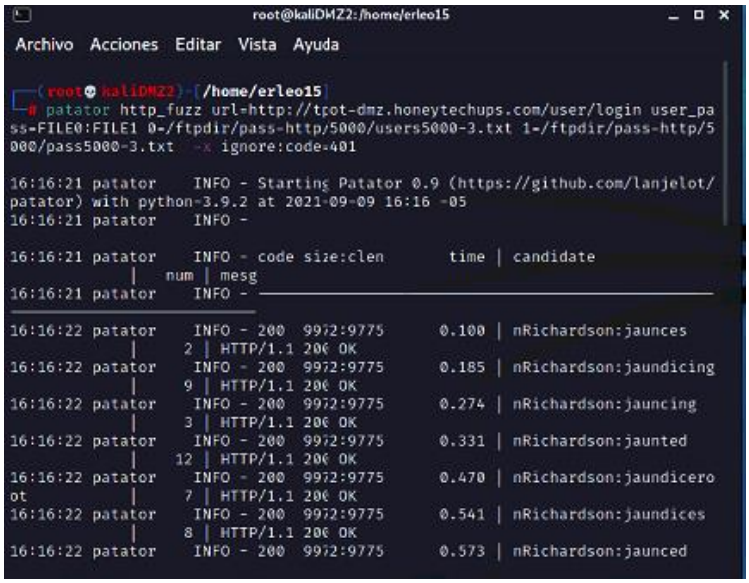


Figura 141. Resultado en consola en la máquina 3 en HTTP

Máquina 4

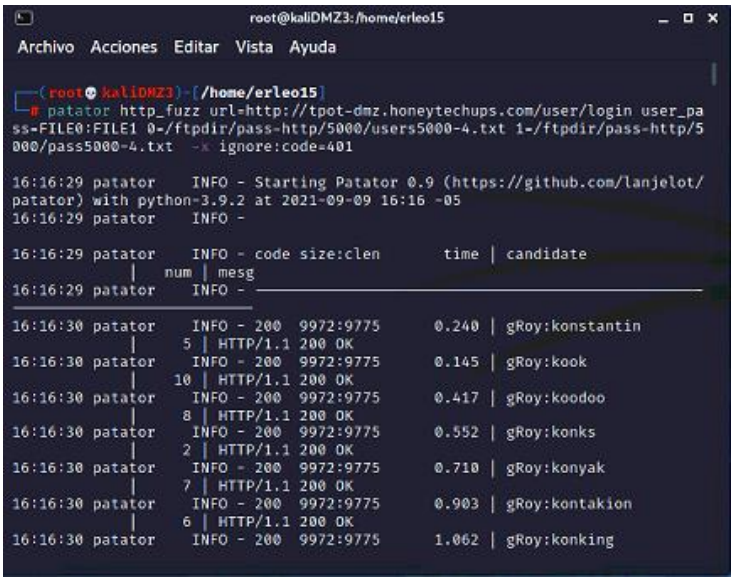


Figura 142. Resultado en consola en la máquina 4 en HTTP

Máquina 5

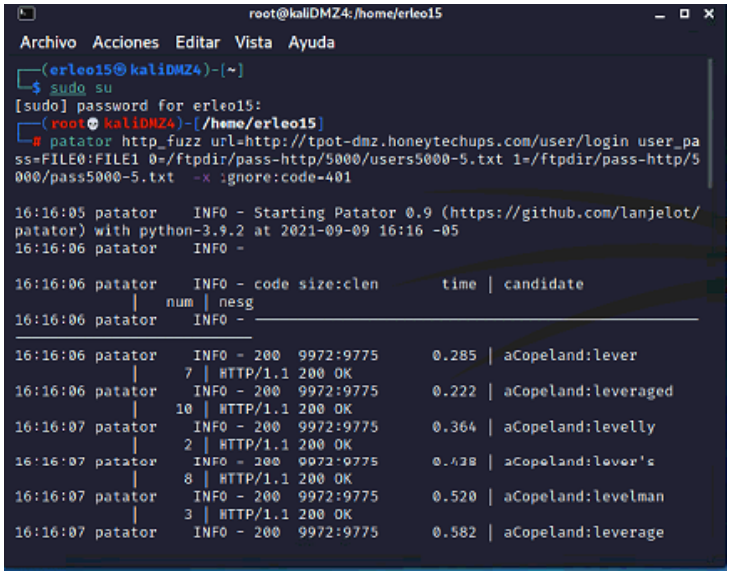


Figura 143. Resultado en consola en la máquina 1 en HTTP

ANEXO 12: Usuarios y contraseñas capturados por los honeypot

En la *Figura 144 y Figura 145* muestran una serie de usuarios y contraseñas que fueron capturados por el honeypot Cowrie así como las *Figura 146* y *Figura 147* muestran de manera similar del honeypot Dioanea, al momento que recibían los ataques desde diferentes máquinas.

