DISEÑO DE MANUAL BÁSICO DE PRUEBAS DE HACKING ÉTICO: ESCANEO DE RED, DE VULNERABILIDADES Y ATAQUE

Oscar Alberto Rodríguez Cuadros

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
BUCARAMANGA
2018

DISEÑO DE MANUAL BÁSICO DE PRUEBAS DE HACKING ÉTICO: ESCANEO DE RED, DE VULNERABILIDADES Y ATAQUE

Oscar Alberto Rodríguez Cuadros

Monografía para optar por el título de
Especialista en Seguridad Informática

Director del proyecto
Ing. MSc. Alexander Larrahondo Nuñez

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
BUCARAMANGA
2018

Notas de aceptación

_____

Firma del jurado

_____

Firma del jurado

Dedicatoria

Dedico principalmente a mi esposa y su apoyo fundamental para el desarrollo del, mismo. También, a Dios y mi madre quienes han estado siempre conmigo.

Agradecimientos

Infinitamente a Dios y mi familia por permitirme desarrollar este proyecto. Igualmente, resaltar la importancia que la Universidad Abierta y a Distancia generó en mí, con los conocimientos indicados para el desarrollo como profesional especializado.

Tabla de contenido

Lista de imágenes

Glosario

Hacking ético: Es una auditoría efectuada por profesionales de seguridad de la información, quienes reciben el nombre de “pentester”. A la actividad que realizan se le conoce como “Hacking ético” o “Pruebas de penetración”. Guevara 2018

Nmap: Es una herramienta de código abierto para exploración de red y auditoría de seguridad. Se diseño para analizar rápidamente grandes redes. Funciona muy bien contra equipos individuales. Nmap.org 2018

Nessus: Permite el escaneo de vulnerabilidades en servidores web, servicios web. Además de verificar la configuración errónea del sistema y parches faltantes. Muestra informes personalizados en diferentes formatos. Gomes 2014

Metasploit: Es una herramienta que permite ejecutar y desarrollar sploits contra sistemas objetivos. Actualmente se encuentra integrado con Kali Linux, una distribución de Linux con diversas herramientas orientadas a la seguridad. Catoira 2018

OpenVas: Es un framework que tiene como base servicios y herramientas para la evaluación de vulnerabilidades y puede utilizarse de forma individual o como parte de un conjunto de herramientas de seguridad. Mendoza 2018

Pentesting: Es una práctica para poner a prueba un sistema informático, red o aplicación web para encontrar vulnerabilidades que un atacante podría explotar. Rouse. 2016

Resumen

El siguiente documento se enfoca en la identificación de las herramientas de *pentesting*, con el fin de detectar las vulnerabilidades que puedan afectar los equipos que hacen parte de una red. Esto permite analizar los requerimientos que se deben tener en cuenta para aumentar los niveles de seguridad de la red y los equipos conectados. Finalmente, se genera un manual como guía para los administradores de servidores que pueden aplicar a los servidores.

1. Introducción

Este documento presenta un breve recorrido por el *pentesting* o pruebas de penetración que permiten conocer el estado actual de seguridad de un sistema, red o infraestructura. La mayoría de las empresas han optado por entrar al mundo digital con el fin de automatizar, agilizar y mejorar sus procesos. Entre tanto, la seguridad de la información es un tema muy importante para las empresas puesto que, mantener la seguridad de su información, y prever posibles vulnerabilidades permitiendo que se esté, un paso adelante ante un posible ataque de terceros, y se constituye como el insumo principal de las empresas.

En tal sentido, para el presente documento se pretende abarcar tres objetivos principales que demarcan el hilo conductor y cumplir así con el propósito fundamental, que consiste en la elaboración del Manual de configuración básica para prever las posibles vulnerabilidades por configuraciones erradas. Por ende, se trabajan tres objetivos, a saber: a. Determinar las principales herramientas de *pentesting* para el hallazgo y explotación de las vulnerabilidades. B. Aplicar y analizar los resultados de las principales pruebas de pentesting, para determinar la configuración mínima de seguridad que debe ser aplicada por los administradores de estos sistemas. Por último, c. Elaborar un manual de pruebas de hacking ético, con los resultados obtenidos en las pruebas y recomendaciones en configuración para los administradores de estos sistemas.

Ahora bien, el alcance que se pretende lograr con el presente documento consiste en la elaboración de dicho manual, cuyo enfoque radica en el escaneo, detección de vulnerabilidades y ataque. Con el manual elaborado se pretende beneficiar a los administradores de red o especialistas en seguridad informática que comienzan en el mundo del testeo de vulnerabilidades.

En cuanto a la metodología, se seleccionan tres pasos principales para alcanzar el objetivo principal, como se ha mencionado anteriormente, que es el Manual. En primera instancia, es la “determinación de las herramientas de *pentesting*”, que en el caso particular, se seleccionaron tres, a saber; Nmap, Nessus y Metasploit Framework. En segunda instancia, se realiza una “ejecución y análisis de las pruebas”, mencionadas anteriormente. En tercera y última instancia, se “genera una guía o manual”, que permita en los administradores de red o especialistas en seguridad informática identificar los riesgos y vulnerabilidades a los que está expuesta una red o lo equipos que la integran.

Finalmente, en cuanto a la aplicación del área de conocimiento en particular, pese a que en el campo de la ingeniería y específicamente en la seguridad informática, se usan las más comunes tales como: debian, redhat entre otros, es preciso resaltar que quienes se podrán ver principalmente beneficiados, son las pequeñas

y medianas empresas que optan por la versión gratuita de Linux como sistema operativo. Por ende, se invita a conocer más de este sistema operativo y desarrollar nuevas utilidades, que en el caso particular incluye a los administradores de servidores, y demás personas que deseen adquirirlo, sin olvidar que la seguridad informática debe ser parte fundamental de todos los sistemas operativos, sobre todo Linux.

2. Definición del Problema

Normalmente las empresas que implementan sus propios sistemas de redes y servidores necesitan una metodología que les indique que tan seguros están sus datos y que tan fuertes son sus medidas de seguridad. A menudo los administradores de sistemas confían en los protocolos implementados en el software de diario uso como son los sistemas operativos.

En muchos casos las intrusiones a los sistemas se dan por el aprovechamiento de los fallos propios del software y que en el peor de los casos estas vulnerabilidades nunca son puestas a la luz y se crea una falsa sensación de seguridad.

Las pruebas de pentesting o pruebas de intrusión permiten medir los niveles de seguridad de los sistemas dejando en evidencia las vulnerabilidades o huecos de seguridad que pueden ser aprovechados y permite a las organizaciones tomar medidas en pro de fortalecer sus puntos débiles.

Tratar de medir el nivel de seguridad de las organizaciones es un proceso complejo ya que se está en juego la información que es esencial para su funcionamiento. Lo anterior conlleva a determinar.

¿Cómo el diseño y aplicación de pruebas de hacking ético permiten descubrir vulnerabilidades y determinar los niveles de seguridad?

2.1. Justificación

En la actualidad las empresas tienden a tener su propia infraestructura en cuanto a redes, servidores, *routers*, etc. Esto les permite estar seguros de que toda su información y los datos críticos de sus operaciones estarán bajo su control y no en manos de terceros.

Muchas empresas y administradores de sistemas son muy novatas en temas como seguridad de la información, seguridad en redes, servidores o sistemas operativos, lo cual implica que la información puede estar expuesta ante amenazas e intrusiones que puedan perjudicar a la organización.

Por este motivo se hace necesario realizar un análisis, descripción y definición de las pruebas de pentesting o pruebas de penetración requeridas que le indiquen a administradores el nivel de seguridad y las vulnerabilidades que se deben fortalecer para evitar ser víctima de ataques y exponer la integridad de su red e información.

El desarrollo de este proyecto permite establecer los pasos básicos para realizar pruebas de penetración sobre un sistema o red, permitiendo descubrir vulnerabilidades con el objetivo de minimizar los riesgos de intrusión.

3. Objetivos

3.1. Objetivo General

Diseñar manual básico para pruebas de hacking ético: escaneo de red, de vulnerabilidades y ataque.

3.2. Objetivos Específicos

- Definir las principales pruebas de pentesting o pruebas de intrusión.
- Aplicar y analizar los resultados de las principales pruebas de pentesting o pruebas de intrusión en un entorno controlado o virtualizado.
- Elaborar un manual básico de pruebas de hacking ético.

4. Marco referencial

4.1. Marco conceptual

Sistema operativo: “Conjunto de programas que gestionan los recursos del sistema y optimizan su uso”.

Vulnerabilidad: “Agujero de seguridad de una parte del software, hardware o sistema informático, que proporciona una potencial fuente de ataque al sistema. Una vulnerabilidad puede comenzar desde una contraseña débil hasta una inyección de SQL.

Exploit: Es un programa que aprovecha la vulnerabilidad de un sistema y provee un acceso a la información de usuario y privilegios del sistema.

Payload: Programa que permite tomar el control de un sistema después de ser explotada alguna vulnerabilidad.

Ataque: Intento indebido y no autorizado de una persona o grupo organizado de acceder a un sistema aprovechando alguna de sus fallas ya sea de software, hardware o de su personal, por medio de los diferentes estilos de ataque.

Pruebas de intrusión: Ejecución de herramientas especializadas en un entorno real con el fin de determinar el nivel de seguridad y protección con que cuenta un sistema informático. Este tipo de pruebas permiten determinar qué tipo de fallos se encuentran en un sistema y que puedan resultar desastrosas si se llegasen a explotar. Para la ejecución de las pruebas de intrusión es necesario tener la aprobación de la organización a la cual se le medirá el nivel de seguridad de su sistema de información o sistemas informáticos, ya que de otra forma se consideraría como un acto ilegal de acceso a la información.

Reverse TCP: Ataque que consiste en la comunicación de un equipo victima con el equipo atacante a través de un puerto de escucha que recibe la conexión y permite que el atacante pueda ejecutar diferentes comandos sobre la maquina víctima.

Bind TCP: Ataque donde por medio de un puerto abierto en el equipo víctima se crea una conexión entrante, esta conexión permite la ejecución de código y comandos por parte del atacante.

Caja blanca: Son pruebas en las que se tiene conocimiento total del sistema que se pretende atacar. El objetivo de esta prueba es simular el comportamiento de un atacante que cuenta con permisos de acceso a la información del sistema.

Caja negra: Son pruebas que se realizan a un sistema informático desconociendo su funcionamiento interno pero conociendo sus entradas y sus salidas. Permiten simular el escenario donde el atacante no tiene información del funcionamiento del sistema e intenta obtener acceso a la información. De esta forma se permite establecer el nivel de seguridad y protección de los sistemas testeados.

Caja gris: Es una combinación entre las pruebas de caja negra y caja blanca, se conoce parte del código y su funcionamiento, además de las entradas y las salidas. Para llevar a cabo este tipo de pruebas es necesario tener permisos al sistema para simular un ambiente casi real a los procesos que realizan los usuarios.

Ethical Hacker: Profesional calificado que entiende y sabe cómo buscar debilidades y vulnerabilidades en los sistemas y usa sus conocimientos y herramientas con el fin de alertar y prevenir la explotación de fuentes de ataques informáticos.

Activo: Recurso del sistema informático o relacionado con este, necesario para que la organización pueda realizar sus operaciones de negocio y alcance los objetivo.

4.2. Marco teórico

Las pruebas de intrusión consisten en probar los métodos de protección del sistema sometiéndolo a una situación real.

Generalmente las pruebas de intrusión incluyen tres métodos:

- El método de la caja negra: Consiste en intentar realizar una intrusión sin tener el menor conocimiento del sistema o sus implementaciones de seguridad.
- El método de la caja blanca: Consiste en intentar penetrar el sistema conociéndolo por completo para poner a prueba todas sus implementaciones de seguridad.
- El método de caja gris: Consiste en realizar una simulación de ataque por parte de un empleado interno de la organización que dispone de cierta información como por ejemplo un usuario y una contraseña. El objetivo de este test de intrusión es detectar vulnerabilidades que permitan elevación de privilegios a usuarios.

Normalmente estas pruebas son una buena forma de generar conciencia en las organizaciones a cerca de las falencias que puedan tener los sistemas. También

permite detectar el nivel de seguridad de los sistemas evaluados y conocer el nivel de acceso que tendría un atacante.

Estos tests permiten a las organizaciones conocer el grado de vulnerabilidad de sus sistemas de información y permite tomar medidas ante los posibles ataques externos e internos.

Es importante definir el alcance de un test de intrusión antes de su ejecución. Los test de intrusión pueden ser muy amplios, estos pueden incluir toda la infraestructura de la organización o puede estar enfocado a un determinado sistema, equipo o aplicación.

Una vez se defina el alcance del test de intrusión se debe tener en cuenta lo siguiente:

- Horario en el que se llevará a cabo el test de intrusión.
- ¿Está permitido llevar a cabo denegaciones de servicio?
- ¿Está permitido instalar Backdoors?

¿Está permitido realizar defacement de las aplicaciones web?

- ¿Está permitido realizar borrados de logs?
- ¿El personal de la organización tendrá conocimiento del test de intrusión?
- ¿Se llevarán a cabo técnicas de ingeniería social?

4.2.1. Características de un test de caja negra:

La principal característica de un test de caja negra es simular el ataque de un hacker con sus propios recursos. Solo se dispone de información pública sobre el objetivo, y a través de esta se dispone a identificar los agujeros de seguridad que puedan comprometer la información sensible de la organización o las operaciones del sistema. Este test simula un verdadero ataque que intenta obtener acceso al sistema.

Ventajas:

1. Se determina el estado real de las amenazas al sistema.
2. Se obtienen los resultados a través de información pública.
3. Requiere un esfuerzo mínimo del cliente.

Desventajas:

1. Puede ser un esfuerzo recopilar la información.
2. Pueden pasar desapercibidas puertas traseras o vulnerabilidades parciales.

3. Las recomendaciones para reparar fallos pueden ser parciales.

Pruebas realizadas con esta metodología:

1. Pruebas de penetración de infraestructura.
2. Pruebas de penetración de aplicaciones.
3. Ataque simulado completo.

4.2.2. Características de los test de caja blanca:

Se debe tener suficiente información para evaluar la seguridad del entorno a prueba, incluyendo código fuente, archivos de configuración, documentos y diagramas. Esto permite una revisión a fondo del sistema, identificando no solo las vulnerabilidades inmediatas, sino también las secciones de código y configuraciones potencialmente peligrosas. Este tipo de test es adecuado para entornos muy sensibles.

Ventajas:

1. Extremadamente minucioso.
2. Las recomendaciones para reparar los fallos son muy precisas.
3. Detecta las amenazas inmediatas, así como los defectos de configuración.

Desventajas:

1. Requiere muchos recursos tanto del que realiza el test como del cliente.
2. No se simula un verdadero ataque.

Pruebas realizadas con esta metodología:

1. Solicitud de revisión de código.
2. Examen de diseño y construcción.
3. Configuración del servidor de Auditorías.
4. Auditorías de red.

4.2.3. Características de un test de caja gris:

En este test se combinan los test de caja negra y caja blanca. Se realizan pruebas similares a los test de caja negra, simulando ataques reales. Para esta prueba se debe disponer de información técnica del sistema e información adicional del sistema.

Este test es ideal para obtener un número mayor de vulnerabilidades reales en un menor tiempo. Es un método que permite realizar una evaluación eficaz de seguridad.

Ventajas:

1. Más rentable.
2. Proporciona una estimación realista de las amenazas.

Desventajas:

1. No simula las condiciones reales de un ataque de caja negra.

Pruebas realizadas con esta metodología:

1. Aplicación de pruebas de penetración (con la revisión de código parcial).
2. Test de la infraestructura y la red de penetración (con revisión de la red y la configuración del servidor.)
3. Examen de alto nivel de diseño y construcción.

4.3. Principales herramientas para realizar pruebas de intrusión y verificación de vulnerabilidades

OWASP Zed Attack Proxy (ZAP): herramienta de fácil uso para encontrar vulnerabilidades en aplicaciones web. Está diseñada para ser utilizada tanto por desarrolladores y probadores funcionales (que son nuevos en tests de intrusión) como por personas con una amplia gama de experiencia en seguridad. Permite automatizar las pruebas y también facilita un número de herramientas para hacerlas manualmente.

BeEF (The Browser Exploitation Framework): marco modular que utiliza técnicas pioneras que proveen la posibilidad de realizar pruebas de intrusión y poder experimentar varios vectores de ataques de carácter práctico. La herramienta se centra en el aprovechamiento de las vulnerabilidades del navegador web para abarcar la seguridad desde un punto objetivo.

Burp Suite: herramienta que permite realizar test de intrusión en aplicaciones web, permitiendo combinar técnicas manuales y automáticas para enumerar, analizar, atacar y explotar aplicaciones Web. Puede funcionar como proxy entre nuestro navegador e Internet, pero además tiene otras muchas funcionalidades como un spider, un escáner o un repetidor, por ejemplo.

PeStudio: herramienta gratuita para realizar análisis estáticos de binarios ejecutables de Windows. Un fichero analizado con PeStudio no se ejecuta por lo que puede evaluarse sin riesgo, aunque sea malware. Además, tiene entorno gráfico y es portable.

OWASP Xenotix XSS Exploit Framework: es un marco avanzado de explotación y detección de vulnerabilidades de tipo Cross Site Scripting (XSS). No proporciona

casi fasos positivos y utiliza un triple motor de navegación (Trident, WebKit y Gecko). Tiene más de 1500 payloads distintos y es capaz de evadir WAF.

Lynis: herramienta de seguridad y auditoría para sistemas basados en Unix/Linux. Escanea el sistema ejecutando muchas pruebas de seguridad, revisa el software instalado y determina si cumple los estandars. También detecta fallos de seguridad y errores de configuración.

Suricata: es un IDS/IPS de red de alto rendimiento y un motor para la monitorización de la seguridad de la red. Es de código abierto y propiedad de una fundación sin ánimo de lucro dirigida a la comunidad, la Open Information Security Foundation (OISF). Es altamente escalable, puede identificar la mayoría de los protocolos, permite identificación y extracción de ficheros y checksums MD5.

WPScan: es un escaner de vulnerabilidades de tipo caja negra para Wordpress. Está escrito en Ruby y permite comprobar vulnerabilidades conocidas en instalaciones de Wordpress.

4.4. Clasificación de los ataques

Podemos encontrar cuatro diferentes tipos de ataques: Ataques activos, ataques pasivos, ataques a contraseñas y ataques de código malicioso.

4.4.1. Ataques activos

Este tipo de ataques son realizados para causar el mayor daño posible a un sistema, mediante la obtención de los servicios, con el fin de modificar la configuración de cada uno de ellos o detener su ejecución. Los ataques activos son visibles dado que sus consecuencias son detectables a simple vista. En esta categoría se encuentras:

- Denegación de servicios.
- Bufer Overflows
- Spoofing
- MITM – Man In The Middle
- TCP/IP Hijacking
- Ingenieria social

4.4.2. Ataques pasivos

Los ataques pasivos no afectan directamente a la red víctima, solo escuchan lo que viaja a través de esta, recopilando la información importante, desde conversaciones hasta claves de seguridad. Entre estos ataques se encuentran:

- Análisis de vulnerabilidades.

- Escaneo de redes y espionaje.

4.4.3. Ataques a contraseñas o criptografía

Los ataques de contraseña son los ataques que más se aplican por su facilidad y por la cantidad de herramientas disponibles para esto. Hay dos tipos de ataques:

- Ataque de fuerza bruta.
- Ataque basado en diccionario.

4.4.4. Ataque de código malicioso

A través de los años los malware han sido el tipo de infección informática que más tiene reconocimiento debido a su capacidad de cambio para ser detectado, esto sumado a la capacidad que tiene para propagarse en internet a través de correos electrónicos, memorias USB, descargas desde páginas poco conocidas. La ingeniería social es la forma más activa de propagación de este tipo de infección ya que por medio de enlaces o correos electrónicos con supuestos contenidos atractivos para los usuarios y se descargan en el equipo y allí permanecen hasta ejecutarse y cumplir su objetivo. Entre los tipos de malware se encuentran:

- Virus.
- Troyanos.
- Bombas lógicas.
- Gusanos.
- Puertas traseras.

4.4.5. Ataque de denegación de servicios DoS

Aprovecha la capacidad de uso de la red para saturar un sistema con envío de solicitudes simultaneas, con el objetivo de superar la capacidad de respuesta del sitio o sistema y de esta forma detener el buen funcionamiento de este. Normalmente los DoS se aprovechan de la capacidad limitada que tienen los recursos de red, como los servidores web, cuando el número de solicitudes supera este limite el servicio se ve afectado generando respuestas lentas y desechando las demás solicitudes de los usuarios.

4.4.6. Desbordamiento de memoria

Este tipo de ataque aprovecha los errores cometidos por los programadores. Estas deficiencias pueden ser explotadas por este ataque conocido como desbordamiento de memoria, este ataque envía demasiados datos al buffer con el fin de que este se sature, esta parte del sistema es un área de memoria temporal que almacena datos o instrucciones. Normalmente para ejecutar este tipo de ataques se reemplazan los datos de la memoria por otros datos que la mayoría de

las veces son caracteres sin ordenamiento, ocasionando que el programa deje de funcionar.

4.4.7. Spoofing

Proporciona información falsa acerca de su identidad, con el fin de ganar acceso no autorizado al sistema. El ejemplo más clásico es IP spoofing, en donde un atacante crea un paquete IP con la dirección de origen de otra máquina.

4.4.8. Ataque Man in The Middle

Realiza sniffing a una red posicionándose en medio de la puerta de enlace y un servidor o red, esto se logra realizando un ataque al ARP (Protocolo de resolución de direcciones) que tome como puerta de enlace la máquina del atacante, para luego cambiar la Mac de la puerta de enlace por la Mac del atacante.

4.4.9. Ingeniería Social

La ingeniería social facilita conseguir información de las personas que tiene acceso a un sistema. Se podría definir como la metodología que permite conseguir información de interés a través de terceros. Para esto los malware se capacitan de habilidades sociales, para engañar fácilmente al usuario.

4.4.10. Análisis de vulnerabilidades

Por medio de este análisis se puede extraer información de los sistemas y servicios para determinar si existe algún exploit conocido. Existen herramientas especializadas para encontrar dichas falencias como lo es Nmap, que escanea los puertos enviando paquetes al host con el fin de recopilar información importante como el tipo de sistema operativo.

Es necesario saber qué tipo de información se desea recopilar para escoger la herramienta que más se adapte. Otro ejemplo es Nessus, el cual, permite el análisis de vulnerabilidades por medio de escaneo múltiple sobre diferentes arquitecturas, los resultados de los análisis son mostrados en un informe de manera detallada.

4.4.11. Ataques a contraseñas por fuerza bruta y diccionario.

Los procesos para este tipo de ataque se realizan por fuerza bruta, mediante diccionarios de datos, que combinados miles de veces se logran encontrar la contraseña establecida por el usuario. Es por esto que los sistemas han implementado HASH en sus contraseñas, permitiendo mayor seguridad en el proceso de identificación.

4.5. Metodologías de pruebas de intrusión

La aplicación de estándares de pruebas de intrusión permite realizar este proceso de manera ordenada y sistemática. Estas metodologías son aplicadas por personal especializado en la seguridad informática.

4.5.1. Manual de la metodología abierta de testeo de seguridad (OSSTMM)

Es uno de los estándares profesionales más completos, provee de un manual con una metodología abierta del test de penetración de seguridad. La metodología está dividida en diferentes secciones:

- Sección A: Seguridad de la información.
- Sección B: Seguridad de los procesos.
- Sección C: Seguridad en las tecnologías de internet.
- Sección D: Seguridad en las comunicaciones.
- Sección E: Seguridad inalámbrica.
- Sección F: Seguridad Física.

4.5.2. Marco de evaluación de la seguridad de los sistemas de información (ISSAF)

El marco de evaluación de la seguridad de los sistemas de información es una metodología más detallada técnicamente y que divide el proceso de evaluación en fases:

- Fase 1: Planeación
- Fase 2: Evaluación.
- Fase 3: Tratamiento
- Fase 4 – Acreditación.
- Fase 5 – Mantenimiento.

En la segunda fase se realiza la prueba de intrusión. Esta fase requiere de las siguientes actividades:

- Recopilación de información.
- Mapeo de la red trabajo.
- Identificación de vulnerabilidades
- Penetración.
- Obtener acceso y escalar privilegios.
- Enumeración.
- Componer usuarios remitos y sitios

4.6. Kali Linux

En marzo de 2013 fue liberada la distribución que reemplazaría la anterior versión BackTrack. Los cambios más importantes son el cambio de Ubuntu a Debian. Después de una revisión de sistema BackTrack se eliminan las herramientas innecesarias, quedando con un poco más de 350 herramientas. Soporta el sistema de archivos HFS, y las arquitecturas basadas en ARMEL y ARMHF.