Figura 144. Usuarios recolectados por Cowrie

Figura 145. Contraseñas recolectadas por Cowrie

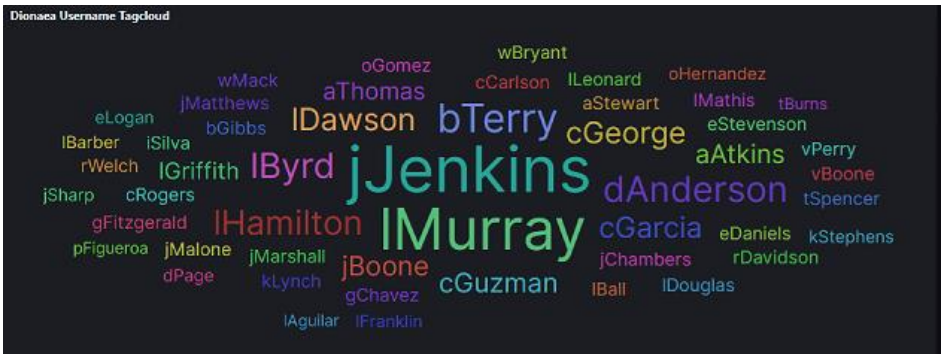


Figura 146. Usuarios recolectados por Dionaea

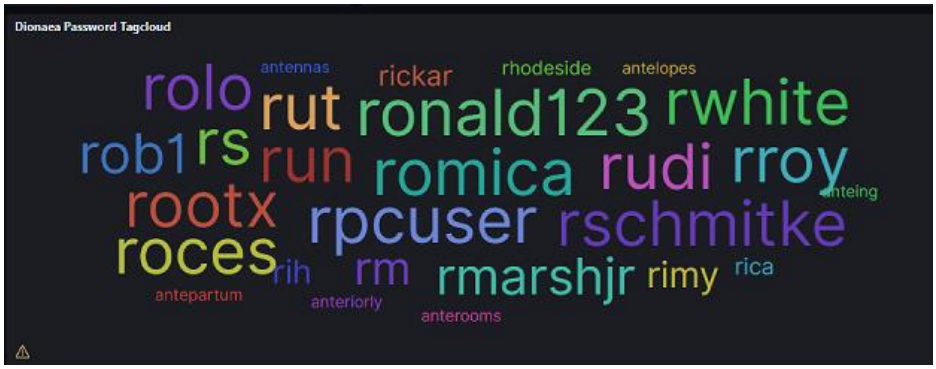


Figura 147. Contraseñas recolectadas por Dionaea

ANEXO 13: Instalación de servicios en Ubuntu Server

Para cada servicio que se instale se recomienda actualizar la base del paquete:

> sudo apt update

1. Instalación de openssh-server

Para la instalación y configuración del sistema se realizó los siguientes pasos:

a. Instalamos OpenSSH-Server con el comando:
>sudo apt-get install openssh-server

b. Habilitamos el puerto de escucha con el comando:
>sudo ufw allow 22/tcp

c. Modificamos el archivo de configuración con el comando
>nano /etc/ssh/sshd_config

y podemos cambiar el puerto de escucha en el apartado “port 22” o más opciones que deseemos.

d. Luego iniciamos el servicio, además de habilitar para que arranque al encender el servidor, con los comandos:
>sudo systemctl start sshd
>sudo systemctl enable sshd

2. Instalación de VSFTPD

Para el proceso de instalación de este servidor FTP se realizó los siguientes pasos:

a. Instalamos Vsftpd con el comando:
>sudo apt install vsftpd

b. Abrimos los puertos que hace uso el servicio con el comando:
>sudo ufw allow 20/tcp && sudo ufw allow 21/tcp && sudo ufw allow 40000:50000/tcp

c. Modificamos la configuración del servidor FTP en el archivo vsftpd.conf con el comando:

>sudo nano /etc/vsftpd.conf

d. Luego reiniciamos el servicio y habilitamos el arranque automático con los siguientes comandos.