4.6.1. Categoría de las herramientas de Kali linux

Las herramientas de Kali Linux están organizadas por categorías que permiten identificar el tipo de análisis o acción a realizar al momento de atacar un sistema.

Entre las herramientas más comunes se encuentran:

- **Vulnerability Analysis:** Posee herramientas de escaneo en búsqueda de vulnerabilidades en sistemas Cisco. Una de las herramientas de esta categoría más conocida es Nmap, el cual es potente escáner que permite detectar los equipos en la red con todas sus características para poder analizarlos.
- **Web aplications:** Herramientas necesarias para lanzar escaneos en internet, compuesto por scripts con capacidades de análisis de bases de datos, inyección de SQL, captura de elementos de páginas WEB, etc:

- **Password attacks:** Tiene alrededor de 36 herramientas para realizar este tipo de ataques, algunas de estas emplean la fuerza bruta para lograr sus objetivos.
- **Wireless attacks:** Tiene un listado de 32 herrramientas, que permiten el scaneo de trafico de red.
- **Explotation tools:** Herramientas exclusivas para las labores de explotación de vulnerabilidades. En la lista se encuentran herramientas para la realización de pruebas de penetración en sitios web, pruebas básicas y avanzadas en dispositivos Cisco, ataques de inyección SQL en aplicaciones web.
- **Sniffing/spoofing:** Este tipo de herramientas son muy utilizadas para búsqueda y explotación de vulnerabilidades. Entre las herramientas más utilizadas se encuentra HexInject, la cual mediante el uso de scripts altera el tráfico de la red mediante modificaciones o interceptaciones.
- **Reverse Engineering:** Categoría que contiene un listado de herramienta para la realización de ingeniería inversa, realizando la extracción del diseño.

- **Forencis:** Herramientas destinadas a realizar tareas forenses en escenas donde lo importante es conocer las últimas acciones realizadas por el sistema
- **Stress Testing**. Entre las herramientas de esta sesión se encuentra FunkLoad, la cual permite realizar pruebas de funcionamiento y de regresión de proyectos.

- **Hardware Hacking:** Herramientas especializada en atacar y comprometer la parte física de los sistemas con el fin de alterar el código Shell del hardware para alterar su funcionamiento.

CAPÍTULO I

HERRAMIENTA DE PENTESTING PARA IDENTIFICACIÓN DE VULNERABILIDADES

5. Capítulo I: Herramienta de pentesting para identificación de vulnerabilidades

En la actualidad, las empresas sin importar su magnitud, alcance, su enfoque económico o su objetivo de negocio han optado por llevar su información a algún tipo de infraestructura tecnológica, o sistema que le permita agilizar los procesos de negocio. Entre tanto, lo que antes tomaba un tiempo procesar, analizar o inclusive gran cantidad de espacio para almacenar en información, hoy se puede reducir a sistemas informáticos en la nube, almacenamiento en línea y medios electrónicos, para compartir información en segundos en cualquier parte del mundo.

No obstante, la mayoría de las empresas ya sean de base tecnológica o perteneciente a otro gremio, cuentan con un departamento de tecnología e innovación o con algún proveedor tecnológico que les permita integrar sistemas informáticos dentro de sus procesos. Para las empresas y en especial las que no son propiamente del entorno de la tecnología, suelen no tener muy en cuenta la importancia de la seguridad de la información, quedando expuestos a que individuos, tanto fuera como dentro de la entidad, puedan acceder a datos confidenciales de la empresa sin que esta tenga conciencia sobre ello. Teniendo en cuenta lo anterior, es importante enfatizar en que todas las empresas sean conscientes de mantener sus activos de información e infraestructura protegidos y que constantemente sean auditados, con el fin de identificar fallas o puntos de mejora que permitan aumentar los niveles de seguridad de la información dentro de la entidad o empresa. Para ello, todos los colaboradores de una organización deben tener presente que la primera línea de defensa de un sistema informático es el usuario final de éste, puesto que, como usuarios de un sistema, son el punto más débil por el cual un atacante puede romper cualquier tipo de seguridad implementada y llegar a la información o inclusive modificar e inhabilitar el acceso a los datos.

Entre tanto, en Colombia según datos de la fiscalía y entes gubernamentales, los delitos informáticos han aumentado en un 60%. Por ende, una de las partes más importantes dentro de la infraestructura de una organización son los servidores, ya que estos contienen información crítica y confidencial de la organización, además de otros contenidos de vital importancia como son las aplicaciones de los usuarios y las bases de datos.

En síntesis, el objetivo del presente documento centra su desarrollo sobre la verificación y auditoría básica de un servidor Debian Linux, con el fin de generar un manual de recomendaciones en configuración a nivel de seguridad que permita a los administradores de estos sistemas tener una seguridad mínima.

5.1. Aproximación sobre la seguridad informática

La seguridad informática consiste básicamente en tres elementos: a. disponibilidad de la información. B. Integridad de la información, y c. Privacidad de la información. Por ende, todas las empresas necesitan auditar sus sistemas informáticos tanto redes como servidores, lo que permite que se descubran puntos débiles en la seguridad y se puedan tomar medidas preventivas para evitar ataques. Este proceso, es comúnmente denominado *Ethical Hacking* o Hacking Ético.

5.2. El procedimiento formal de hacking ético

Para comprender a profundidad, y lograr llegar a una definición sobre el concepto del procedimiento formal del Hacking Ético, es importante hacer mención al manual OSSTMM (Open Security Testing Methology Manual) mencionado por Barreto (2018) que la define como una metodología abierta para el testeo de la seguridad. Este manual es una de las más completas referencias para realizar auditoría de sistemas o redes informática.
Entre tanto, haciendo mención a Veloz (et. Al) (2017) el objetivo del Hacking ético consiste, en realizar una auditoría de seguridad para determinar los puntos débiles, vulnerables e identificar configuraciones no adecuadas, en los sistemas o aplicaciones. Con base en lo anterior, se pueden presentar unos referentes de mejora, con el fin de aumentar los niveles de seguridad, y disminuir los tiempos de reacción ante posibles situaciones de peligro, como, por ejemplo, un ataque informático. Esto permite que la información de una empresa pueda tener disponibilidad, ser confidencial e íntegra. Lo anterior, son los tres pilares fundamentales para la seguridad informática y de la información.

A continuación, se exponen las pruebas de intrusión a tener en cuenta para la realización del objetivo principal. A saber, las herramientas principales a utilizar son las siguientes: 1. NMap. 2. Nessus y 3. MetaSploidframework.

5.3. Prueba de intrusión o pentesting

Las herramientas para realizar auditorías de seguridad o *pentesting* en servidores buscan identificar debilidades o vulnerabilidades existentes, evidenciando puertos abiertos, cerrados, filtrados y los servicios que se exponen a través de estos. Para realizar un *pentesting* se deben tener en cuenta unas fases o pasos que permiten de forma ordenada obtener información sobre el sistema a auditar. La información obtenida sobre el sistema auditado o servidor no permite saber cuáles son los puntos débiles y qué tipo de ataque puede ser realizado con el fin de obtener beneficios de sus vulnerabilidades.

5.3.1. Fases del pentesting o prueba de intrusión

Según Blanco López (2018) el *pentesting* define cuatro fases en las que se lleva a cabo un ataque informático, a saber:

Fase de reconocimiento: permite obtener la mayor cantidad de información sobre el sistema víctima. Normalmente, esta fase se realiza en dos formas distintas:

- En la primera se intenta obtener información del sistema víctima sin tener un conocimiento previo del sistema, ni la versión del servidor o algún detalle que permita conocer alguna debilidad del sistema.
- En la segunda forma se intenta obtener información importante del servidor basado en un conocimiento previo del sistema, tal como el sistema operativo instalado en el servidor o su versión. Esta información es importante ya que permite reducir la incertidumbre al momento de realizar un ataque y que se pueden explotar vulnerabilidades conocidas si se sabe qué versión está corriendo el servidor.

Fase de escaneo: consiste en analizar la información que se obtuvo de la fase de reconocimiento y utilizarla para realizar un sondeo de la red o servidor. Según Benavidez y Velásquez (2015) en esta fase se pueden usar herramientas como:

- Escáner de puertos
- Mapeadores de red
- Escáner de vulnerabilidades

Cualquier información en esta etapa puede dar muchas orientaciones sobre qué tipo de vulnerabilidad puede tener el sistema.

Fase de toma de acceso: permite al intruso tomar posesión de la información o del sistema según las vulnerabilidades encontradas en la fase anterior.

Fase para mantener el acceso: el intruso ejecuta algunas técnicas que le permiten mantener el acceso al sistema ya sea por medio de alguna puerta trasera, la ejecución de código que le permita tomar el acceso remoto en cualquier otro momento, etc.

5.4. Herramientas para realizar auditorías de seguridad

Según Cebrián (et.al) (2018) las principales herramientas con las cuales se puede realizar una prueba de penetración en búsqueda de vulnerabilidades son las siguientes:

NMAP: Es un sistema escáner de puertos, que se ejecuta contra un servidor. Nmap envía paquetes simultáneamente y ping TCP para obtener información de los puertos. La respuesta del ping TCP indica que el servidor de destino está en línea y se puede seguir escaneando. Si el servidor escaneado no responde puede indicar que está fuera de línea o que está protegido por un firewall.

NetCat: Es un escáner de vulnerabilidades para analizar y manipular datos en los protocolos TCP/UDP. Desarrollado originalmente para escanear sistemas Unix. Este software ha sido desarrollado en diferentes versiones a pesar de no tener una base de código mantenible. Actualmente, Netcat tiene versiones para sistemas Windows y varias distribuciones Linux.

WebScarab: Es un sistema que permite ver la información que se envía entre el navegador y el servidor. Con esta herramienta se puede enviar información a una aplicación web como forma de probar la seguridad de la aplicación tomando los paquetes que se envían y modificándolos. Las características generales de esta herramienta son:

- Permite obtener campos ocultos de aplicaciones web para poder editarlos.
- Busca fallo de XSS en las páginas visitadas
- Permite el scripting para modificar aspectos de la aplicación.
- Extrae datos de script de la aplicación analizada.

Nessus: Es un escáner de vulnerabilidades. Permite realizar controles de seguridad en los servidores en los que se requiere verificar su nivel de seguridad. Su lenguaje de programación es NASL (Nessus Attack Scripting Language).

WFuzz: Es un escáner que permite analizar archivos y servidores. También, está diseñada para realizar ataques de fuerza bruta en aplicaciones web. Permite buscar recursos como directorios, servlets y archivos de scripts, a su vez, posibilita realizar diferentes pruebas de inyección de código como SQL injection, XSS, y fuerza bruta en formularios.

OpenVas: Es un escáner de vulnerabilidades muy poderoso, y esta liberada bajo licencia GPL. Esta herramienta está diseñada a partir de una derivación del código de la herramienta Nessus. Posee una arquitectura de cliente servidor, y es precisamente en el servidor donde se ejecuta el procesamiento y la búsqueda de

vulnerabilidades. En la parte del cliente ofrece una interfaz para administrar el proceso de escaneo.

Metasploit Framework: Es un proyecto de código abierto que provee un marco de trabajo para la búsqueda de vulnerabilidades. Ofrece software de pruebas de penetración. Permite la creación de nuevas herramientas de seguridad y módulos para la explotación de vulnerabilidades

En suma, las herramientas aquí presentadas permiten buscar y explorar las vulnerabilidades detectadas en una red, aplicación o servidor. Para realizar una prueba de penetración se debe tener un plan o un orden específico para realizar las exploraciones. Como ejemplo se podría decir que, si se piensa como un atacante en el mundo real, se debe obtener conocimiento del sistema que se desea atacar para saber qué puntos débiles se pueden explotar.

Finalmente, las herramientas que se desean explorar en este documento, para verificar la seguridad de un servidor Debian Linux son las siguientes:

- **NMap:** Permite el escaneo de puertos y que permite tener un conocimiento previo del objetivo a ser atacado.
- **Nessus:** Una vez se conocen los detalles preliminares del objetivo esta herramienta permite conocer las vulnerabilidades y puntos débiles que pueden ser atacados.
- **Metasploit Framework:** Esta herramienta permite la explotación de vulnerabilidades conocidas en los sistemas.

CAPÍTULO II

APLICACIÓN DE PRUEBAS DE PENTESTING

6. Capitulo II: Aplicación de pruebas de pentesting

6.1. Desarrollo de entorno controlado

Las pruebas de penetración involucran la realización de una serie de actividades sobre un sistema o red, con el fin de identificar vulnerabilidades que algún atacante podría aprovechar para robar información o causar daño sobre los sistemas.

Este proceso beneficia a las organizaciones ya que se puede conocer el estado actual de los sistemas y redes, permitiendo que se tomen acciones sobre los puntos débiles evitando que puedan ser aprovechados.

Para la realización de esta guía se tiene configurado un entorno controlado, ya que permitirá la ejecución de las herramientas que en un entorno real no se podrían ejecutar, dado que se podría incurrir en un delito si se realiza sin el consentimiento de la organización o podría causar daños a los sistemas atacados.

La red para las pruebas de penetración, está compuesta por los siguientes elementos:

- 2 servidores Windows 2008
- 1 equipo de escritorio con Windows 10
- 1 equipo con sistema operativo Kali Linux

Para conocer las direcciones IP asignadas a cada equipo se ejecuta el comando “ipconfig” en una terminal para sistemas Windows e “ifconfig” en una terminal para sistemas basados en Linux.

Equipo samsung c
on Windows 10 -> IP: 192.168.0.7

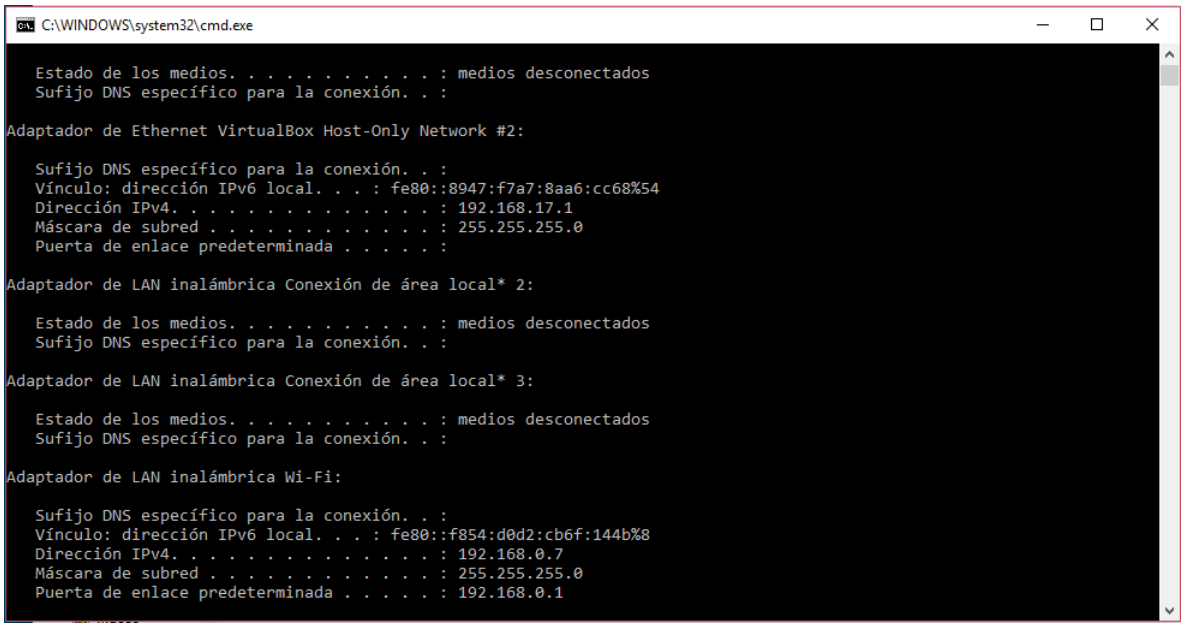


Figura 1: Ip del equipo con windows 10

Equipo Windows Server 2008 1 -> IP: 192.168.0.8

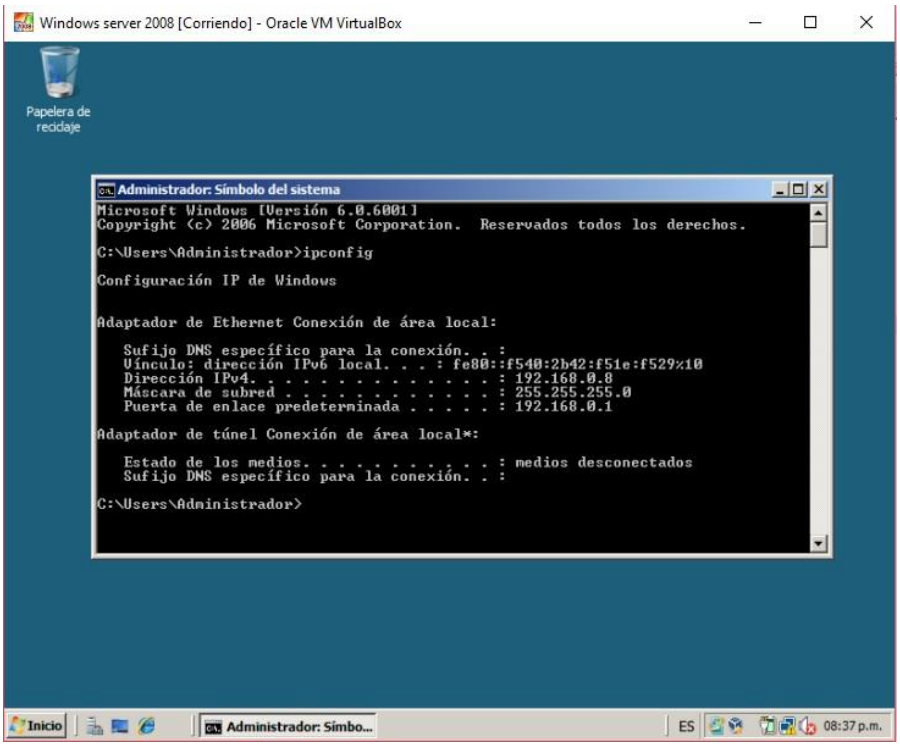


Figura 2: IP Windows Server 2008 I

Equipo Windows Server 2008 2 -> IP: 192.168.0.9

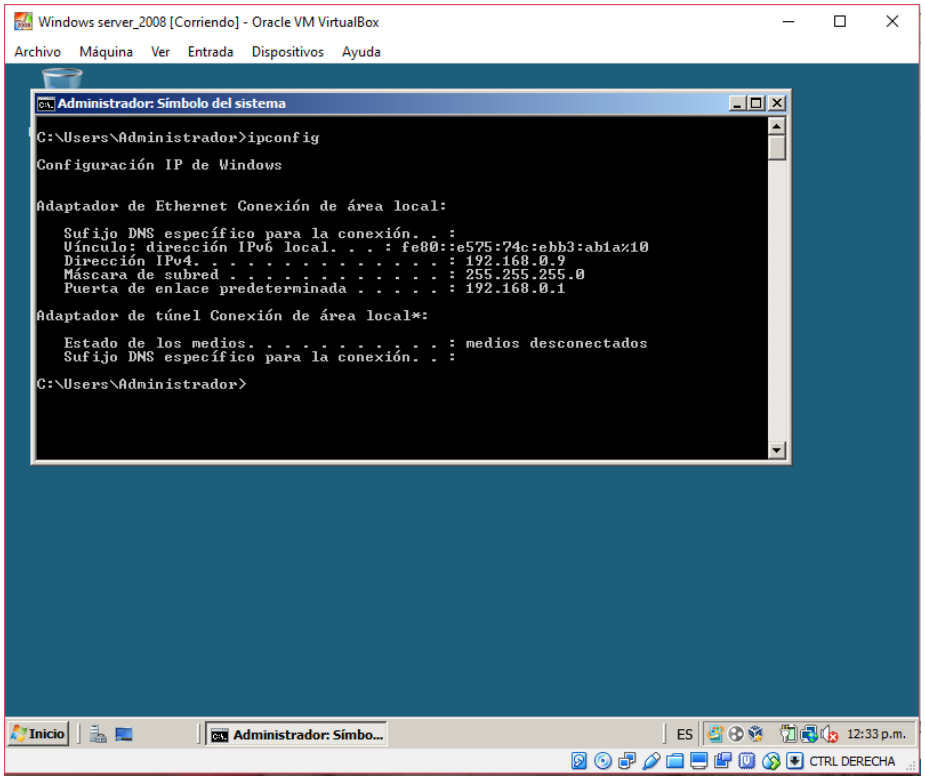


Figura 3: IP Window Server 2008 2

Para sistemas basados en Linux se ejecuta el comando “ifconfig” en la terminal para conocer la IP del equipo.

Equipo Kali Linux -> IP: 192.168.0.11

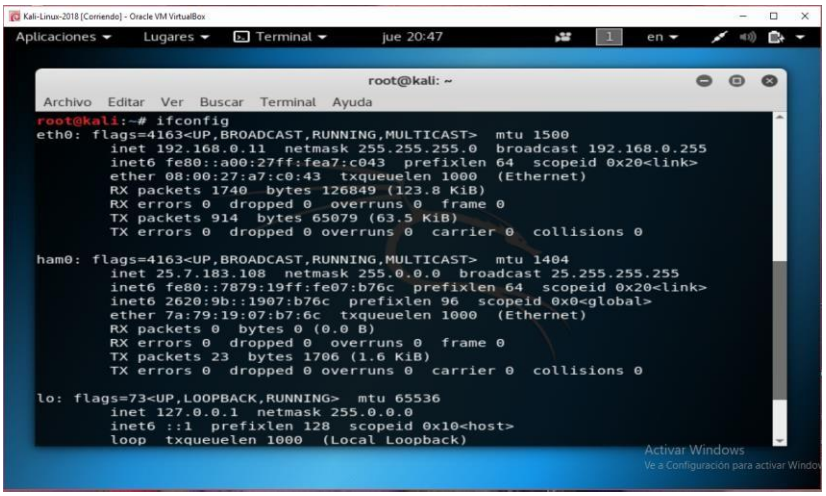


Figura 4: IP del equipo Kali Linux

6.2. Aplicación de pruebas de intrusión

6.2.1. Nmap, escaner de red

El primer paso para la realización de pruebas de penetración es identificar los elementos que conforman una red, para esta actividad se hace necesario usar Nmap. Nmap es una herramienta que permite realizar el escaneo de redes y elementos conectados dentro de la red.

Ahora bien, Kali Linux es un sistema operativo diseñado para realizar test de seguridad. En el caso de Nmap, viene preinstalado como herramienta para realizar escaneo de redes. Luego, se ejecuta el comando "Nmap" para saber si realmente se encuentra instalado, adicionalmente como resultado se despliegan todas las opciones que tiene disponibles para ejecutar.

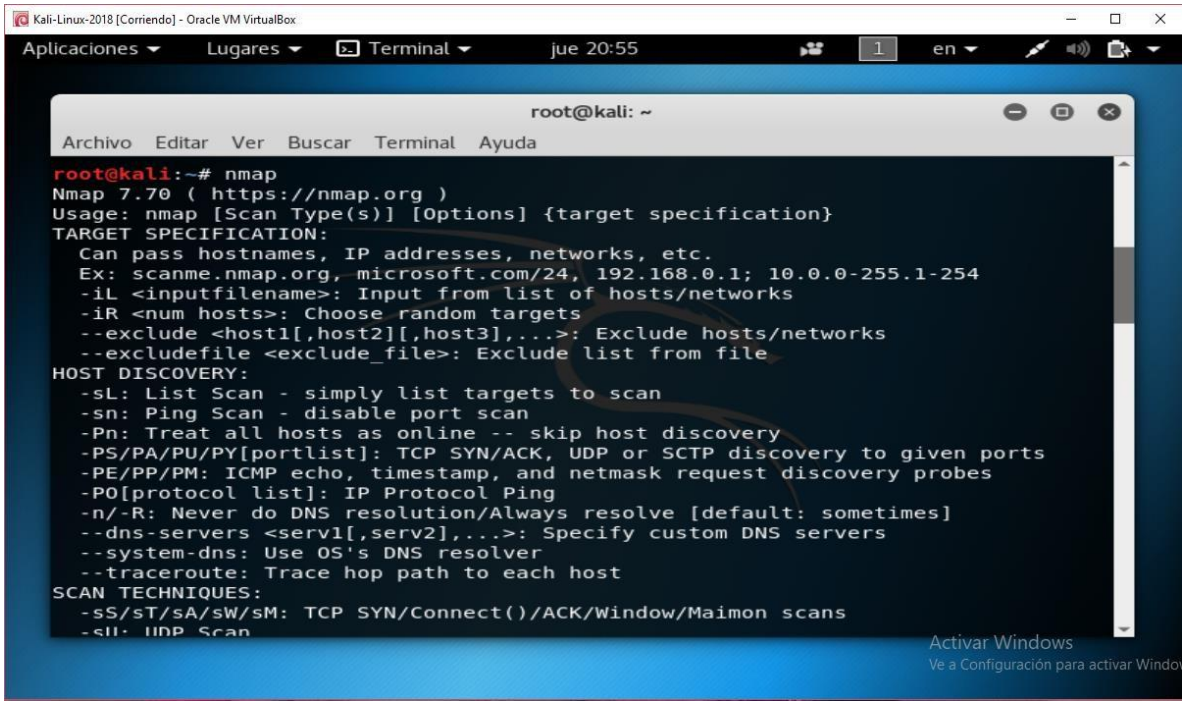


Figura 5: Ejecución Comando en Nmap – Parte 1

En la presente gráfica se evidencian los comandos disponibles que se pueden ejecutar.

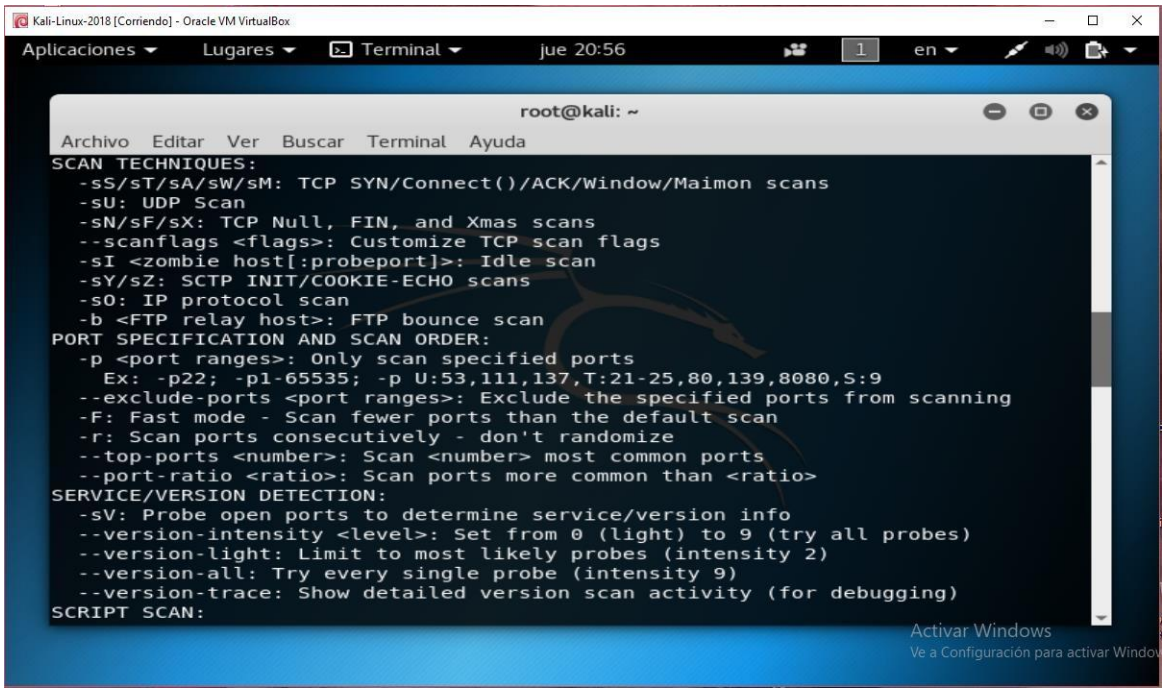


Figura 6: Ejecución comando Nmap – Parte 2

Aquí se exponen los siguientes comandos tales como: comando de escaneo, comando para especificar puertos y comando para detección de versiones.

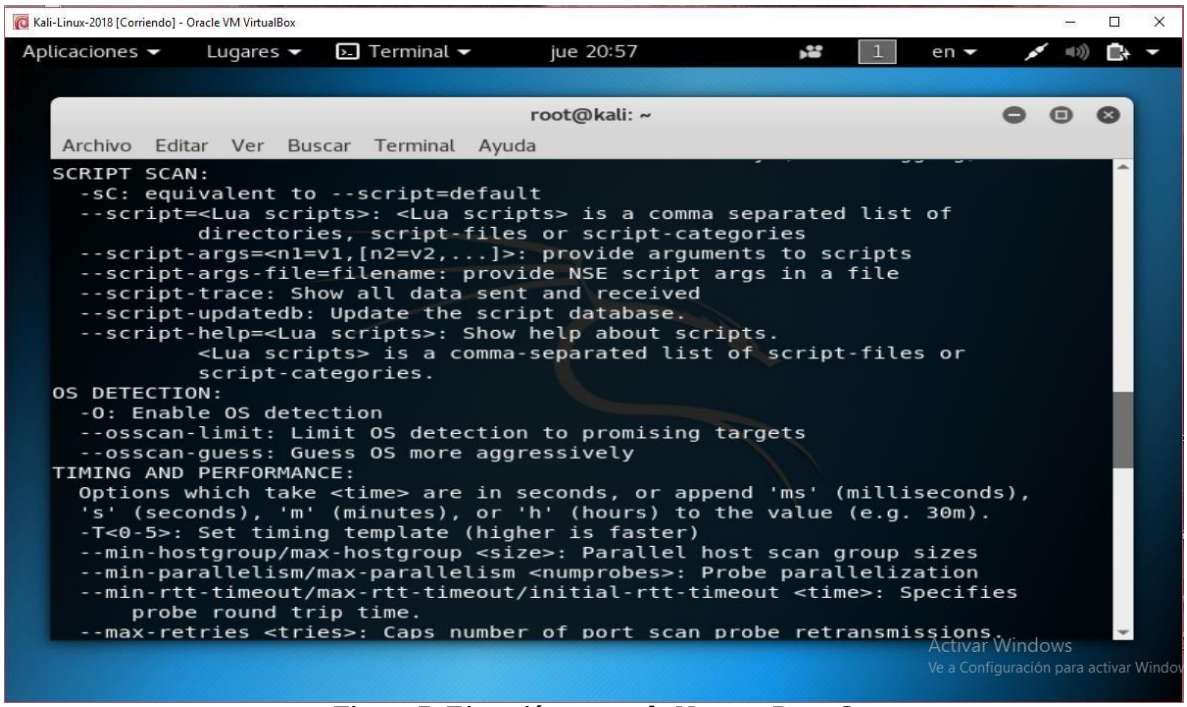


Figura 7: Ejecución comando Nmap – Parte 3

En la figura siete se evidencian los comandos para la detección del sistema operativo, que en este caso es -O.

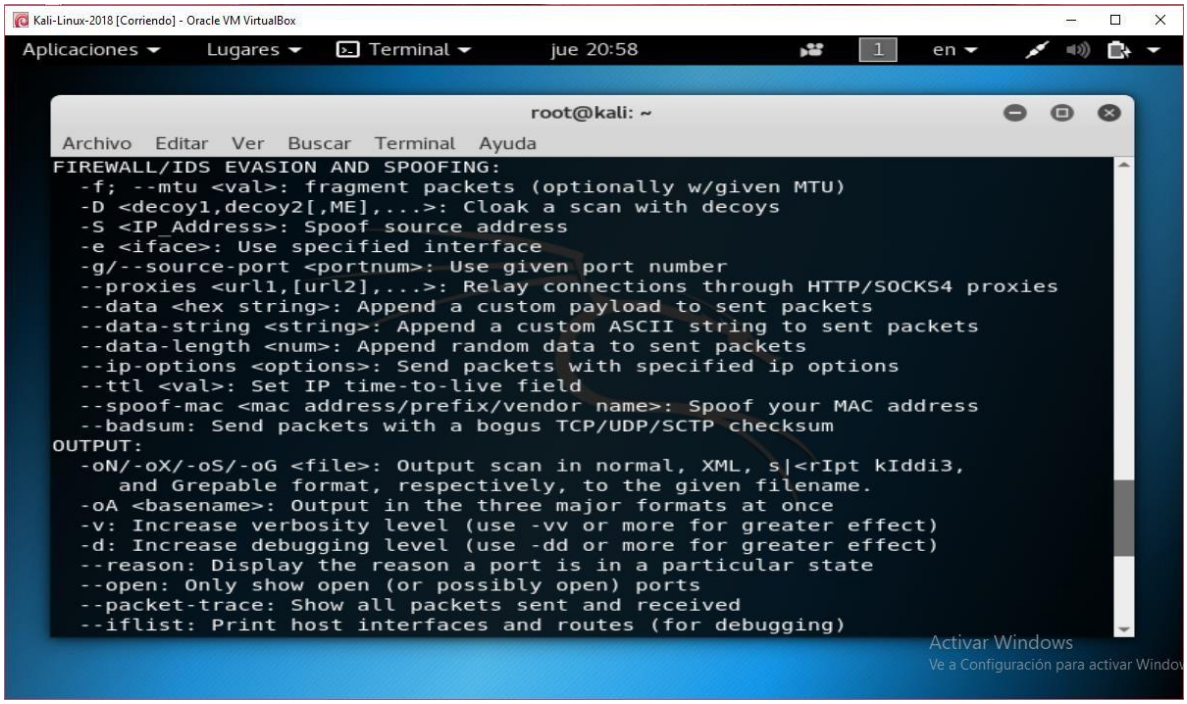


Figura 8: Ejecución comando Nmap – Parte 4

En la presente se evidencian los comandos de escaneo y evasión de *Firewalls.*

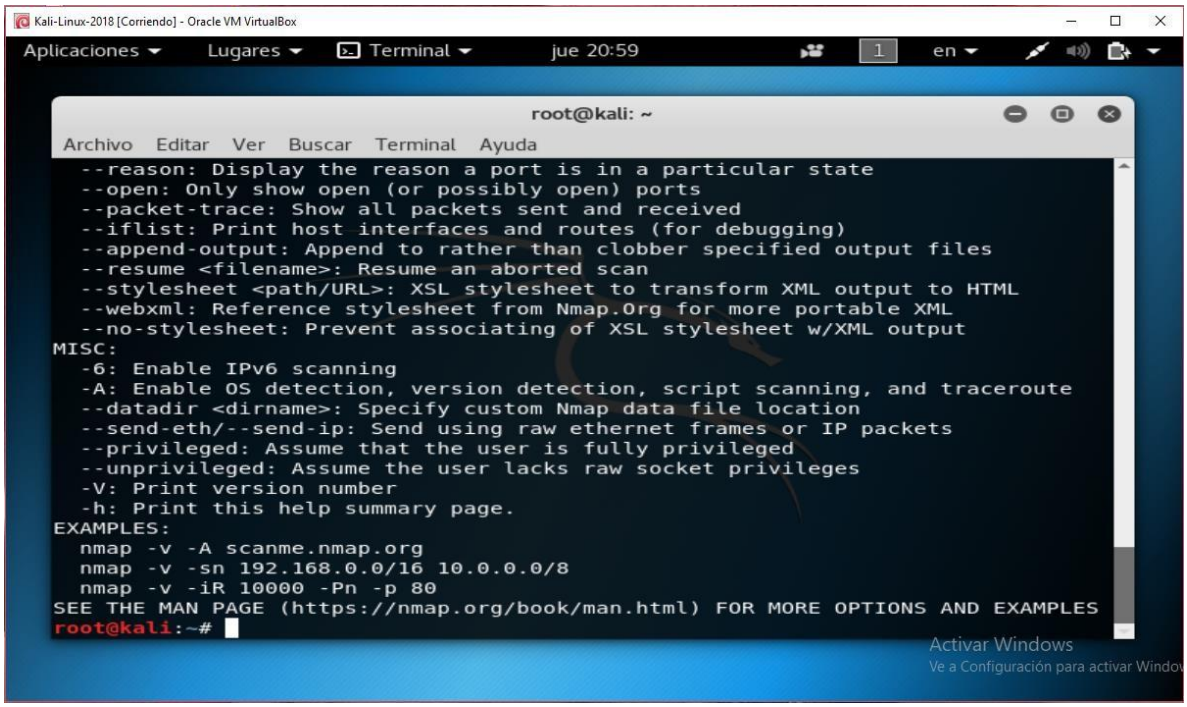


Figura 9: Ejecución comando Nmap - parte 5

Aquí, se exponen los ejemplos sobre la ejecución de los comandos. Entonces, si el usuario de la herramienta desea conocer la versión debe ejecutar el siguiente comando: "nmap -V" .

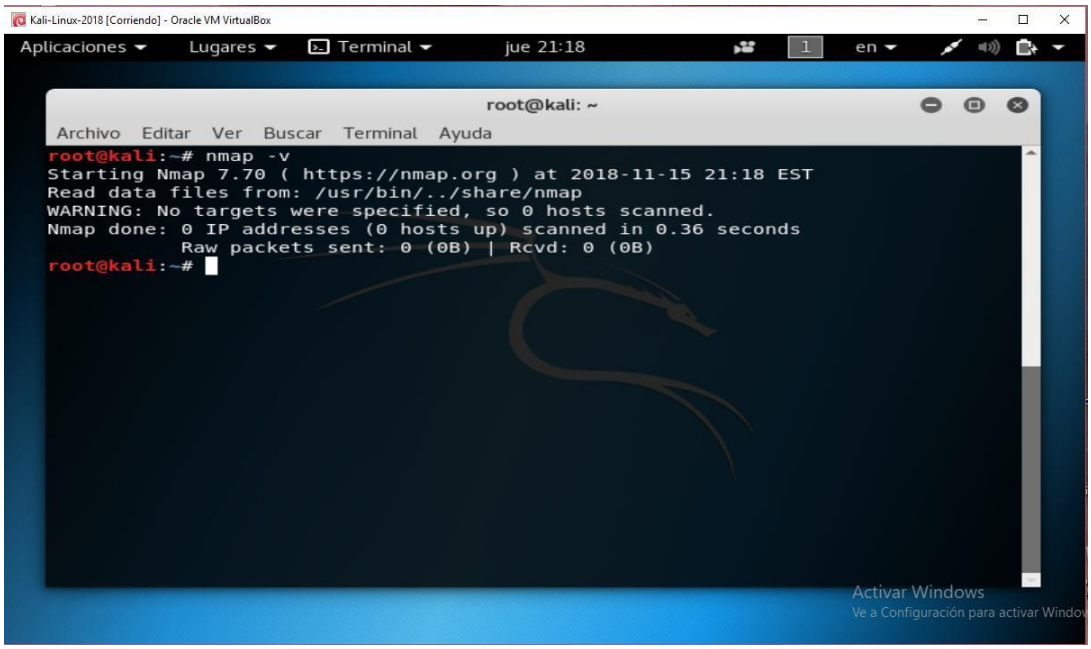


Figura 10: Ejecución Comando -V

En la presente gráfica se expone la ejecución del comando -V con el fin de conocer la versión del Nmap que se encuentra corriendo.

Por otra parte, en una prueba de intrusión lo primero que se debe realizar es el reconocimiento del terreno, en otras palabras conocer el escenario en el que se encuentran los elementos a auditar, entre ellos como; estructura de la red, direcciones IP, equipos. Por el contrario, en Nmap proporciona una gran cantidad de información útil y nos da una idea de lo que podemos encontrar en una red. Seguido de lo anterior, se ejecuta el comando "nmap -sn 192.168.0.0/24" para conocer que equipos están en la red y cuales de ellos están en línea tal y como se presenta en la siguiente imagen.

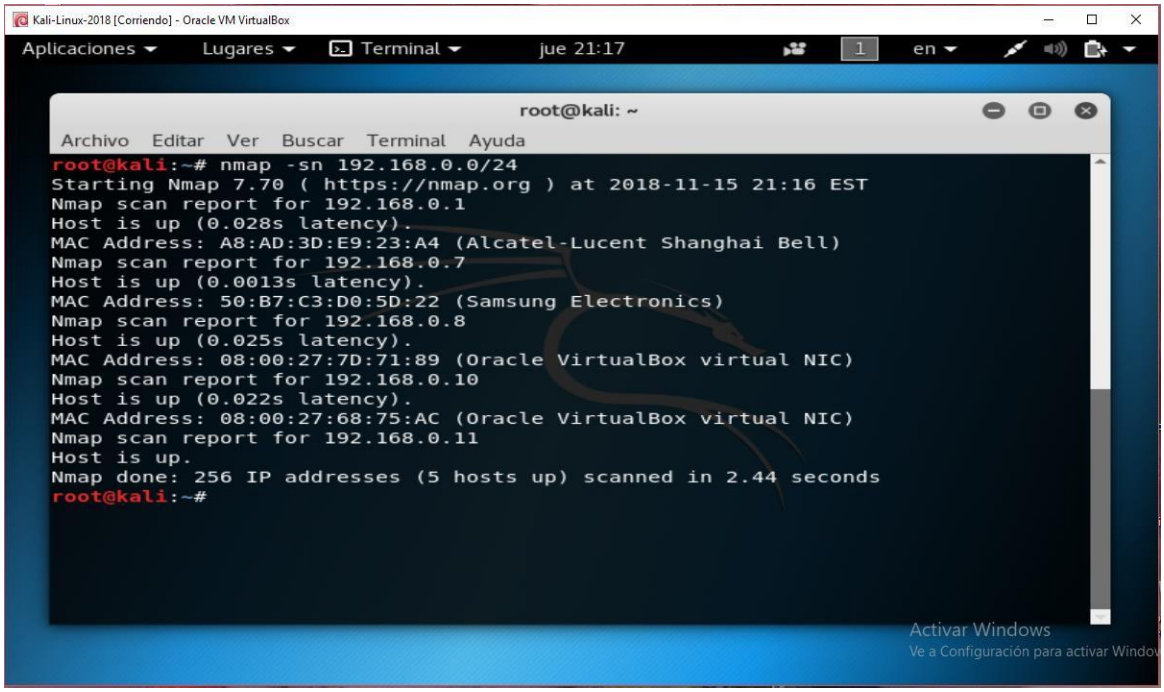


Figura 11: Escaneo general de la red

Entre tanto, una vez termina Nmap de ejecutar el escaneo de la red se pueden ver los resultados de los equipos encontrados:

Namp scan report for 192.168.0.7
Host is up (0.00047s latency)
MAC Address: 50:B7:C3:D0:5D:22 (Samsung Electronics)

Nmap scan report for 192.168.0.8
Host is up (0.0018s latency)

MAC Address: 08:00:27:7D:71:89 (Oracle VirtualBox Virtual NIC) Equipo con Windows server 2008

Nmap scan report for 192.168.0.9
Host is up (0.00064s latency)
MAC Address: 08:00:27:68:75:AC (Oracle VirtualBox Virtual NIC) Equipo con Windows server 2008

Nmap scan report for 192.168.0.11
Host is up
Equipo Kali Linux.

Nmap done: 256 IP addresses (5 hosts up) scanned in 2.39 seconds.

Por ende, los resultados indican que en la red se tienen 4 equipos en funcionamiento. Hasta el momento se tiene solo información básica sobre que equipos conforman la red.

Para el siguiente paso se selecciona la maquina con IP "192.168.0.8". Lo siguiente consiste en realizar un escaneo para conocer los puertos abiertos que tiene la máquina. Para esto se debe ejecutar el comando "nmap -open 192.168.0.8"

Los resultados que arroja el comando son los siguientes:

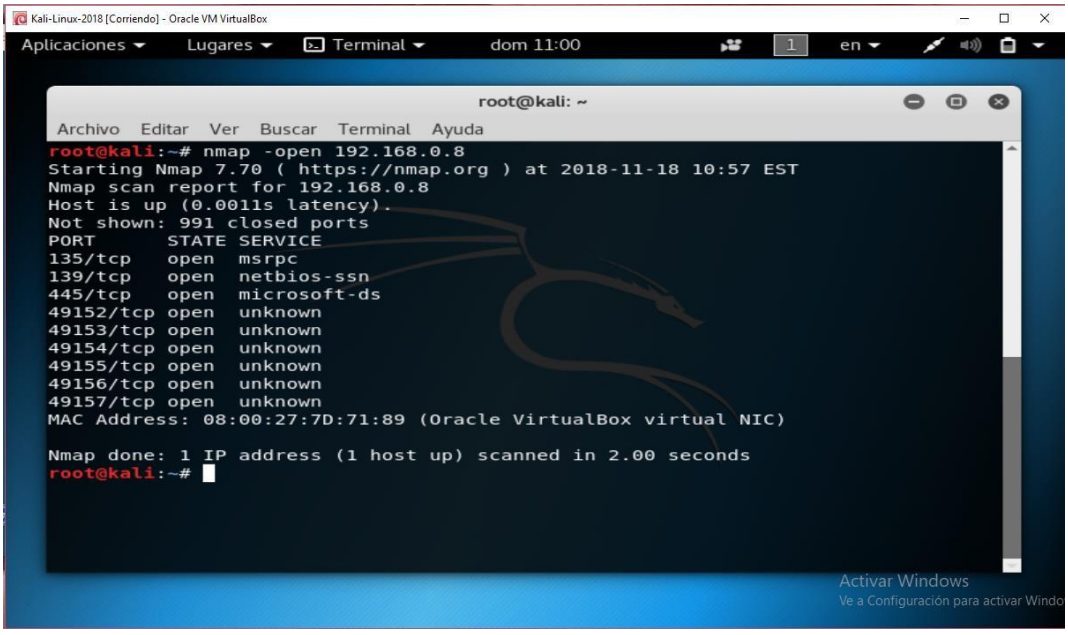


Figura 12 Escaneo de puertos abiertos

```
Nmap scan report for 192.168.0.8
Host is up (0.0011s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
135/tcp open  msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
49152/tcp open unknown
49153/tcp open unknown
49154/tcp open unknown
49155/tcp open unknown
49156/tcp open unknown
49157/tcp open  unknown
MAC Address: 08:00:27:7D:71:89 (Oracle VirtualBox virtual NIC)
```

Para el sistema operativo Windows Server 2008 escaneado se encuentran 9 puertos abiertos. Nmap reconoce algunos de los servicios que se exponen por algunos puertos abiertos.

En el puerto 135/tcp se expone el servicio msrpc. Msrpc es un servicio de Microsoft que permite el acceso y control de un sistema windows en modo remoto. Esto puede ser un hueco en la seguridad del equipo ya que si se pudiera explotar alguien externo podría posiblemente ingresar al sistema.

En el puerto 139/tcp se expone el servicio netbios-ssn. Netbios-ssn es un servicio de Microsoft que permite que los equipos de una red puedan compartir archivos e impresoras.

En el puerto 445/tcp se expone el servicio microsoft-ds. Este servicio se utiliza para el acceso a redes TCP/IP sin que se necesite la capa netbios-ssn. Este servicio se implementó en las versiones de Windows xp en adelante. Este puerto puede exponer el sistema a varios ataques de gusano.

Los demás puertos expuestos que se encuentran abiertos usan el protocolo tcp el cual es un protocolo que se usa para el envío de paquetes, garantizando su estado al momento de la entrega. El protocolo TCP se ocupa de garantizar la comunicación.

La importancia de realizar el escaneo de puerto radica es el descubrimiento de servicios con vulnerabilidades conocidas, servicios que pueden estar obsoletos y que ponen en riesgo la seguridad de un equipo y por tanto de la red completa.

Ahora escanear equipo por equipo en una red grande puede ser una actividad muy compleja y que manualmente requeriria de mucho tiempo. Nmap permite

escanear un rango de puertos dentro de la red en la que se encuentran los equipos, de esta forma nmap expondrá el estado de los puertos escaneados en cada una de las máquinas, y los servicios que se exponen a través de cada puerto.
Para esto usamos el comando "nmap -p1-1023 192.168.0.0/24".