>sudo systemctl start vsftpd
>sudo systemctl enable vsftpd

3. Instalación de Postfix

En este proceso de instalación de POSTFIX utilizado para enviar y recibir correos electrónicos se realizó los siguientes pasos:

a. Instalamos postfix en Ubuntu Server con el comando
   >sudo apt install mailutils

Al momento de la instalación nos solicitara un nombre de servidor mail

b. Modificamos la configuración de postfix en el archivo main.cf con el comando:
   sudo nano /etc/postfix/main.cf

c. Finalmente iniciamos el servicio postfix y lo agregamos al arranque inicial con los comandos:
   >sudo systemctl start postfix
   >sudo systemctl enable posfix

4. Instalación de Apache

a. Instalamos apache con el comando:
   >sudo apt-get install apache2

b. Configuramos el servidor apache con las opciones deseadas con el comando:
   >sudo nano /etc/apache2/apache2.conf
   y guardamos la configuración.
c. Iniciamos el servicio y lo agregamos al arranque inicial con los comandos:
   >sudo systemctl start sshd
   >sudo systemctl enable sshd

ANEXO 14: Instalación Kali-Linux

1. Iniciamos la maquina con el disco de Kali-linux y seleccionamos “Graphic install”.

Figura 148. Ventana inicial de KaliLinux

2. Seleccionamos el idioma

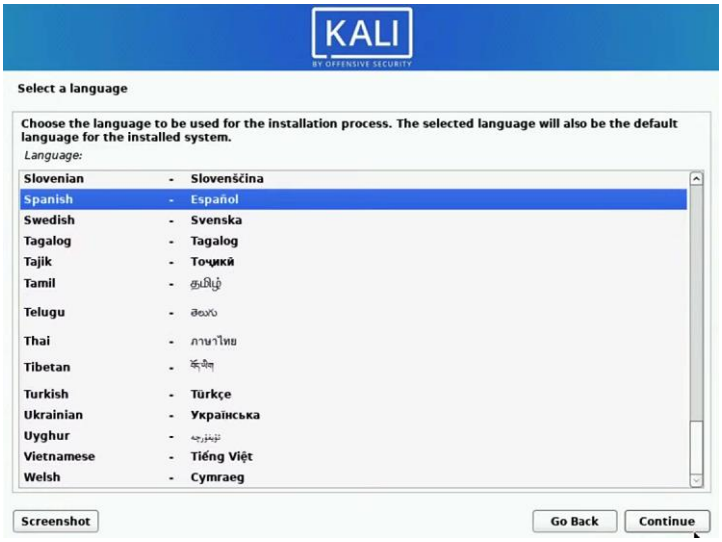


Figura 149. Selección del idioma

3. De la lista buscamos el país de ubicación.

Figura 150. Selección geográfica

4. Se escribió un nombre a la máquina.

Figura 151. Nombre de la máquina