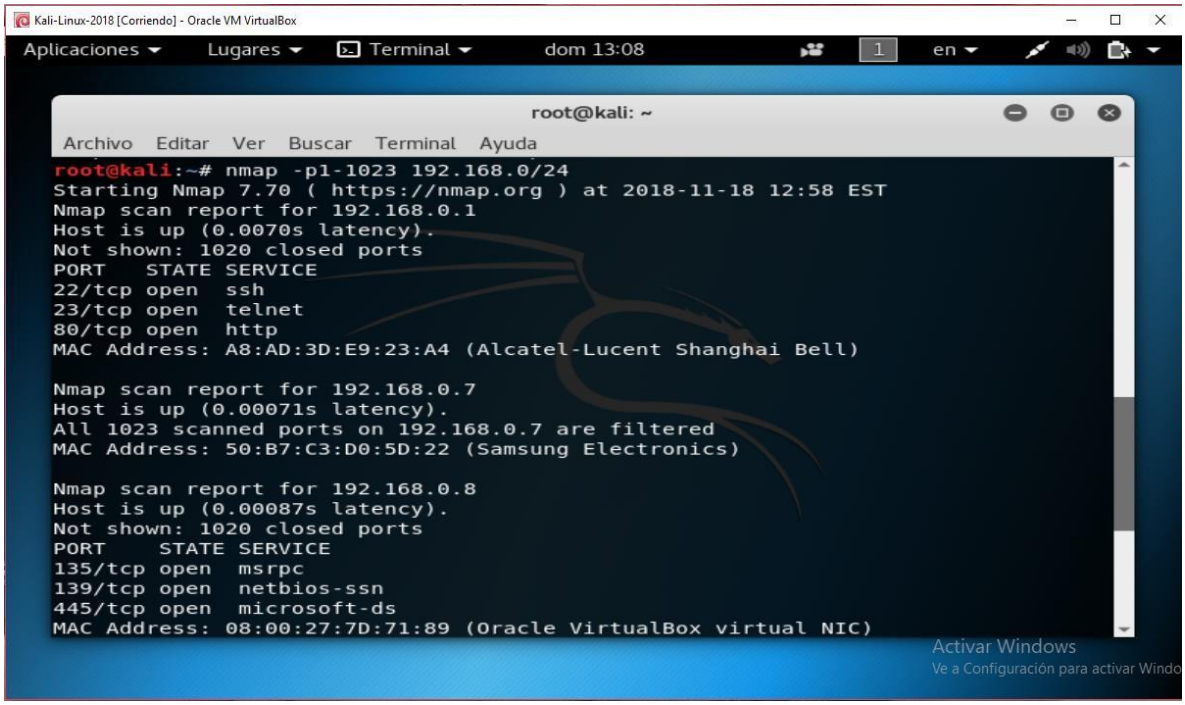


Figura 13 Escaneo de puertos en la red – 1

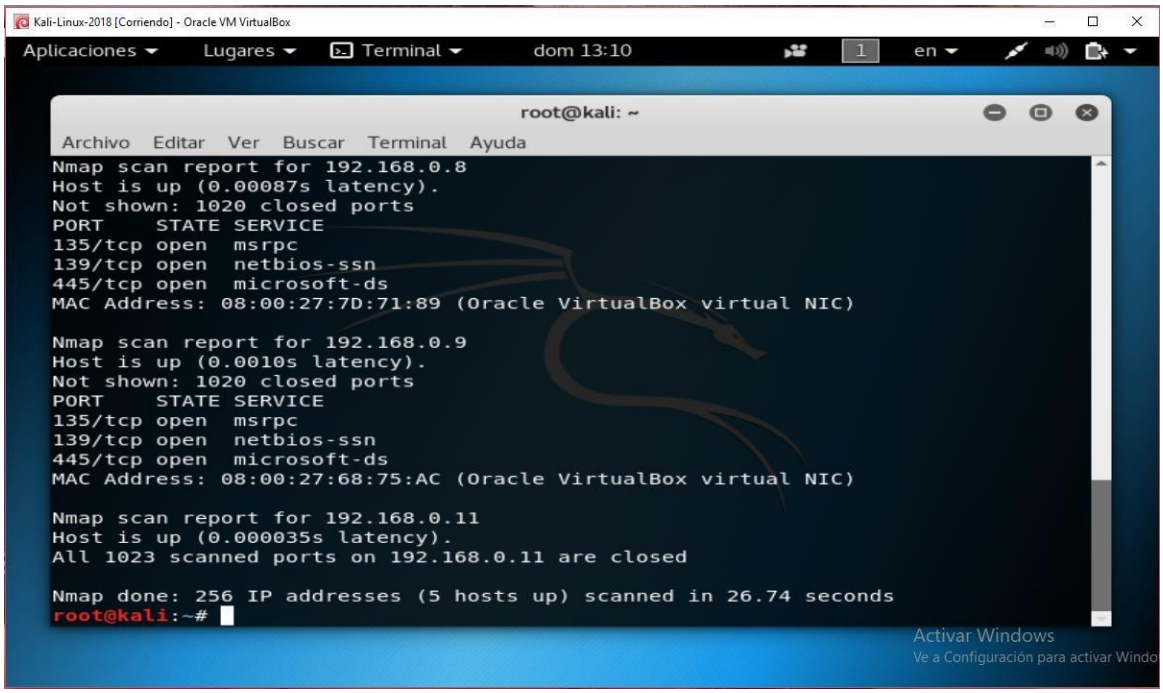


Figura 14 Escaneo de puertos en la red - 2

El resultado nos muestra los puertos abiertos en cada uno de los equipos, y los servicios conocidos que se ejecutan en dicho puerto.

Hasta este punto nmap ha mostrado información básica de los puertos que ha encontrado abiertos en cada uno de los equipos de la red o en uno especifico de la red que se haya indicado. Nmap puede realizar un escaneo un poco más detallado con el fin de detectar el servicio y la versión que esta corriendo.

Para realizar el escaneo más detallado debemos ejecutar el comando: "nmap -sV -T4 192.168.0.8".

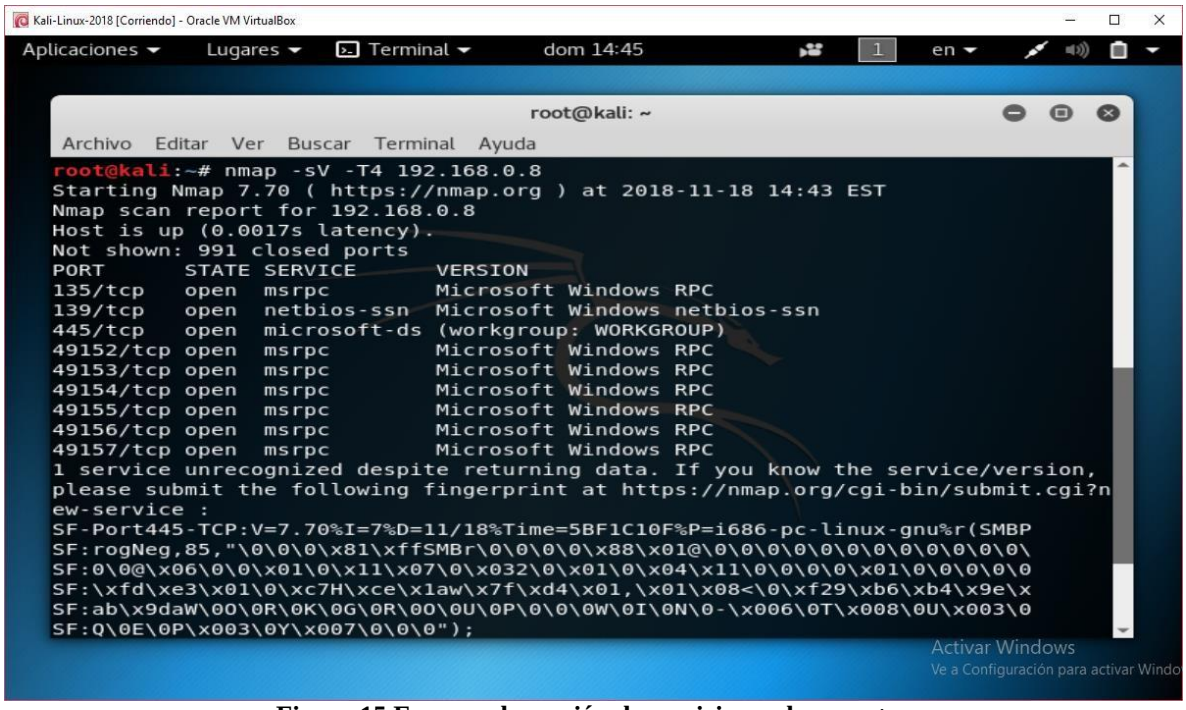


Figura 15 Escaneo de versión de servicios en los puertos

Los resultados indican que se encontró algunas versiones de los servicios en los puertos abiertos en el servidor Windows Server 2008

PORT STATE SERVICE VERSION
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
445/tcp open microsoft-ds (workgroup: WORKGROUP)
49152/tcp open msrpc Microsoft Windows RPC
49153/tcp open msrpc Microsoft Windows RPC
49154/tcp open msrpc Microsoft Windows RPC
49155/tcp open msrpc Microsoft Windows RPC
49156/tcp open msrpc Microsoft Windows RPC
49157/tcp open msrpc Microsoft Windows RPC

La segunda parte del resultado del escaneo está indicando que nmap no está seguro del servicio que se está ejecutando en ese puerto, por lo cual está preguntando si el servicio es conocido, de ser así nmap da la posibilidad de envíar un reporte junto con una huella digital que se genera con el fin de mejorar la detección de servicios a futuro.

1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :

SF-Port445-TCP:V=7.70%I=7%D=11/18%Time=5BF1C26B%P=i686-pc-linux-gnu%r(SMBP
SF:rogNeg,85,"\0\0\0\x81\xffSMBr\0\0\0\0\x88\x01@\0\0\0\0\0\0\0\0\0\0\0\
SF:0\0@\x06\0\0\x01\0\x11\x07\0\x032\0\x01\0\x04\x11\0\0\0\0\x01\0\0\0\0\0
SF:\xfd\xe3\x01\0\xec\xdb\x80\xeaw\x7f\xd4\x01,\x01\x08<\0&\xed\xaaO\x20\x
SF:dds\xb9W\0O\0R\0K\0G\0R\0O\0U\0P\0\0\0W\0I\0N\0-\x006\0T\x008\0U\x003\0
SF:Q\0E\0P\x003\0Y\x007\0\0\0");

Adicionalmente nmap tiene un comando que permite intentar conocer que sistema operativo está corriendo la máquina.
Esto puede ayudar a conocer que versiones y que vulnerabilidades tienen algunos servicios según el sistema operativo que está corriendo la máquina.

Se ejecuta el comando "nmap -vS -T4 192.168.0.8 para escanear el sistema operativo.

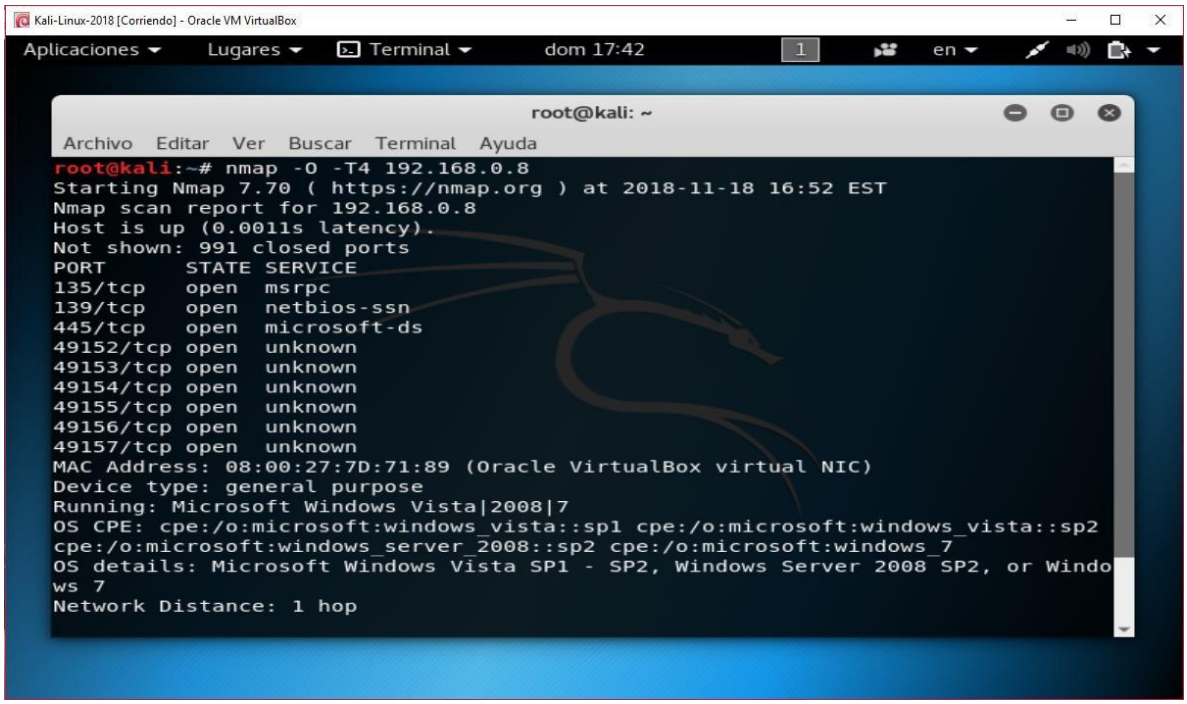


Figura 16 Escaneo de versión del sistema operativo

Device type: general purpose
Running: Microsoft Windows Vista|2008|7
OS CPE: cpe:/o:microsoft:windows_vista::sp1 cpe:/o:microsoft:windows_vista::sp2 cpe:/o:microsoft:windows_server_2008::sp2 cpe:/o:microsoft:windows_7
OS details: Microsoft Windows Vista SP1 - SP2, Windows Server 2008 SP2, or Windows 7

Nmap intenta predecir qué sistema operativo está corriendo en la máquina escaneada, en este caso no hay una certeza del 100% pero según las respuestas del sistema sabe que es un sistema operativo Windows, posiblemente Vista, Windows server 2008 o Windows 7.

Esto da muchas pistas sobre qué tipo de vulnerabilidades podría tener la versión del sistema operativo descubierta.

6.2.2. Escaner de vulnerabilidades (Nessus Vulnerability Scanner)

Según la documentación oficial Nessus es un escáner de vulnerabilidades, que permite conocer qué tipo de vulnerabilidades ponen en riesgo un sistema. Con esta información se podrán tomar las medidas de precaución para evitar diferentes ataques según el tipo de vulnerabilidad.

Nessus tiene una variedad de componentes tecnológicos que permiten realizar un test de vulnerabilidades a distintos tipos de sistemas como: Sistemas operativos, dispositivos de red, Hipervisores, bases de datos y servidores web.

6.2.2.1. Agentes de Nessus

Según la documentación es un programa liviano que ayuda a realizar los escaneos de vulnerabilidades dentro de una red. Este tipo de programa recopila datos de los escaneos realizados centralizándolos para su posterior análisis. Según la documentación oficial de Nessus en la mayoría de los casos no se hace necesario tener credenciales de usuario para realizar los escaneos en los equipos y detectar vulnerabilidades.

6.2.2.2. Características de los agentes de Nessus

La documentación oficial de Nessus detalla las siguientes características:

- Identificar vulnerabilidades.
- Detectar fallas de seguridad que puedan ser utilizadas por programas como malwares.
- Es capaz de detectar malas configuraciones en diferentes tipos de servidores. Estas configuraciones por defecto generan fallos en seguridad que aprovechan los programas maliciosos para obtener acceso a los sistemas o datos confidenciales.

- Los agentes de Nessus pueden realizar verificaciones de las políticas de seguridad o de acceso a datos dentro de una red o un equipo.

Según la documentación oficial de Nessus se tienen varias ventajas al momento de implementarlo en una red o sistema.

- La instalación se puede realizar en cualquier entorno (Equipos físico, máquinas virtuales y plataformas en la nube).
- Su ejecución requiere de un mínimo de recursos de los equipos en los que se encuentra instalado.
- Recopila información sobre vulnerabilidades encontradas y estas son enviadas a los servidores centrales de Tenable para su posterior análisis.

Según la documentación oficial Nessus se instala y se administra mediante HTTPS y SSL y usa el puerto 8834. La instalación de Nessus usa por defecto un certificado autofirmado.

Para que Nessus use todas sus capacidades es necesario configurar los firewalls de terceros o propio del sistema para que admita las conexiones desde las IP's en donde se encuentra instalado Nessus.

Según la documentación oficial de Nessus, este puede realizar exploraciones en sistemas basados en Ipv6. Al menos una interfaz de Ipv6 debe estar configurada en el entorno donde se encuentra instalado Nessus.

La herramienta Nessus no viene instalada por defecto en Kali Linux, por lo que se debe descargar desde la página oficial, e instalar en el sistema. La versión más actual de Nessus al momento de escribir esta monografía es la 8. El archivo descargado se nombra de la siguiente forma: Nessus-8.0.1-debian6_i386.deb

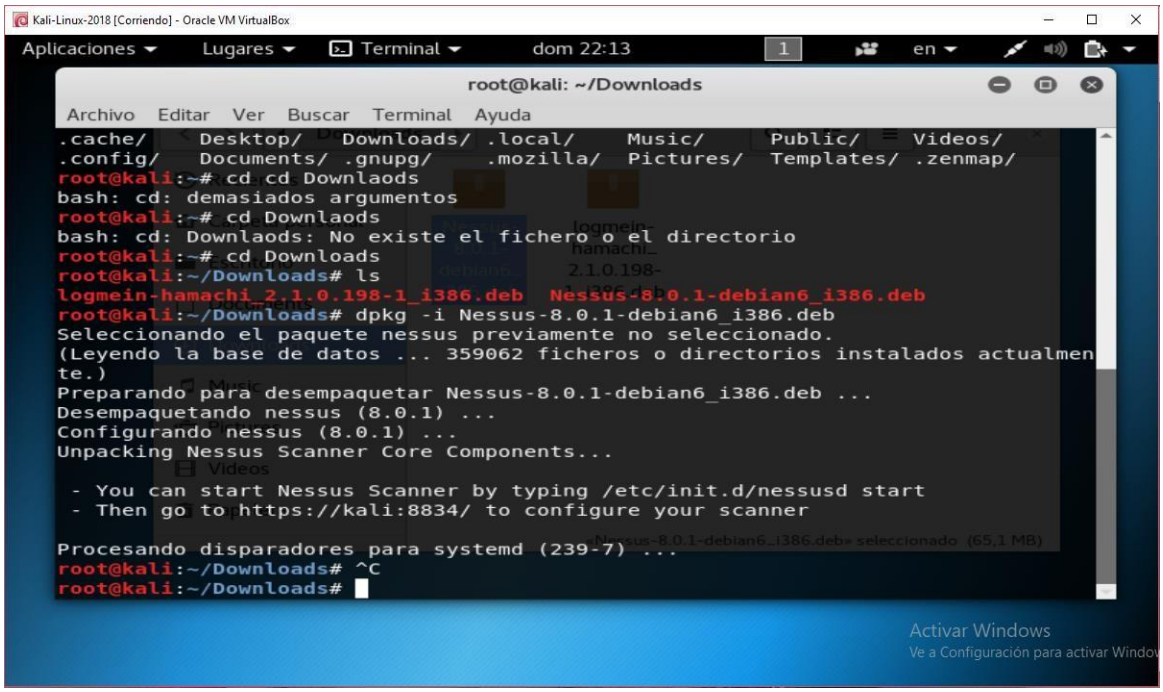


Figura 17 Instalación no de Nessus

Como el instalador tiene extensión .deb se debe usar el comando dpkg para ejecutar la instalación: # dpkg -i Nessus-8.0.1-debian6_i386.deb
Después de haber realizado la instalación se puede iniciar Nessus con el siguiente comando:

/etc/init.d/nessusd start

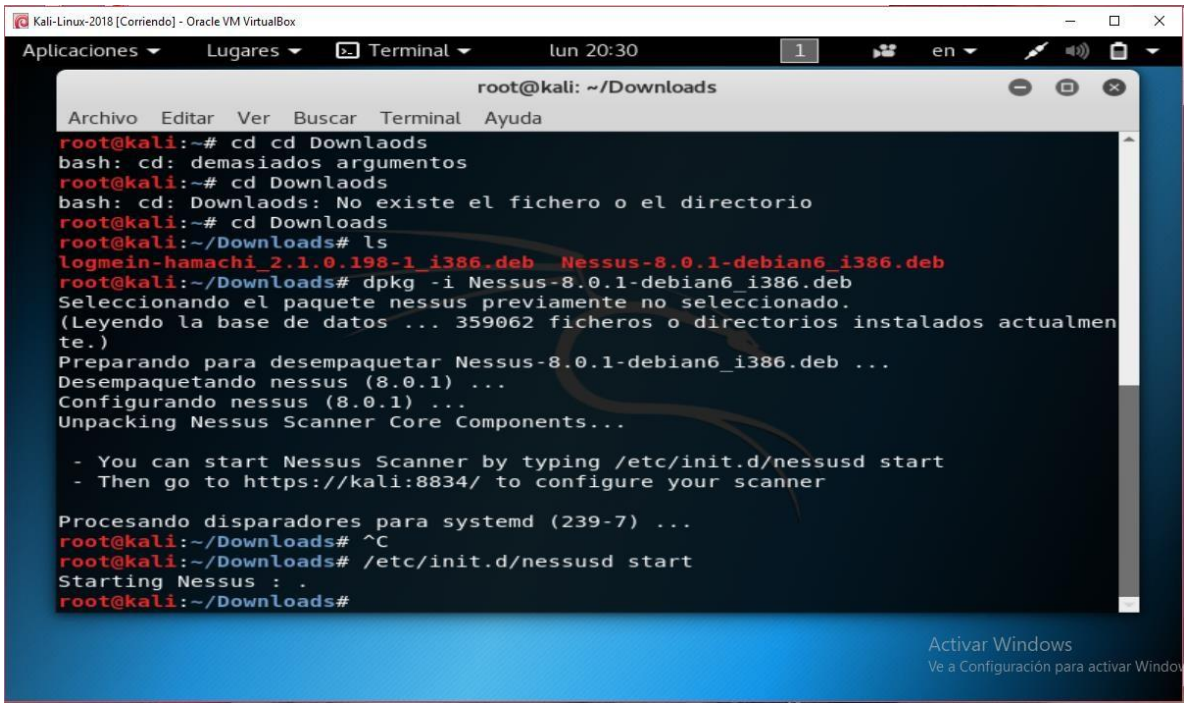


Figura 18 Comando para iniciar Nessus

Una vez se ha iniciado Nessus es necesario configurarlo. Para hacerlo se debe ingresar a la dirección https://kali:8834 desde el navegador por defecto que se tenga instalado en Kali Linux.

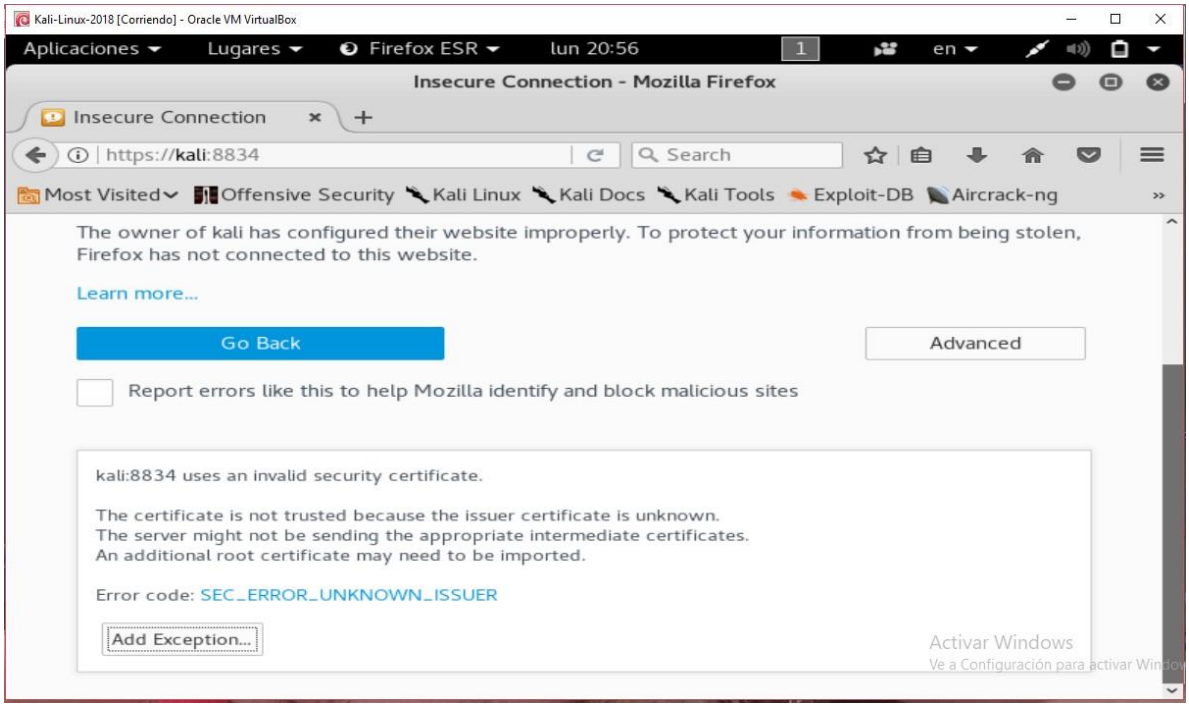


Figura 19 Ingreso a https://kali:8834

Al ingresar al navegador mostrara un mensaje de advertencia puesto que el emisor del certificado es desconocido.

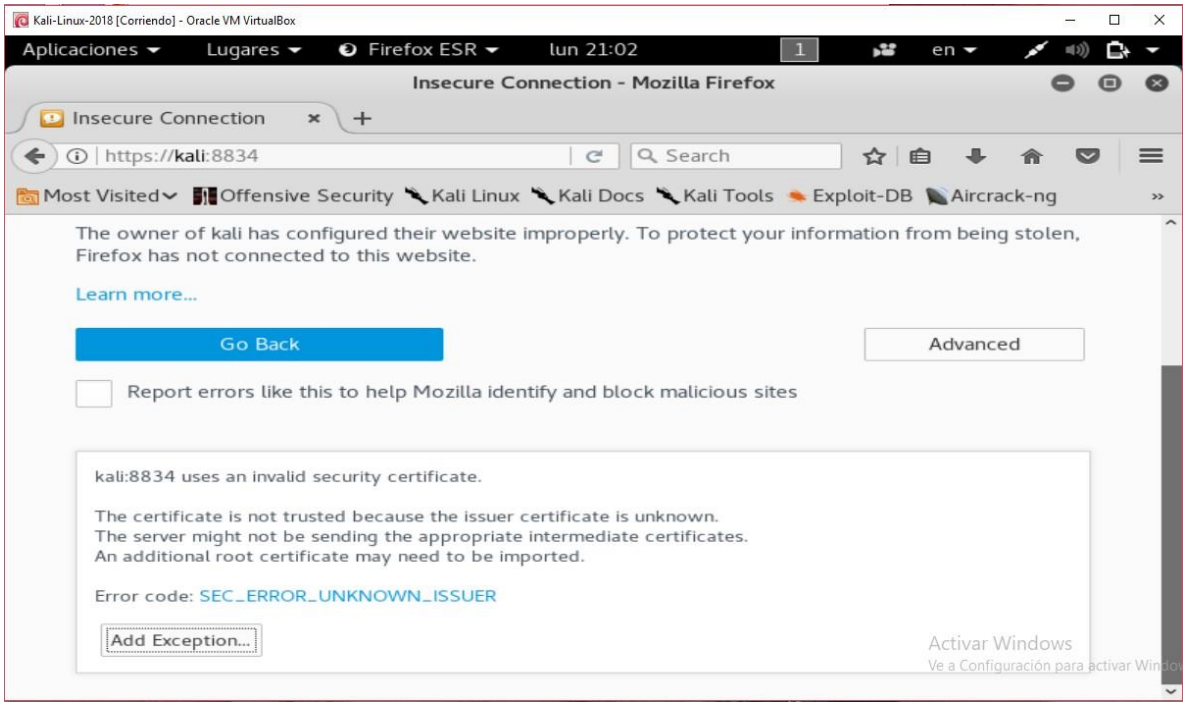


Figura 20 Advertensia del navegador sobre el sitio

Se crea la excepción en las opciones avanzadas para poder ingresar.

Se realiza la confirmación de la excepción de seguridad.

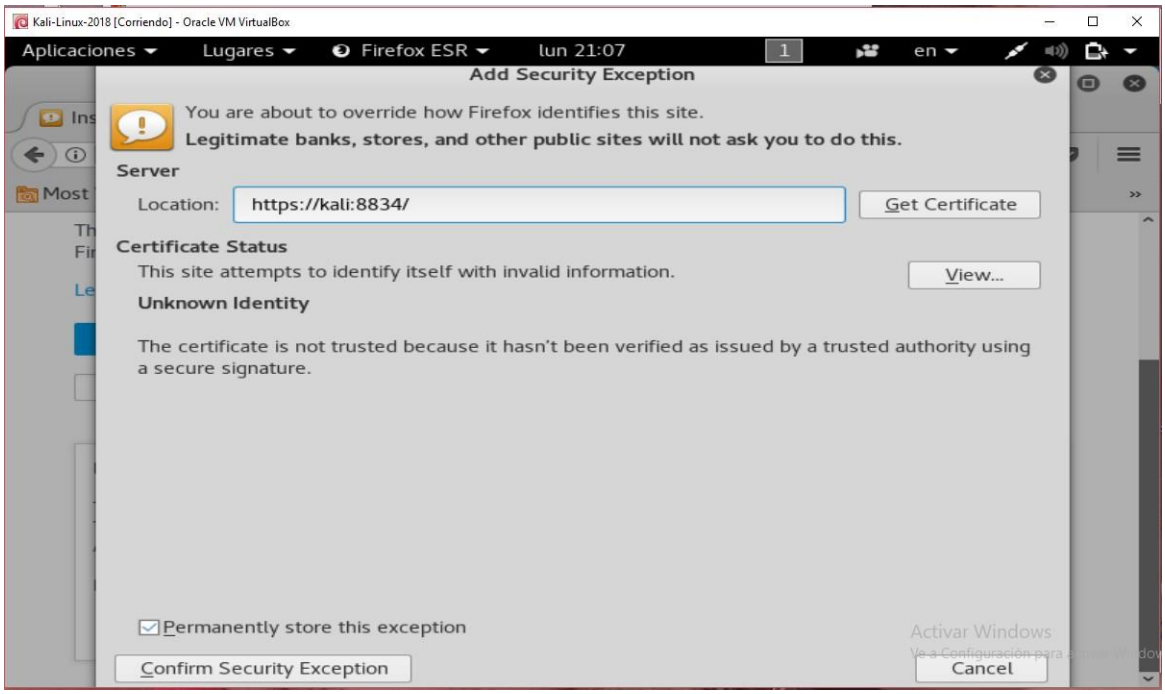


Figura 21 Confirmación de la excepción de seguridad

Una vez se confirma la excepción de seguridad se carga la página para crear una cuenta en Nessus. Esta cuenta de usuario administrador tendrá la facultad para crear, eliminar usuarios y detener escaneos que estén en ejecución. Para registrar Nessus se debe crear una cuenta en la página de Nessus y solicitar un código de activación. Este código de activación llegará al correo con el que se registre la cuenta de usuario.

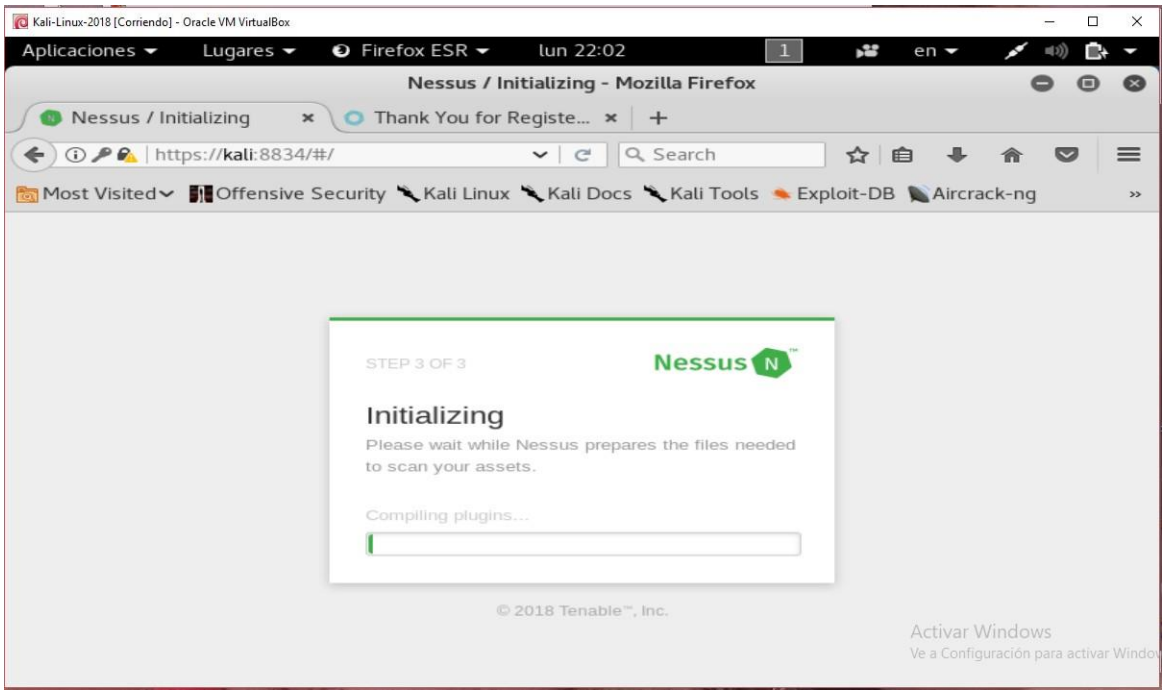


Figura 22 Compilación de plugins de Nessus

Una vez se tenga el código se puede activar Nessus, y comenzar a descargar los plugins necesarios para realizar los escaneos sobre los equipos de una red o un equipo en especial.

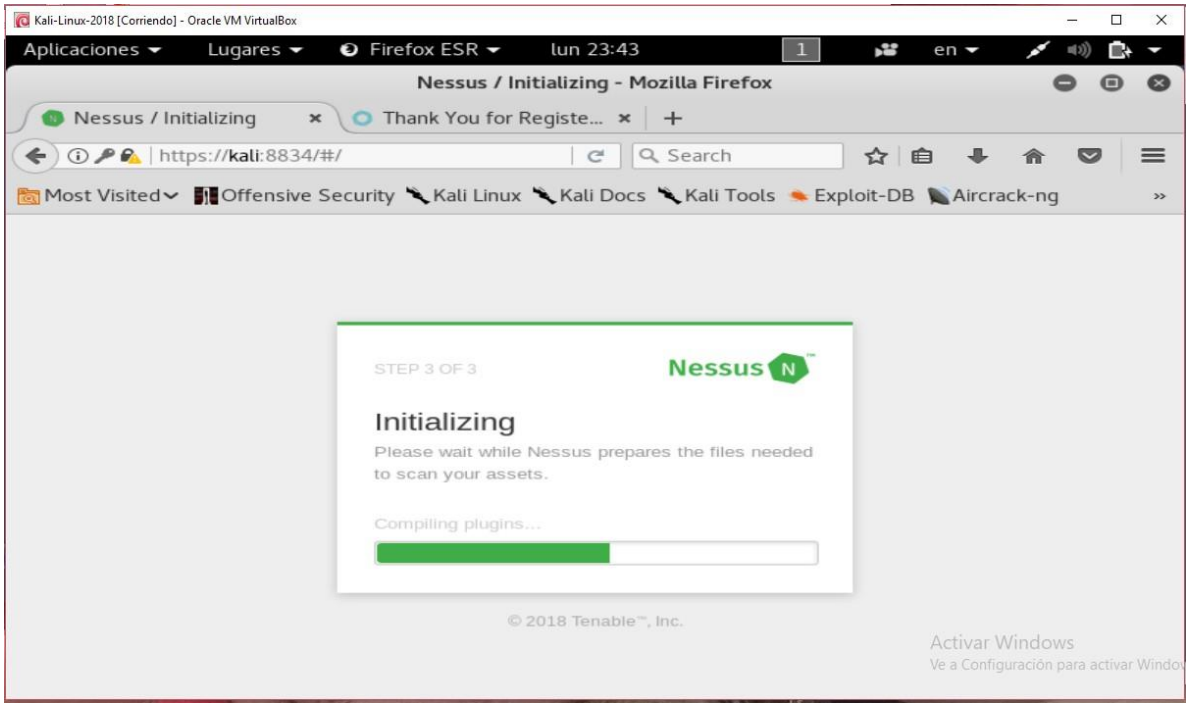


Figura 23 Compilación de plugins de Nessus - 2

Una vez se finaliza la compilación de los plugins de Nessus, este ya está listo para iniciar el analisis de vulnerabilidades sobre un equipo objetivo.

La interfaz de la herramienta se muestra en la siguiente imagen:

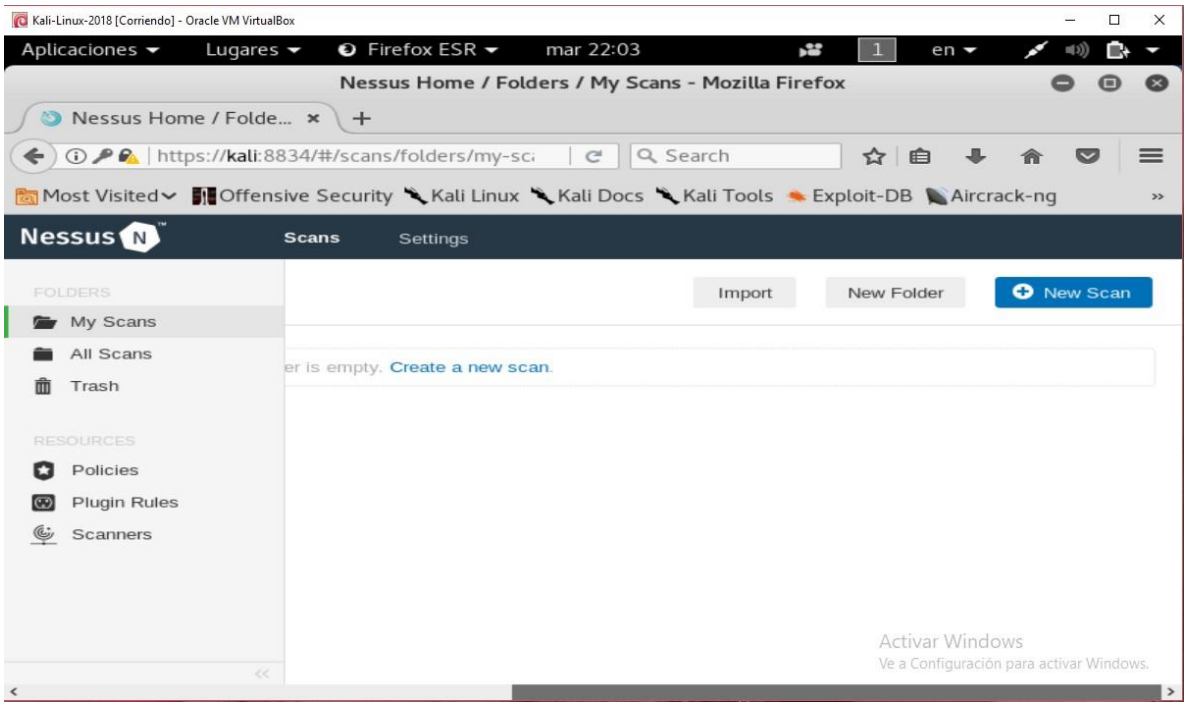


Figura 24 Interfaz de Nessus

Nessus es una poderosa herramienta para realizar escaneo de vulnerabilidades de cualquier tipo sobre un sistema objetivo. Ahora para empezar se debe seleccionar una de las políticas preconfiguradas que trae Nessus, cada política está categorizada por su utilidad o enfoque, ya que cada una de las plantillas trabaja un tipo de vulnerabilidad especial. La versión más actualizada de Nessus permite incluso escanear equipos en busca de vulnerabilidades tipo spectre o meltdown, un tipo de vulnerabilidad que afecta algunos modelos de procesadores Intel y AMD y que fue descubierto en 2018.

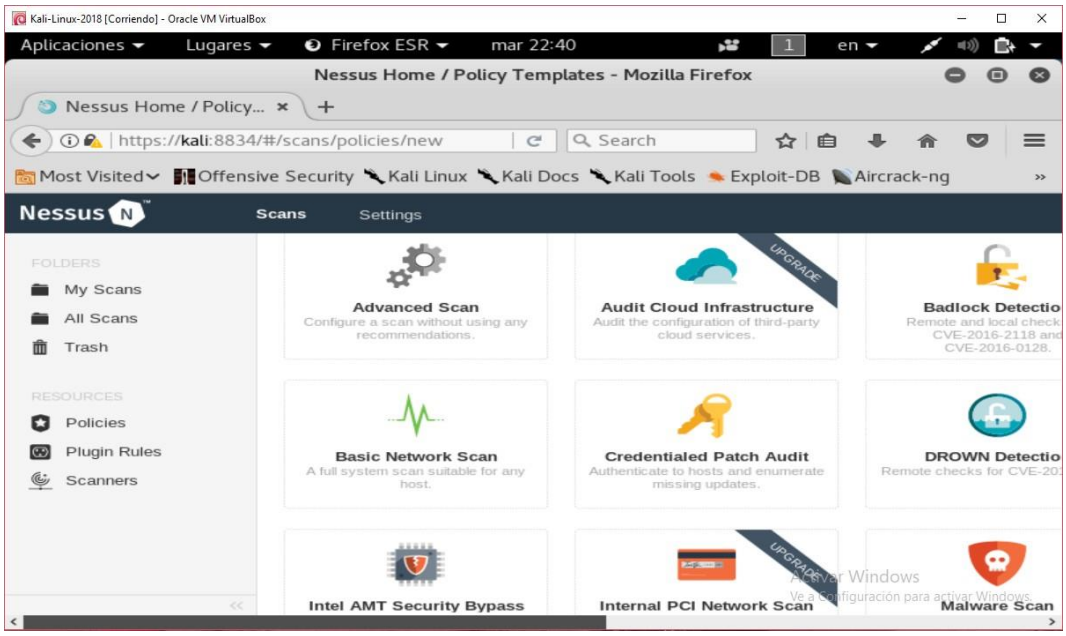


Figura 25 Plantillas de Nessus

Para iniciar el escaneo de vulnerabilidades se da clic sobre el enlace “My scans” y se selecciona la plantilla para verificar huecos de seguridad para malwares y programas maliciosos.

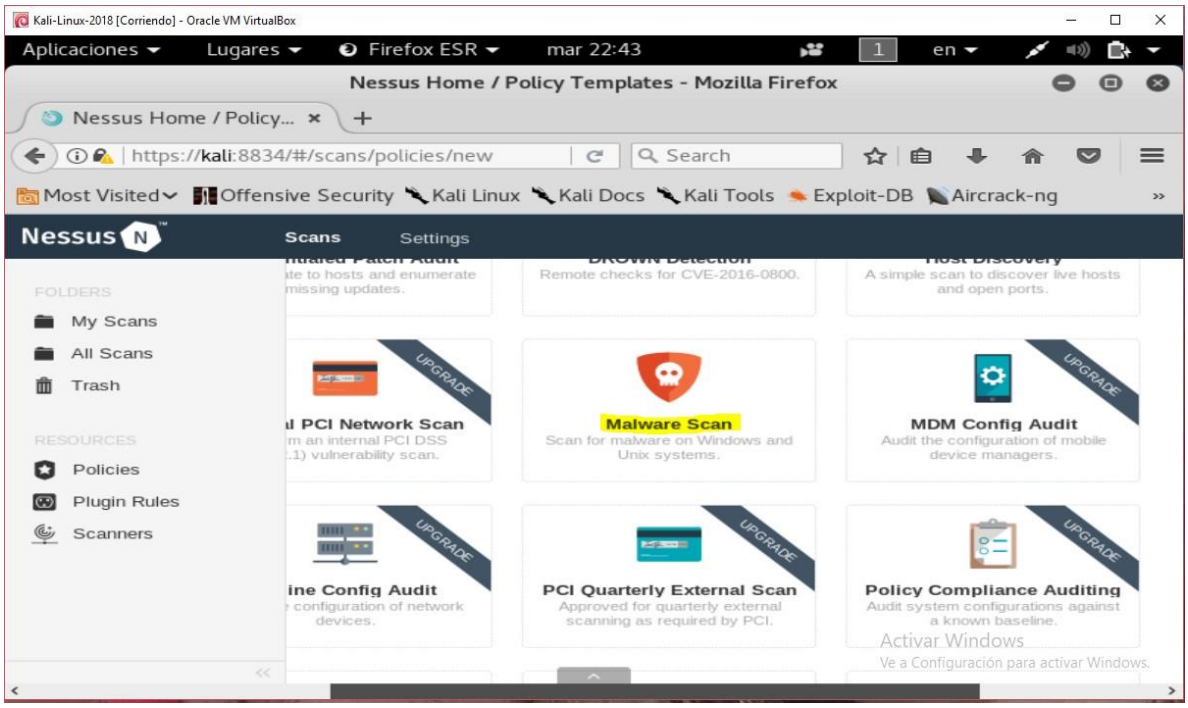


Figura 26 Plantilla para escáner de malware

Luego de seleccionar la plantilla para el escáner de vulnerabilidades, se personaliza con un nombre y una descripción.

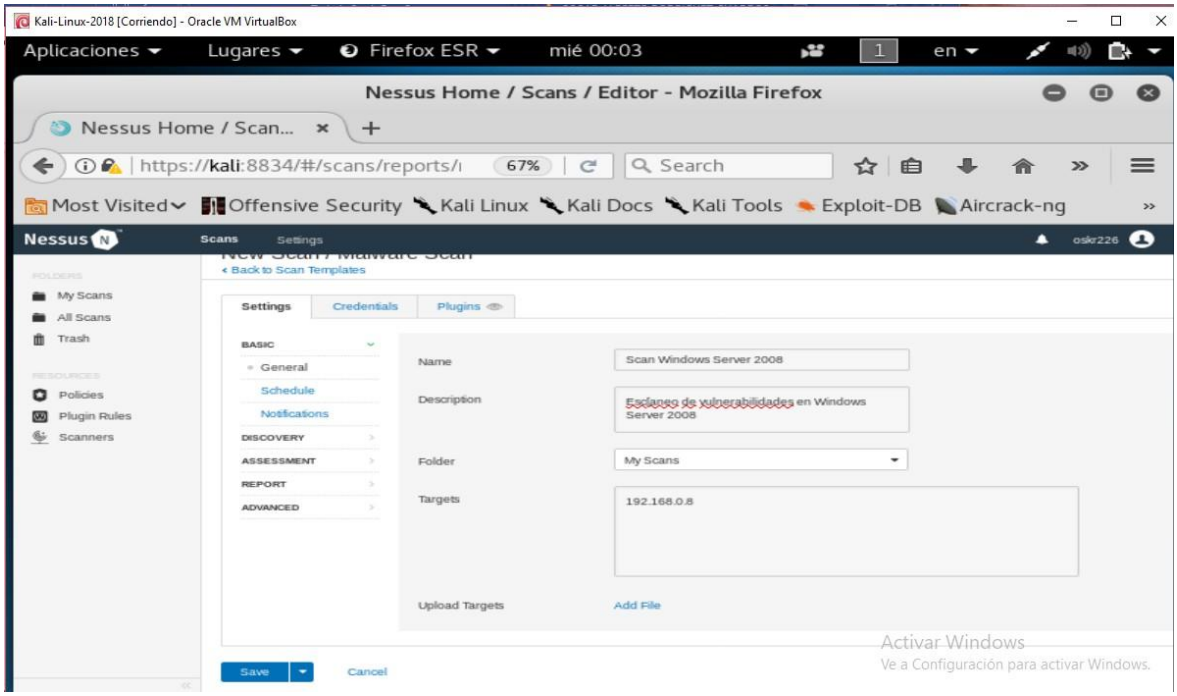


Figura 27 Configuración del primer escáner

Una vez termina la configuración se debe dar clic en “Save” o guardar.

Luego en la sección principal seleccionamos el scan que se configuró y se ejecuta desde la opción desplegable “more” y se da clic en la opción “launch”.