5. Colocamos un nombre de dominio, en caso de ser necesario, de lo contrario dejarlo en blanco.

Figura 152. Nombre de la red o dominio

6. Luego nos pedirá un nombre completo para el usuario.

Figura 153. Nombre de Usuario y Contraseña

7. Entonces nos pedirá un nombre(nick) de usuario.

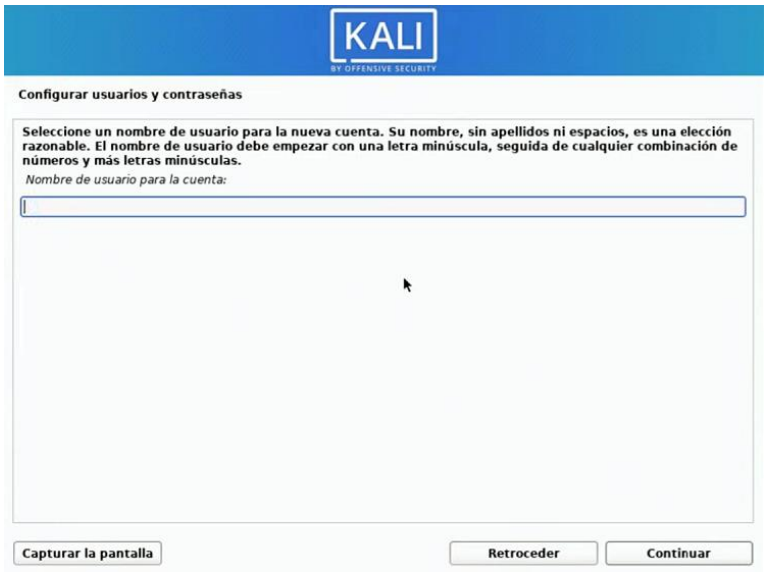


Figura 154. Cuenta del usuario

8. Después tendremos que escribir una contraseña para la cuenta, además de la confirmación.

Figura 155. Confirmación de contraseña

9. Luego se procederá al particionado del disco, donde se instalará el SO.

Figura 156. Partición del Disco

10. Particionar el disco, acorde a las necesidades de instalación.

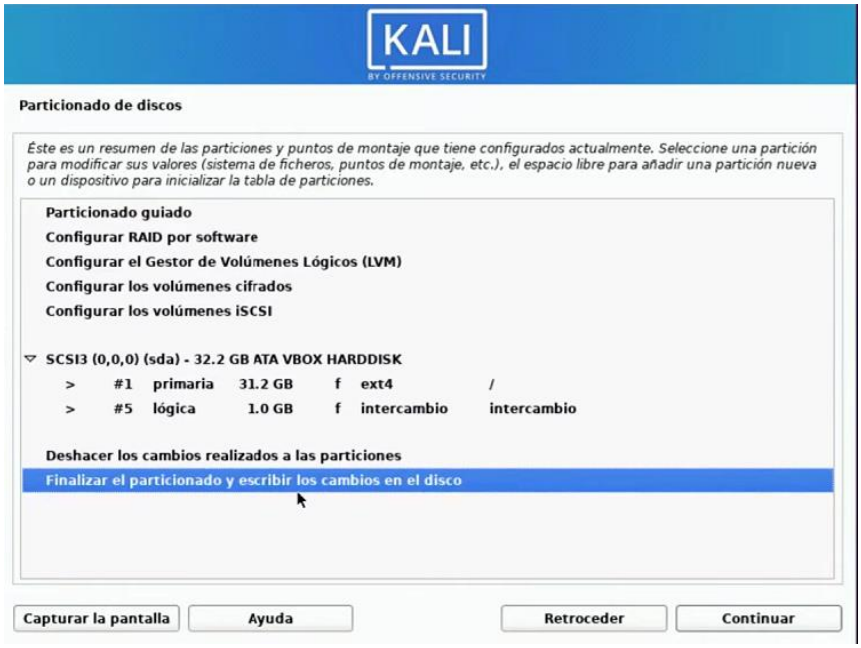


Figura 157. Tabla de particiones del disco

11. En la siguiente ventana confirmamos los cambios que se realizo al disco.

Figura 158. Confirmación de los cambios del disco

12. Luego nos pedirá elegir los programas que se incluirán en la instalación.

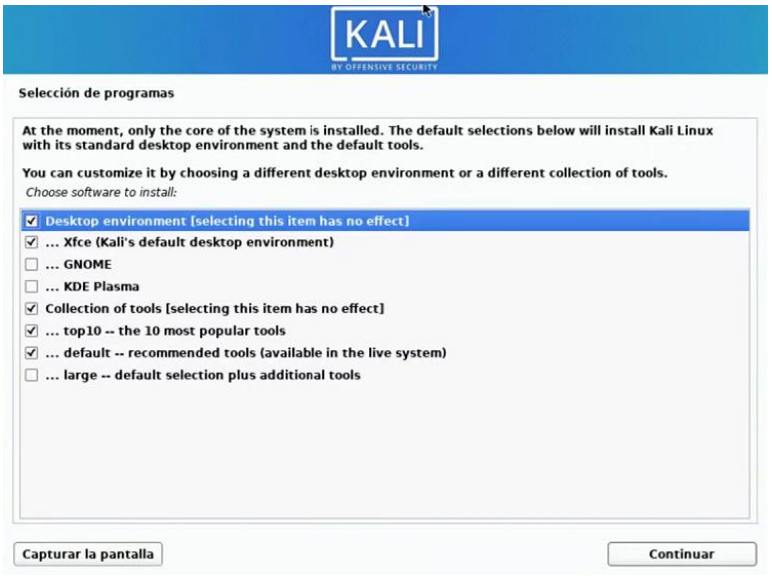


Figura 159. Instalación de Desktop

13. Casi al final de la instalación nos pedirá confirmación para instalar el cargador de arranque GRUB en la máquina, el cual es necesario para arrancar el SO en caso de no tener uno instalado.

Figura 160. Instalación de arranque GRUB

14. Finalmente, se instalará el SO con sus componentes y luego de reiniciarse y completar la instalación tendremos Kali-Linux instalado en la máquina.

Figura 161. Escritorio de Kali-Linux