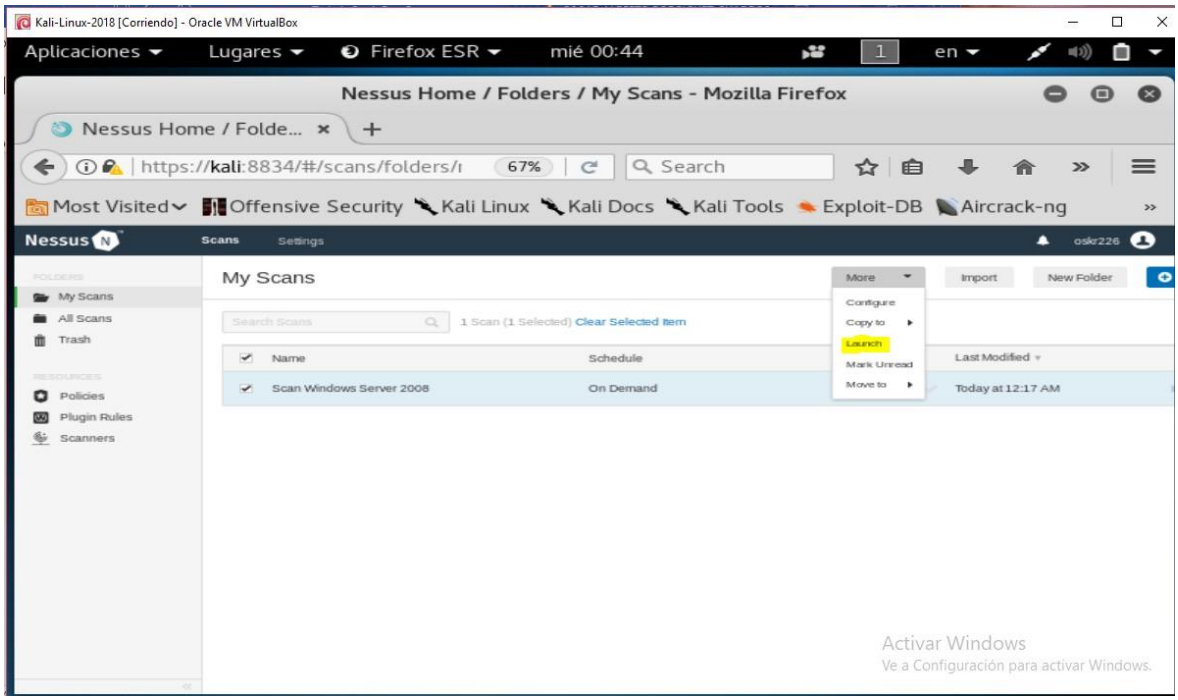


Figura 28 Menú para lanzar el scan sobre el objetivo

Antes de que se ejecute el scan el sistema preguntará si estamos seguros de ejecutarlo. Se acepta y el sistema realizará el scan sobre el sistema objetivo.

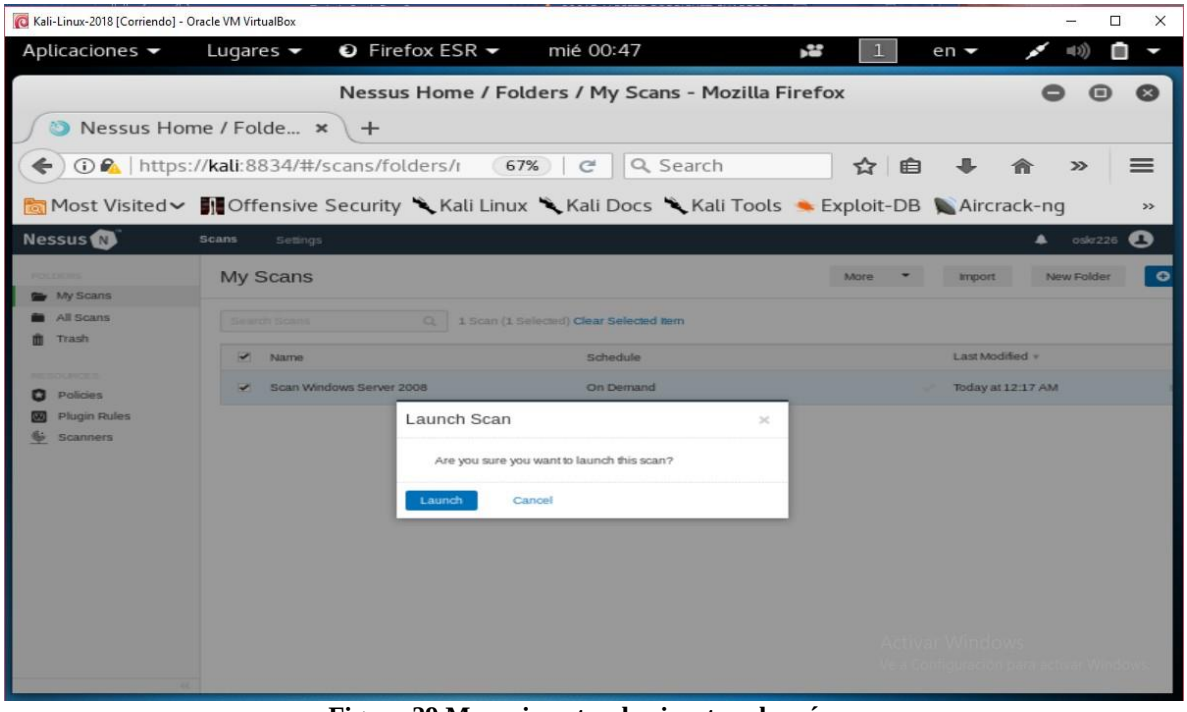


Figura 29 Mensaje antes de ejecutar el escáner

Nessus muestra el estado del escaneo mientras realiza el proceso de detección de vulnerabilidades.

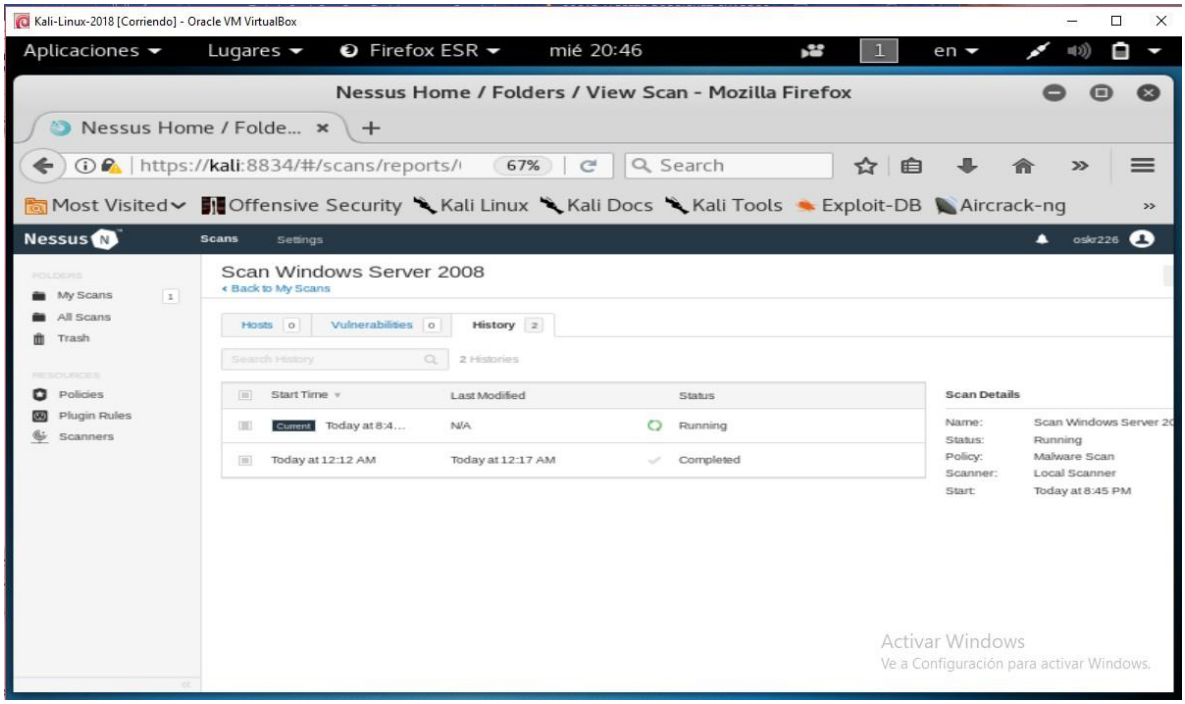


Figura 30 Proceso de escaneo de vulnerabilidades

Una vez el sistema termina el proceso de escanear las vulnerabilidades cambia el estado a “completed”. En la pestaña “Vulnerabilidades” enumera los huecos de seguridad encontrados en el sistema objetivo.

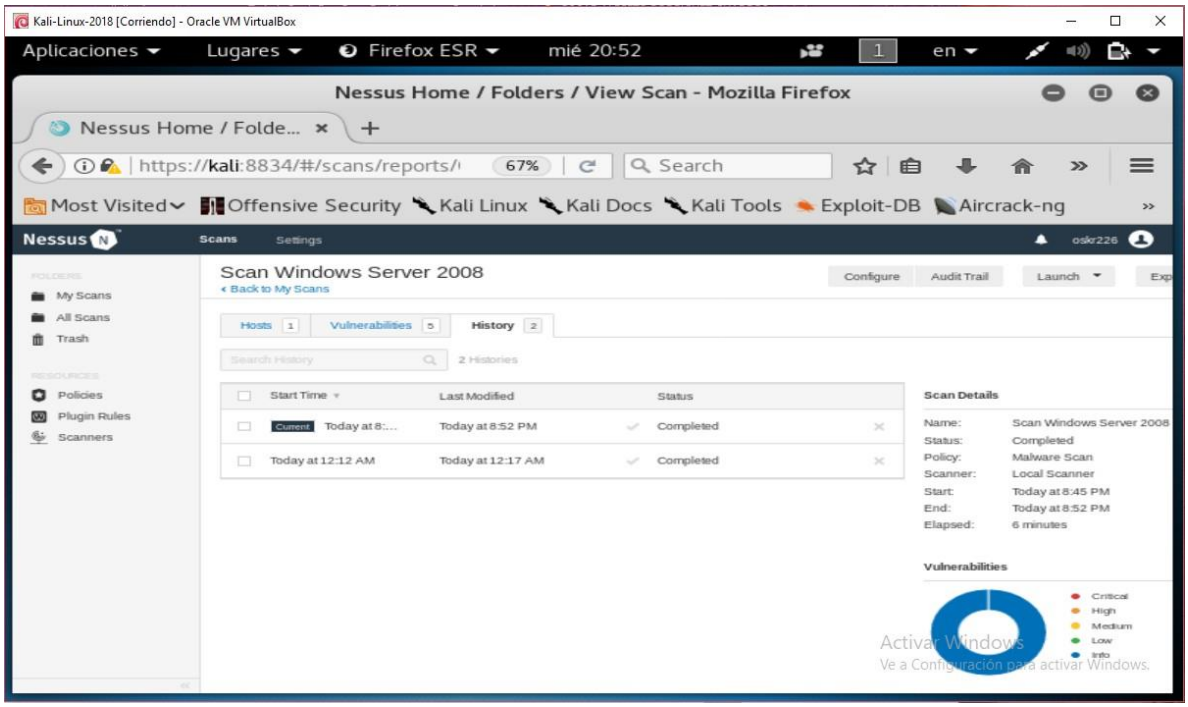


Figura 31 Escaneo de vulnerabilidades completo

Al dar clic en el registro del escaneo se puede ver en detalle los resultados obtenidos por Nessus

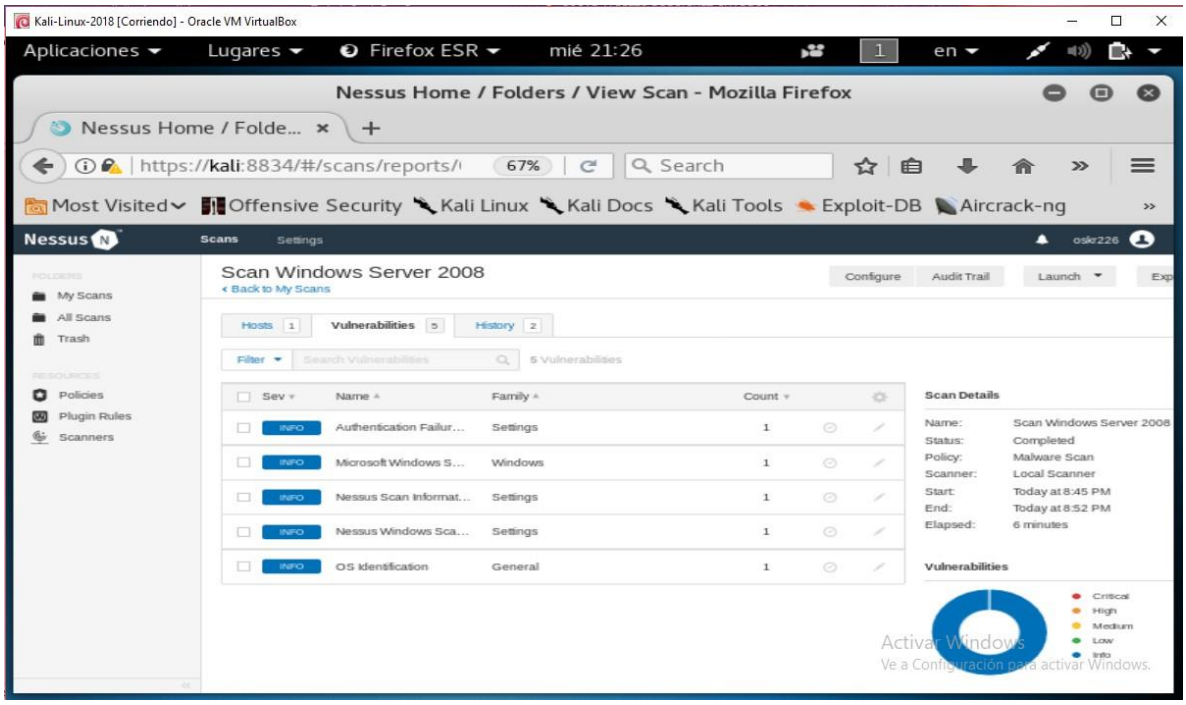


Figura 32 Resultados del escaneo de vulnerabilidades

Nessus clasifica las vulnerabilidades encontradas con un color de acuerdo a su importancia o criticidad. Para los resultados obtenidos se ve que a pesar de que el objetivo es un sistema operativo Windows Server 2008, sin antivirus, sin actualización, con el firewall desactivado y con servicios como telnet y IIS activos no hay un huevo de criticidad alta dentro del sistema.

Pero los resultados pueden dar cierta información importante a cerca del sistema ataquado. Al dar clic sobre uno de los resultados, Nessus abre un detalle con información que puede ser relevante.

Se da clic en el primer resultado para ver el detalle.

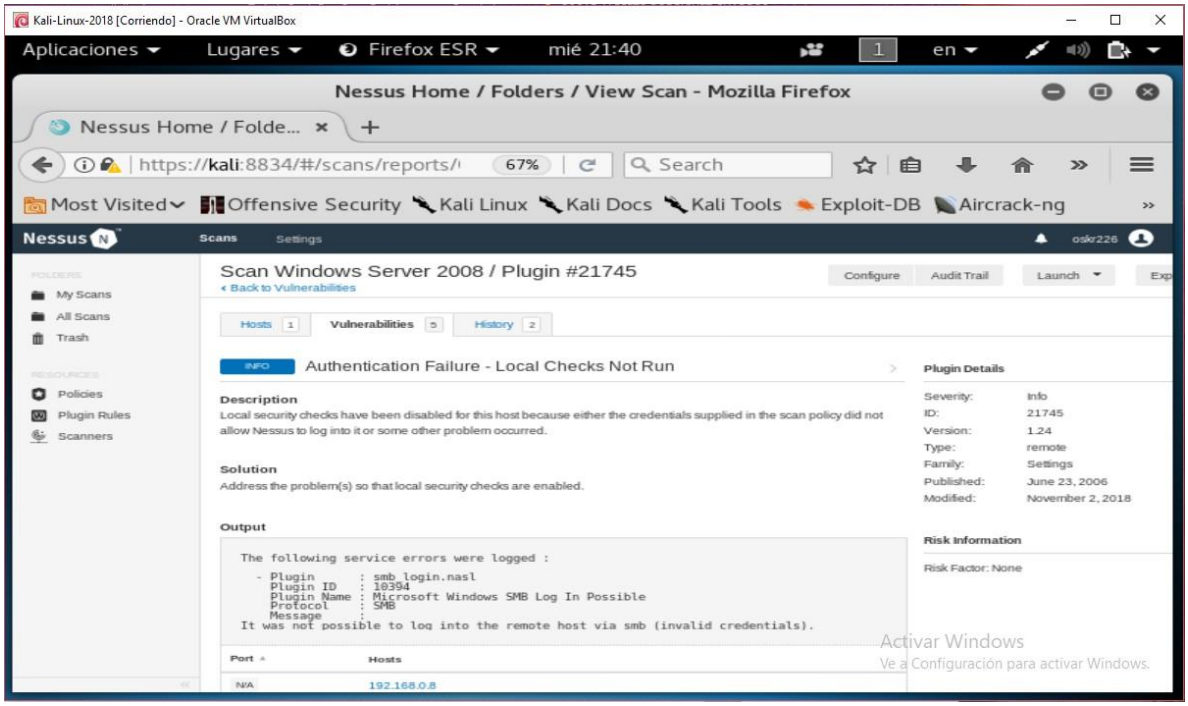


Figura 33 Resultado 1

El primer resultado indica que algo no permitió que Nessus hiciera login dentro de la máquina objetivo

Authentication Failure – Local checks not run: Las comprobaciones de seguridad locales se han deshabilitado para este host porque las credenciales proporcionadas en la política de exploración no permitieron que Nessus iniciara sesión o se produjo algún otro problema

Adicional, es Nessus quien propone una solución para evitar este inconveniente.

Solution: Solucione los problemas para que las comprobaciones de seguridad locales estén habilitadas.

En el campo de salida, agrega detalles un poco más técnicos indicando el plugin utilizado, el ID del plugin, agrega el nombre del sistema operativo.

```
The following service errors were logged :

  - Plugin      : smb_login.nasl
    Plugin ID   : 10394
    Plugin Name : Microsoft Windows SMB Log In Possible
    Protocol    : SMB
    Message     :
It was not possible to log into the remote host via smb (invalid
credentials)
```

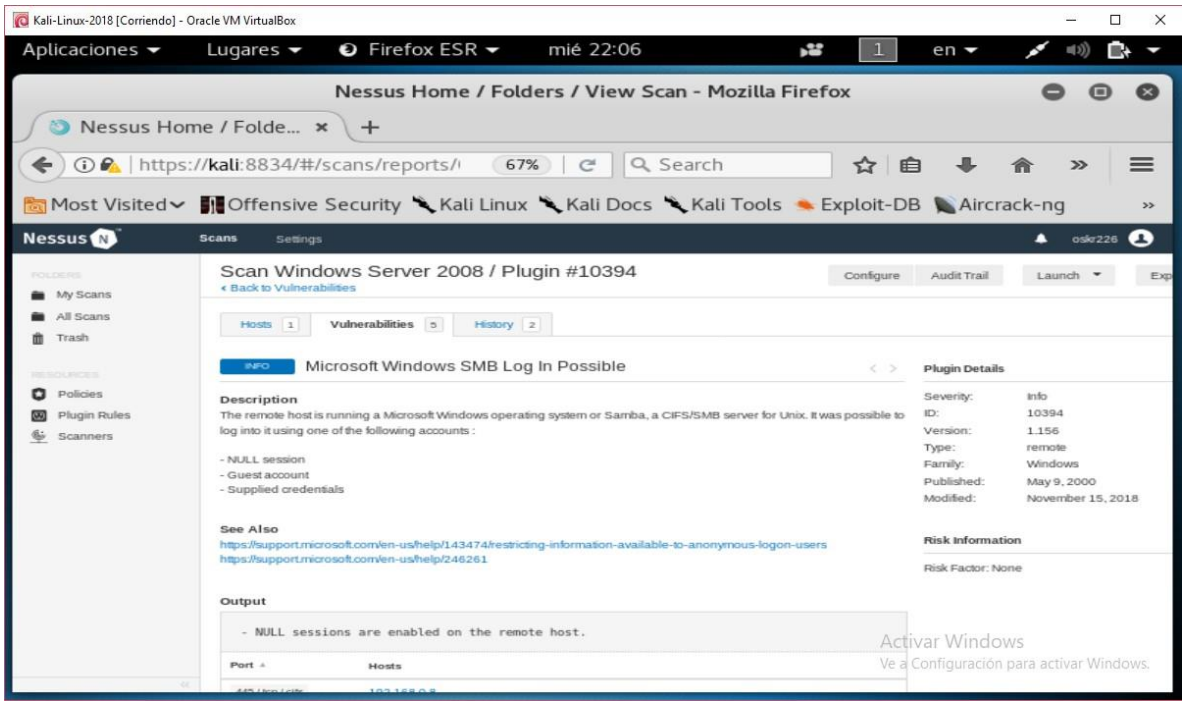


Figura 34 MIcrosoft Windows SMB Log In Possible

El siguiente resultado indica:

Microsoft Windows SMB Log In Posible: El host remoto ejecuta un sistema operativo Microsoft Windows o Samba, un servidor CIFS / SMB para Unix. Es posible iniciar sesión, utilizando una de las siguientes:

- NULL sesión
- Guest account
- Suplied credencials

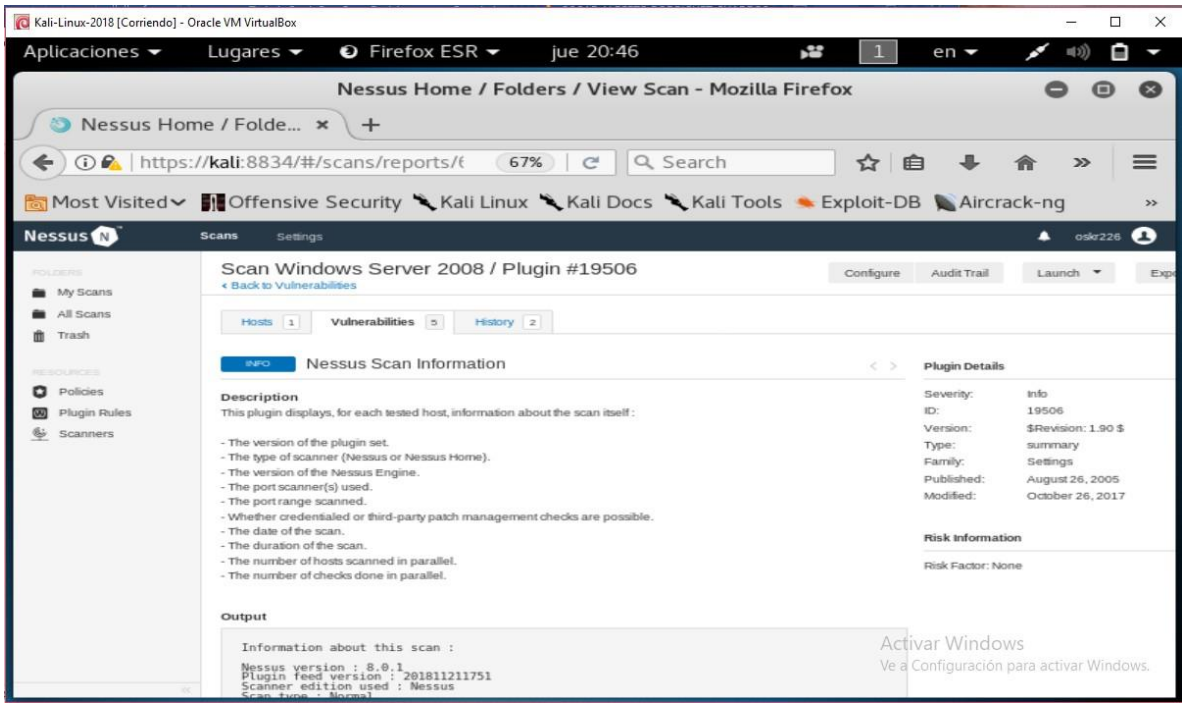


Figura 35 Nessus Scan Information

En la sección de Nessus Scan Information: Este complemento muestra, para cada host probado, información sobre el análisis en sí.

Nessus aporta detalles técnicos de su escaneo:

```
Information about this scan :

Nessus version : 8.0.1
Plugin feed version : 201811211751
Scanner edition used : Nessus
Scan type : Normal
Scan policy used : Malware Scan
Scanner IP : 192.168.0.11

WARNING : No port scanner was enabled during the scan.
```

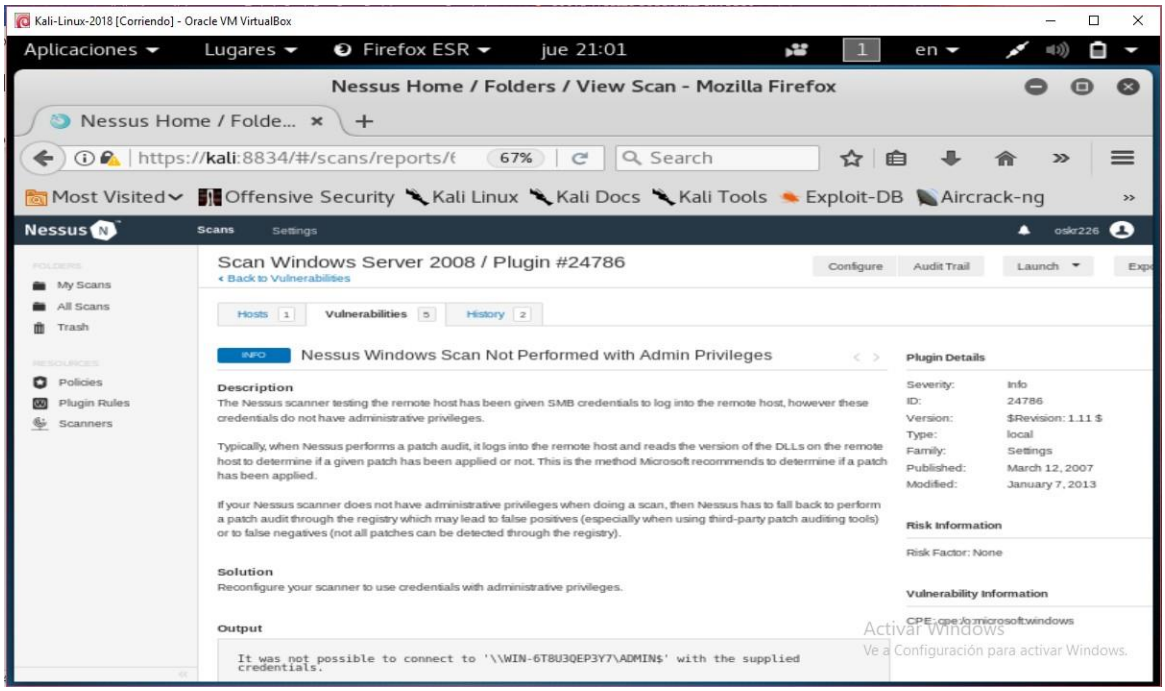


Figura 36 Escaneo sin privilegios – Nessus

Este mensaje del escaneo indica que Nessus recibió credenciales para iniciar sesión en el equipo objetivo sin embargo estas credenciales no tenían privilegios de administrador.

Normalmente, cuando Nessus realiza una auditoría de parches, inicia sesión en el host remoto y lee la versión de las DLL en el host remoto para determinar si un parche dado se ha aplicado o no. Este es el método que recomienda Microsoft para determinar si se ha aplicado un parche.

Si Nessus no tiene privilegios administrativos cuando realiza un escaneo, entonces Nessus tiene que recurrir para realizar una auditoría de parches a través del registro, lo que puede generar falsos positivos (especialmente al usar herramientas de auditoría de parches de terceros) o falsos negativos (no todos los parches pueden ser detectados a través del registro).

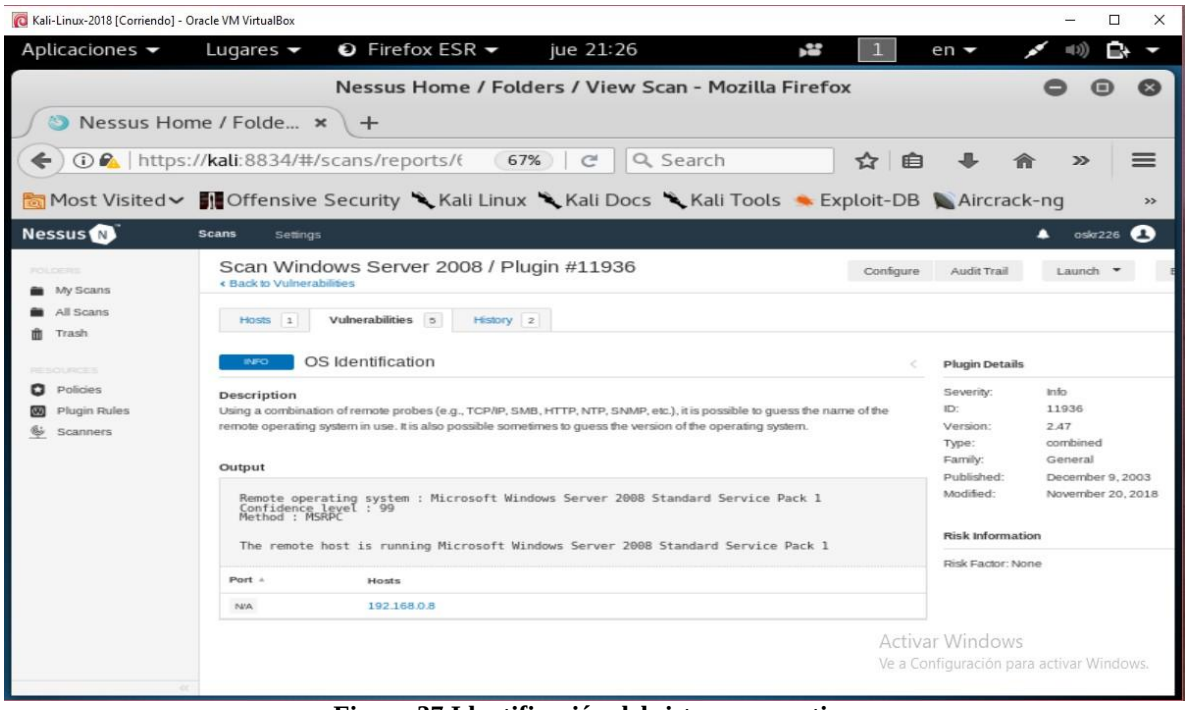


Figura 37 Identificación del sistema operativo

Usando una combinación de sondas remotas (por ejemplo, TCP / IP, SMB, HTTP, NTP, SNMP, etc.), es posible identificar el nombre del sistema operativo remoto en uso. También es posible a veces identificar la versión del sistema operativo.

6.2.3. Metasploit Framework

Según la documentación oficial de Metasploit Framework esta herramienta está basada en el lenguaje de programación Ruby. Esta plataforma permite desarrollar y ejecutar scripts que aprovechan vulnerabilidades conocidas de un sistema operativo en concreto. Desde que se empezó el desarrollo de esta plataforma se han ido incluyendo muchas herramientas desarrolladas para probar vulnerabilidades de seguridad y ejecutar ataques.

6.2.3.1. MSFConsole

Según la documentación oficial de Metasploit Framework MSFConsole es una interfaz de línea de comandos que permite acceder a las herramientas internas de Metasploit framework. Esta consola permite ejecutar comandos y cargar herramientas que realizan diferentes acciones como escanear equipos dentro de una red y explotar vulnerabilidades.

6.2.3.2. Módulos Metasploit

Según la documentación oficial de Metasploit Framework, los módulos son los componentes principales de Metasploit. Cada módulo está diseñado para realizar una tarea en concreto. Cada componente puede ejecutar acciones de exploración de objetivos dentro de una red o explotación de vulnerabilidades.

6.2.3.3. Tipos de módulos

La clasificación de los módulos descrita en la documentación oficial se da según el objetivo para el que fue desarrollado, ya sea de escaneo, explotación o algún otro objetivo.

- Exploit: Es desarrollado para la ejecución de comandos con el fin de explotar una vulnerabilidad concreta dentro de un sistema.
- Auxiliar: Este tipo de módulos son desarrollados para realizar escaneos o ataques de denegación de servicios.
- Post-Explotación: Este tipo de módulos se desarrollan para obtener información adicional de un sistema o mantener el acceso a un sistema luego de la explotación de su vulnerabilidad.
- Payload: Es un código que se ejecuta dentro del sistema comprometido luego de la explotación de una vulnerabilidad. Esto determina que tipo de acciones se realizan dentro del sistema objetivo después de tomar el control de este, entre los cuales esta abrir un Meterpreter o ejecutar un comando Shell. Según la documentación oficial un Meterpreter es un payload avanzado que permite escribir archivos de tipo DLL dentro del sistema afectado para crear nuevas funcionalidades según lo que se desee realizar dentro del sistema objetivo.

6.2.3.4. Payloads

Según la documentación oficial de Metasploit hay muchos tipos de payloads para diferentes escenarios. El objetivo de un payload es obtener un Shell para la ejecución de comandos. Entre los payloads más conocidos según la documentación oficial esta “Windows/meterpreter/reverse_tcp”. También existe el payload “Windows/meterpreter/reverse_https” que se considera más seguro ya que utiliza un canal cifrado.

Algunos payloads constan de dos partes, un cargador inicial y un payload de etapa final. El payload inicial se encarga de explotar la vulnerabilidad y cuando envíe la

respuesta al extremo inicial (Extremo del atacante) pedirá que se envíe el payload final para obtener un Shell.

Aprovechando los resultados obtenidos a través de la herramienta Nmap donde se escanean los puertos abiertos del sistema objetivo Windows Server 2008, se tiene información relevante que permite seleccionar el tipo de sploit que permite aprovechar el hueco de seguridad dentro del sistema.

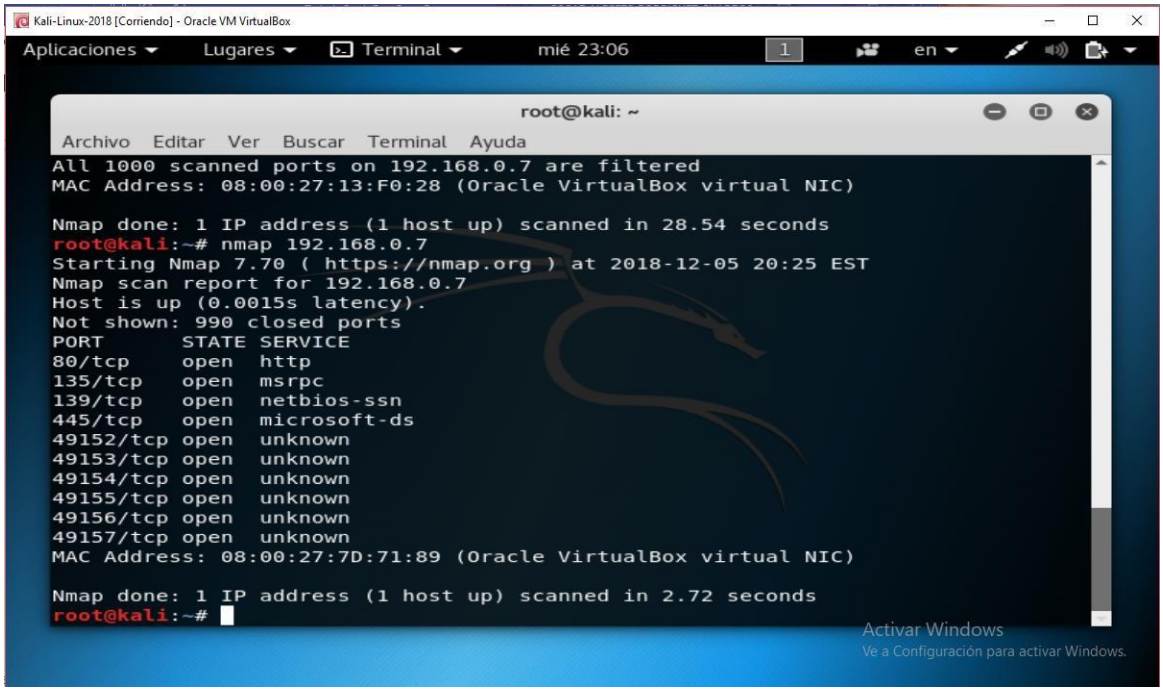


Figura 38 Escaneo de puertos con Nmap - Fuente el autor

Con los resultados del escaneo vemos que el puerto 445 está abierto y que expone un servicio llamado Microsoft-ds. Dado lo anterior, podemos usar un sploit que aproveche el hueco de seguridad conocido para este puerto a través de una conocida falla en la implementación del protocolo SMB2, el cual permite a un atacante la ejecución de código de forma remota.

Lo primero que se debe hacer es iniciar la consola de metasploit framework. Para ello en la terminal se ejecuta el comando “msfconsole”.

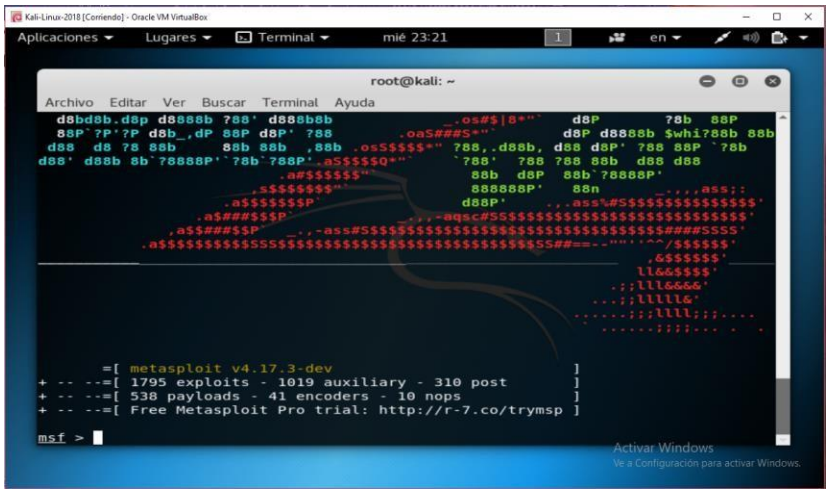


Figura 39 Inicio de la consola de Metasploit Framework – Fuente el autor

La máquina objetivo es un sistema operativo Windows Server 2008 que tiene asignado la IP 192.168.0.7 como lo muestra la siguiente imagen:

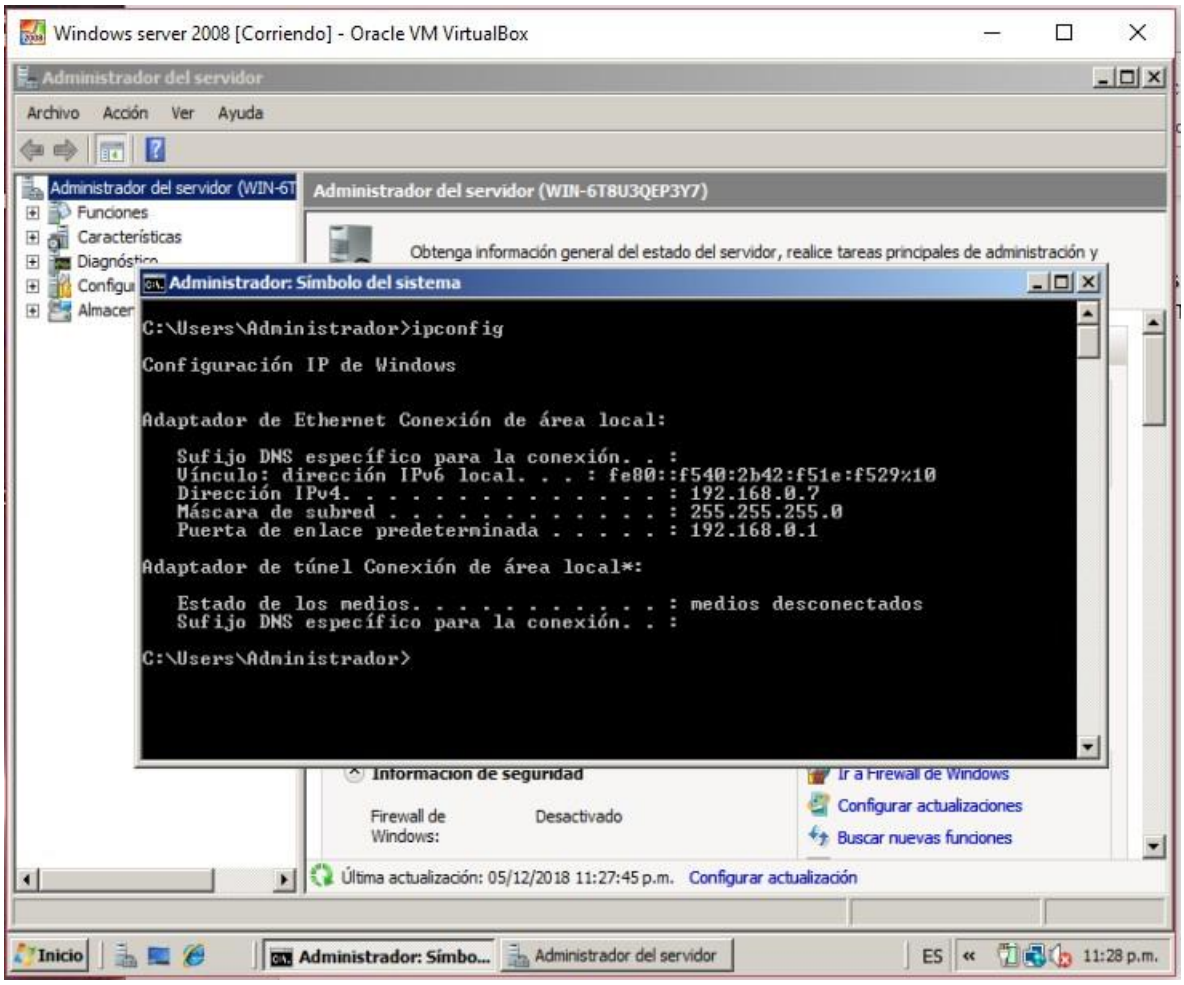


Figura 40 Sistema objetivo Windows Server 2008 - Fuente el autor

Para aprovechar la vulnerabilidad conocida por el puerto 445 se debe usar el sploit “ms09_050_smb2_negotiate_func_index”. Para ello se ejecuta en la terminal el comando: “use exploit/Windows/smb/ ms09_050_smb2_negotiate_func_index”.

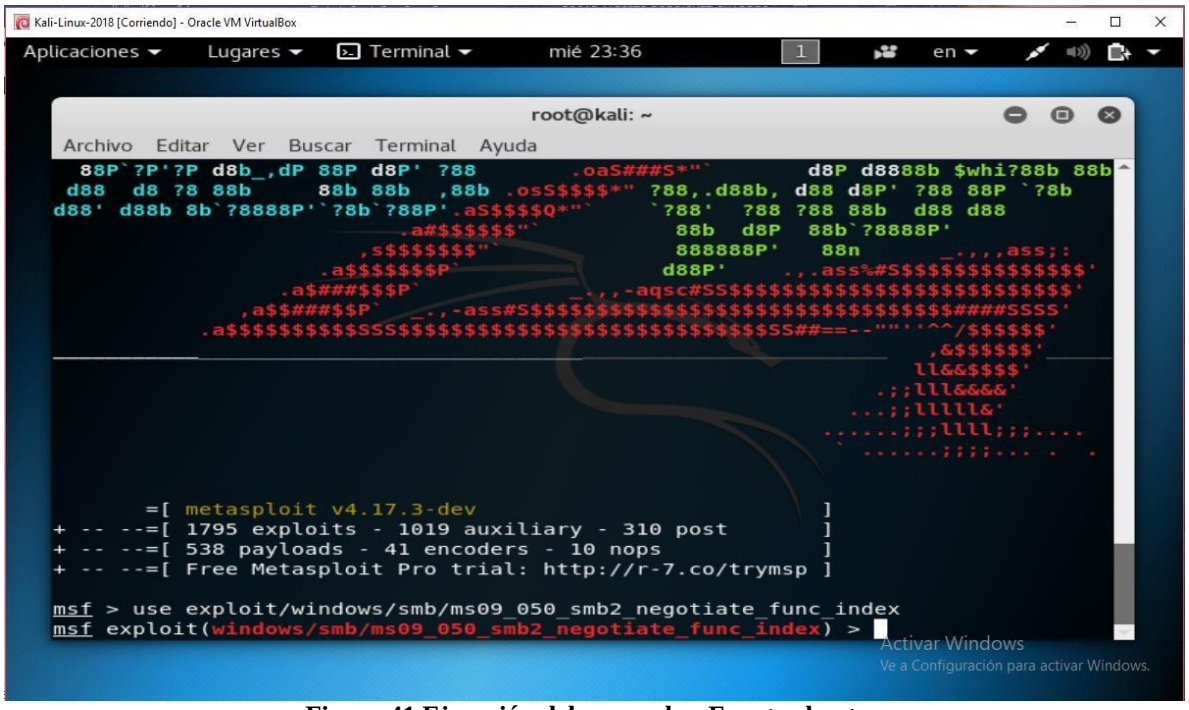


Figura 41 Ejecución del comando - Fuente el autor

Lo siguiente es cargar el payload, para ello se ejecuta el siguiente comando: “set payload Windows/meterpreter/reverse_tcp”.

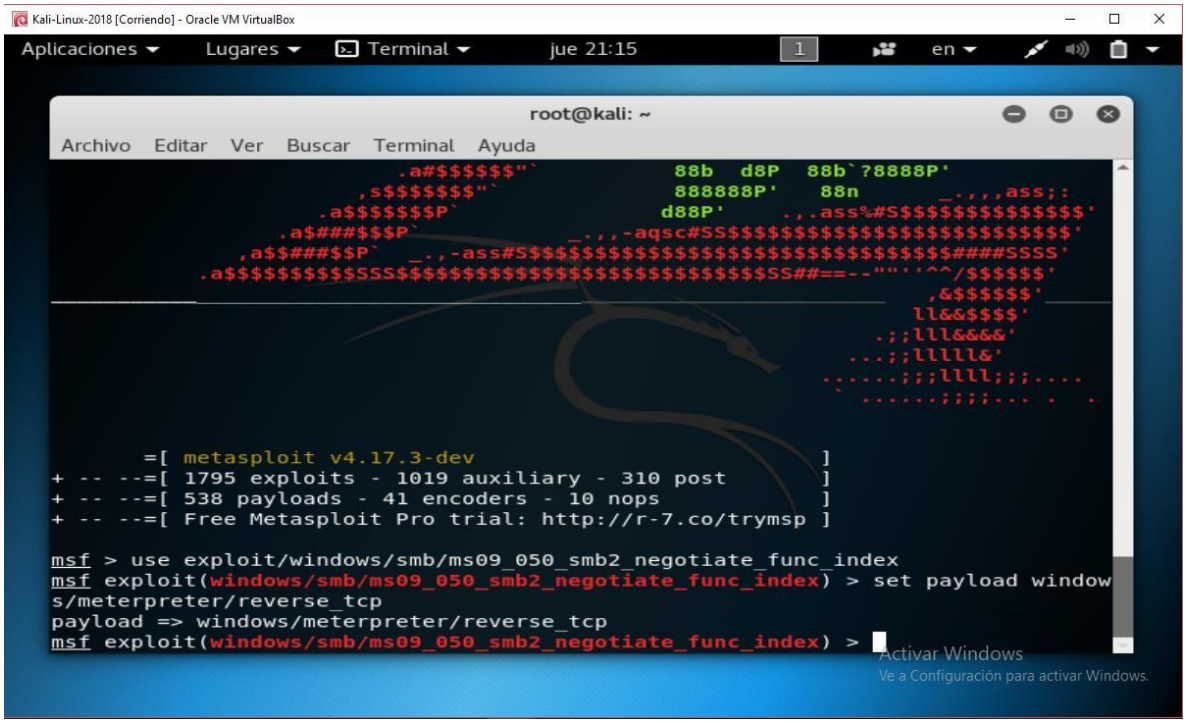


Figura 42 Carga del payload - Fuente el autor

Lo siguiente es agregar la máquina objetivo y el puerto que se quiere atacar. Para este caso la máquina victima es la que tiene la IP 192.168.0.7 y el puerto 445 que tiene abierto y que ejecuta el servicio “Microsoft-ds”.

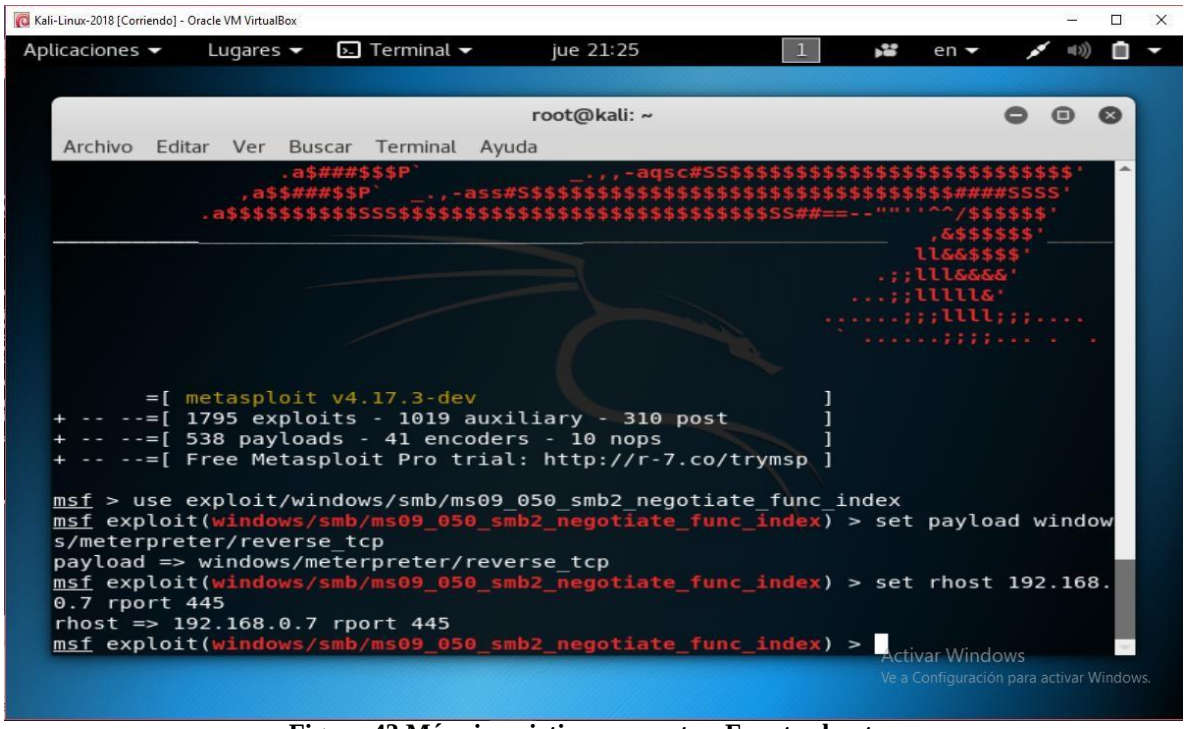


Figura 43 Máquina victima y puerto - Fuente el autor

Lo siguiente es agregar la IP de la máquina atacante, para este caso la IP de la máquina Kali es “192.168.0.7” y el puerto por donde se realizará el ataque será el puerto 4000.

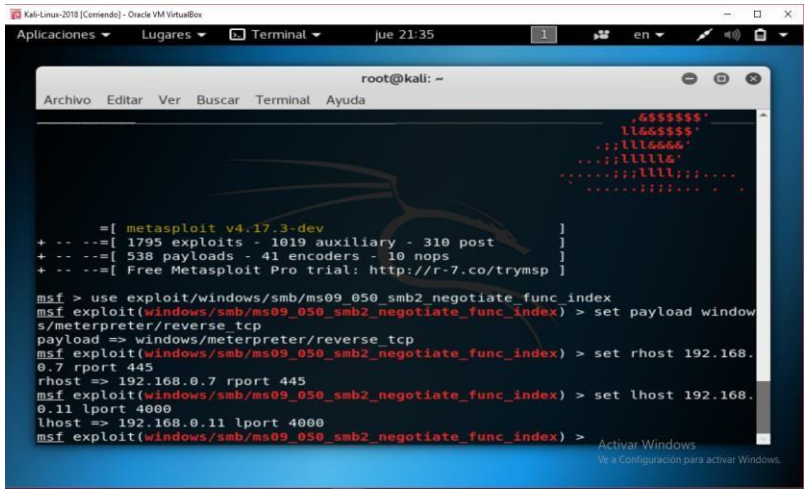


Figura 44 Se registra la máquina que realiza el ataque - Fuente el autor

Después de la configuración, la aplicación ya está lista para realizar el ataque. Para iniciar se debe ejecutar el comando “exploit” en la terminal y el ataque comenzará.

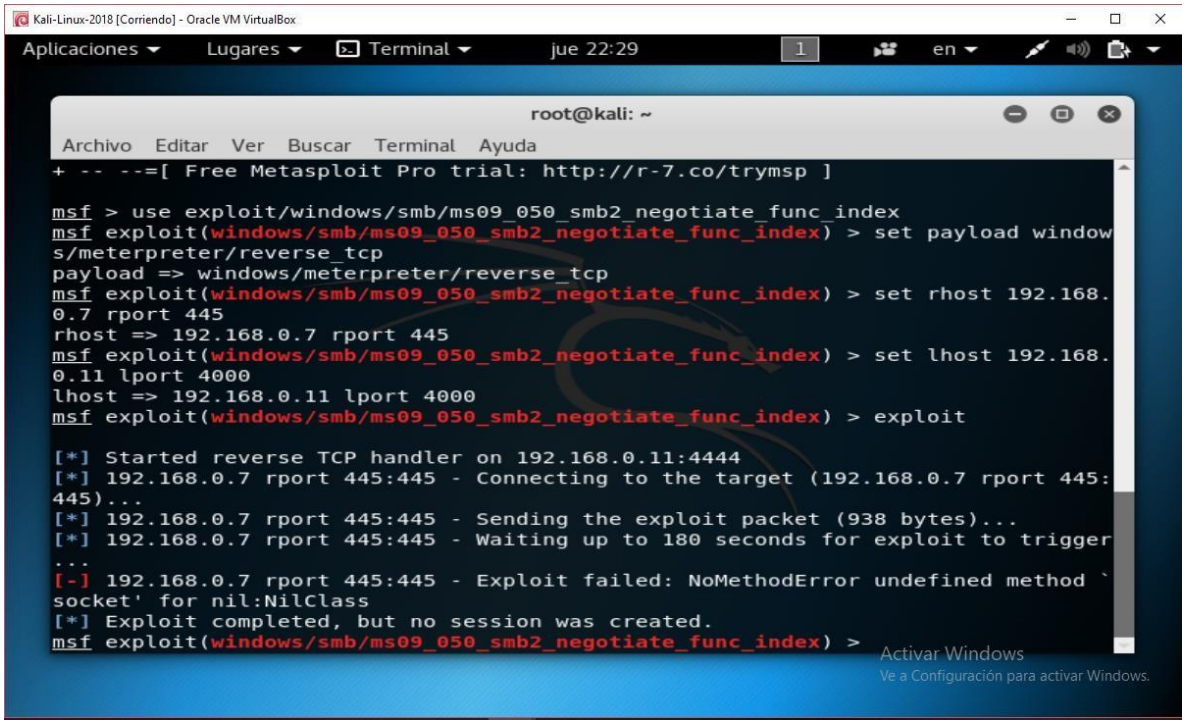


Figura 45 Ejecución del ataque al equipo victima

El ataque originalmente pretende iniciar sesión en la máquina victima con el fin de tener acceso tanto a la información como a la ejecución de comandos. El ataque puede cumplir dos objetivos: el primero descrito anteriormente como el acceso al equipo víctima, el segundo objetivo es generar una denegación de servicios en el equipo objetivo ocasionando un reinicio inesperado provocando un error en la ejecución del sistema operativo.

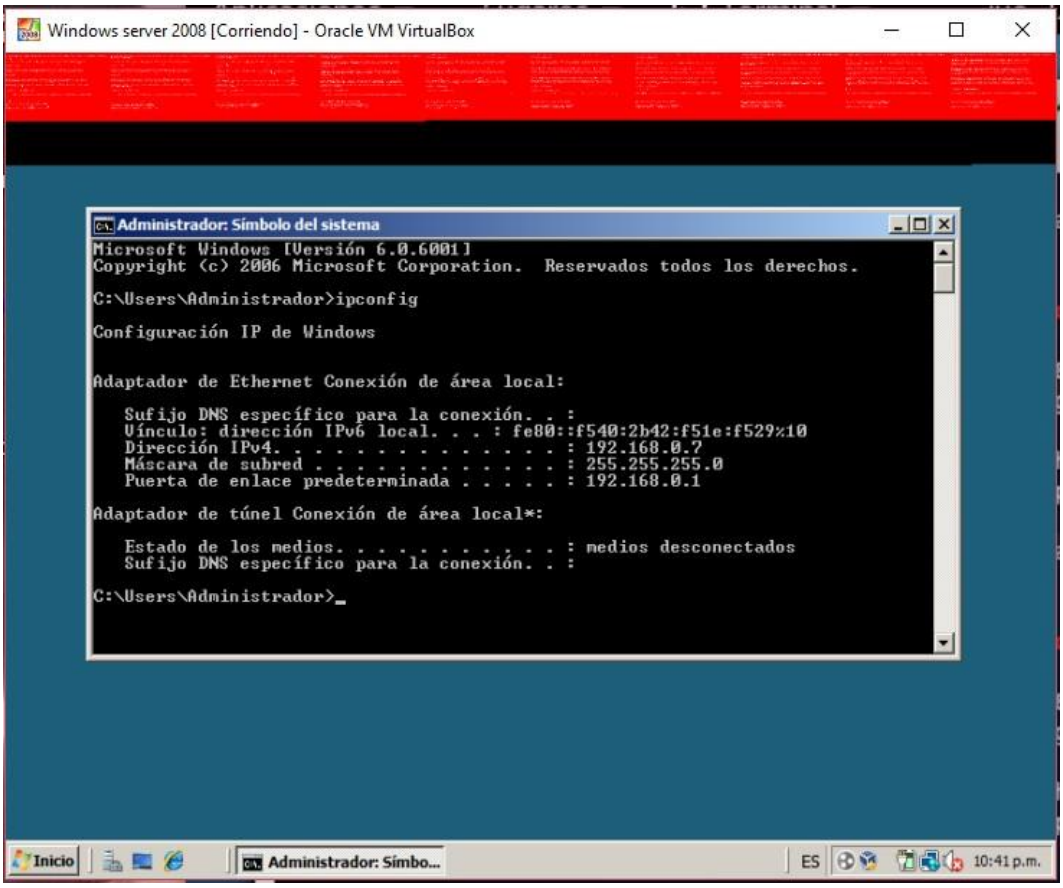


Figura 46 Efectos del ataque al equipo victima

Se evidencia en el equipo víctima los efectos del ataque desde la máquina con el sistema Kali Linux. Inesperadamente el sistema operativo Windows Server 2008 se bloquea y se genera un reinicio en el sistema.

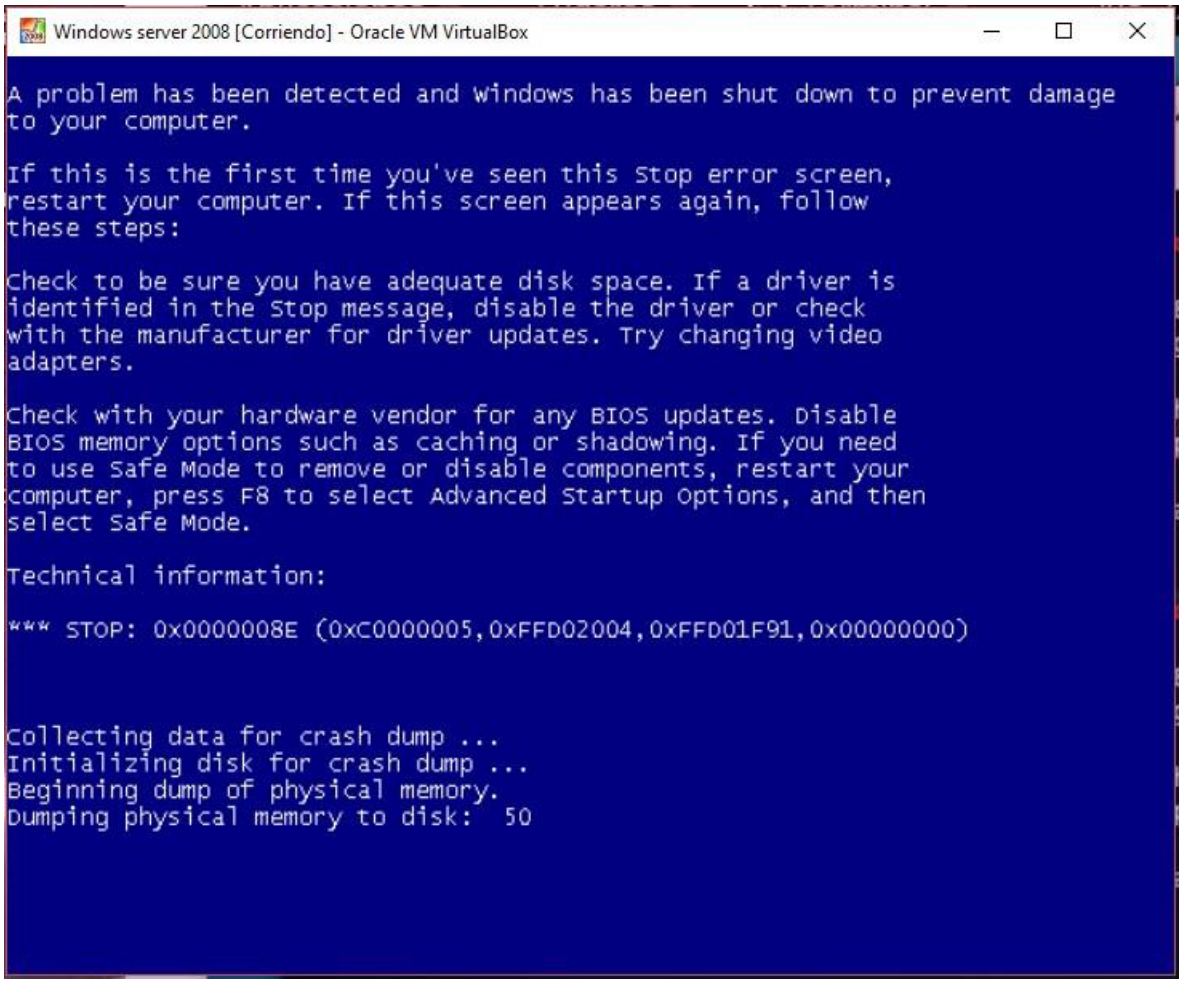


Figura 47 Pantallazo azul a causa del ataque

Una vez que el equipo victima se ha reiniciado el sistema vuelve a trabajar con normalidad, pero a un costo muy alto. El servidor Windows Server 2008 ha tenido que reiniciar todos sus servicios para volver a estar operativo. En un entorno real este tipo de ataque sería muy perjudicial para una organización provocando una parálisis momentánea de procesos internos, acceso a la información, comunicaciones en caso de que sea un servidor de correos.

Este ejercicio de hacking ético pretende orientar de forma básica sobre la forma adecuada en la que se debe realizar las pruebas de seguridad sobre una red o un equipo en específico y resaltar la importancia de conocer las debilidades de una red o sistema y como poder fortalecerlo a partir de las vulnerabilidades conocidas.

CAPÍTULO III

MANUAL BÁSICO DE PENTESTING

7. Capitulo III: Manual básico de pentesting

7.1. Introducción

En el presente capítulo tiene con finalidad la realización de un manual de ejecución en hacking ético, enfocado en el descubrimiento de vulnerabilidades. Por ende, se estructura principalmente en tres particiones fundamentales. En primera instancia, se elabora una fase de reconocimiento, en donde se aplica la herramienta de Nmap. En segunda instancia, se encuentra una partición de descubrimiento de vulnerabilidades, cuyo enfoque corresponde a la ejecución de la herramienta Nessus para identificar las vulnerabilidades anteriormente mencionadas. Para esta sección se debe identificar los tipos de análisis, además de tener en cuenta lo siguiente: selección de plantilla de ejecución, la configuración referente al IP de la máquina y los usuarios y contraseñas y finalmente la ejecución.

Por último, se encuentra la partición de “ataque” en donde se ejecuta la aplicación de Metasploid teniendo en cuenta la configuración para los tipos de ataques, y la ejecución.

Ahora bien a continuación, se realiza una breve explicación sobre la herramienta de Nmap y sus generalidades, basados en el manual de Nmap revisado de www.nmap.org. En primera instancia, se inicia con el abordaje de Nmap como herramienta, pues, ésta utiliza paquetes de IP, para identificar si un host está disponible dentro de una red. Lo anterior, depende de la respuesta que se dé en el puerto, Nmap, puede identificar qué servicios se ejecutan a través del puerto, y qué sistema operativo está instalado en el equipo. Esta herramienta cuenta con grandes bondades, entre ellas se tienen: flexible, pues en ella se utilizan diversas técnicas de escaneo. Es potente, pues escanea miles de máquinas al tiempo. Portátil, pues en ella se ejecutan varios sistemas operativos. Es de fácil acceso y está bien documentado pues contienen en ella manuales disponibles en varios idiomas. Es gratis, pues su código es abierto, y por último muy fácil de usar. En resumen, el portal de Nmap refleja una de las tantas funcionalidades en el siguiente referente: “Nmap se ejecuta en todos los principales sistemas operativos de computadoras, y los paquetes binarios oficiales están disponibles para Linux, Windows y Mac OS X. Además del clásico ejecutable de línea de comandos Nmap,Zenmap), una herramienta flexible de transferencia de datos, redirección y depuración (Ncat), una utilidad para comparar resultados de escaneo (Ndiff) y una herramienta de análisis de generación de paquetes y respuesta (Nping).”

7.2. Fundamento de escaneo de puertos

Con esta herramienta, Nmap, se puede escanear hasta 1.000 puertos de paquetes TCP de manera simultánea, y se divide en seis principales estados:

1. Abierto: acepta conexión TCP data gramas UDP o asociación SCTP.
2. Cerrado: se determina cuando no hay ninguna aplicación escuchando a través de este.
3. Filtrado: impiden que Nmap pueda identificar que el puerto está abierto o cerrado, estos puertos proporcionan muy poca información.
4. Sin filtrar: es accesible pero no se sabe si está abierto o cerrado.
5. Abierto filtrado: se agregan a los puertos que no se pueden determinar si está abierto o cerrado.
6. Cerrado filtrado: se usa cuando no se puede determinar si un puerto está abierto o cerrado.

La anterior información se puede confrontar en el portal de Nmap que contiene un manual de su uso, en múltiples idiomas, incluyendo por supuesto el español. www.nmap.org.

7.3. Técnicas de escaneo de puertos

Nmap es una herramienta que usa las respuestas que recibe de los escaneos para obtener información: versión de sistema operativo tipo de servicio que se

Técnica de escaneo de Nmap
Forma en la que la herramienta puede escanear un puerto, ahora bien, se revisará los tipos de escaneo según el portal de Nmap:

Primera forma de Escaneo TCP SYN: es una opción de escaneo predeterminada de mil puertos por segundo, en una red sin restricciones.

Segunda forma de Escaneo UDP: entre los servicios se encuentran DNS-SNNP-DHCP. Esta forma de escaneo se caracteriza por su lentitud.

Tercera Forma de Escaneo INIT SCTP: eta técnica se denomina semiabierto, puesto que, envía un fragmento INIT y luego espera una respuesta. Un fragmento INIT-ack indica que el puerto está escuchando. Un fragmento ABORT indica que el puerto no está escuchando. Sino se recibe respuesta después de varios intentos, indica que el puerto está filtrado.

Cuarta Forma de Escaneo TCP NULL: para identificar que un puerto está cerrado con esta técnica, se analiza el segmento entrante, sino contiene un RST.

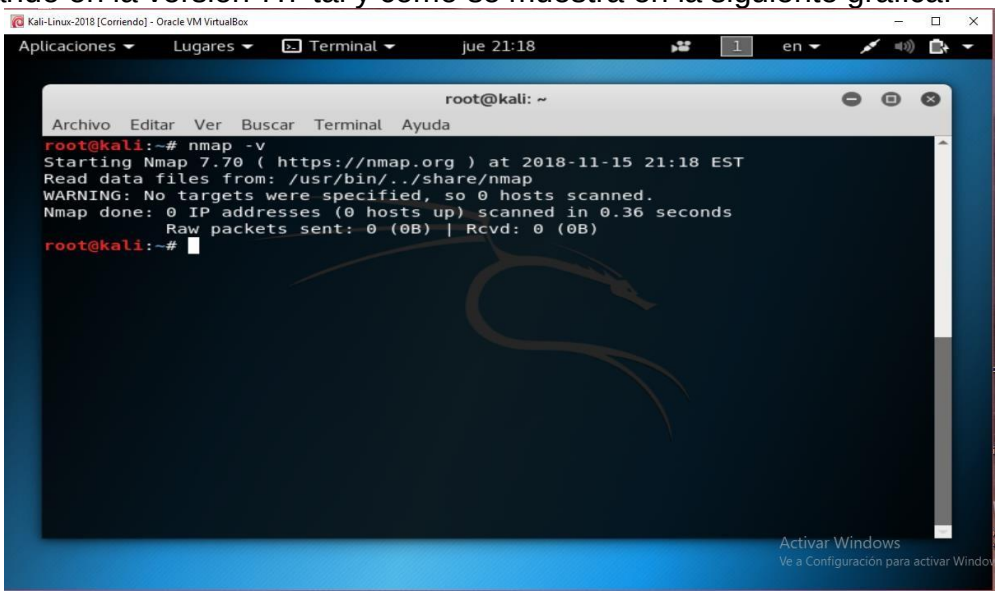
7.4. Fase de reconocimiento

Objetivo: En el ejercicio realizado anteriormente, y en el que se va a trabajar en el siguiente manual, se realiza un ataque desde el equipo con Sistema Operativo Kali Linux a un equipo Windows 2008 server de la red. A continuación, se hará un paso a paso de los lineamientos fundamentales para conocer la aplicación de la herramienta Nmap.

1. **Instalación del Sistema Operativo Kali Linux.**

2. **Verificación de la versión Nmap.**

Para este paso, es necesario utilizar el comando “nmap -b” y debe estar trabajando en la versión 7.7 tal y como se muestra en la siguiente gráfica.



7.5. Escaneo de Red

Para esta función se necesita usar el comando "nmap -sn 192.168.0.0/24" que tiene como función conocer qué equipos se encuentran en la red y cuáles de ellos se encuentran encendidos y en línea.

Namp scan report for 192.168.0.7
Host is up (0.00047s latency)
MAC Address: 50:B7:C3:D0:5D:22 (Samsung Electronics)

Nmap scan report for 192.168.0.8
Host is up (0.0018s latency)
MAC Address: 08:00:27:7D:71:89 (Oracle VirtualBox Virtual NIC) Equipo con Windows server 2008

Nmap scan report for 192.168.0.9
Host is up (0.00064s latency)
MAC Address: 08:00:27:68:75:AC (Oracle VirtualBox Virtual NIC) Equipo con Windows server 2008

Nmap scan report for 192.168.0.11
Host is up
Equipo Kali Linux.

Nmap done: 256 IP addresses (5 hosts up) scanned in 2.39 seconds.

7.6. Escaneo de puertos

Una vez se obtiene el resultado anterior, se ejecuta "nmap -open 192.168.0.8" que sirven para escanear los puertos que tiene abiertos una máquina. El resultado para tal comando debe ser el siguiente:

Nmap scan report for 192.168.0.8
Host is up (0.0011s latency).
Not shown: 991 closed ports
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
49152/tcp open unknown
49153/tcp open unknown
49154/tcp open unknown
49155/tcp open unknown
49156/tcp open unknown
49157/tcp open unknown
MAC Address: 08:00:27:7D:71:89 (Oracle VirtualBox virtual NIC)

7.7. Escaneo de puertos en detalle

Para este paso, es importante conocer los servicios del sistema operativo que se están ejecutando en los puertos abiertos. En tal sentido, se debe usar el comando "nmap -sV -T4 192.168.0.8", cuyo resultado debe responder así:

PORT STATE SERVICE VERSION
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
445/tcp open microsoft-ds (workgroup: WORKGROUP)
49152/tcp open msrpc Microsoft Windows RPC
49153/tcp open msrpc Microsoft Windows RPC
49154/tcp open msrpc Microsoft Windows RPC
49155/tcp open msrpc Microsoft Windows RPC
49156/tcp open msrpc Microsoft Windows RPC
49157/tcp open msrpc Microsoft Windows RPC

7.8. Escaneo del Tipo de Sistema Operativo

El objetivo de esta fase consiste en la identificación del sistema al cual se va a presentar el ataque. Para ello, se ejecuta el comando "nmap -vO -T4 192.168.0.8 para escanear el sistema operativo. Los resultados deben apreciarse así:

Device type: general purpose
Running: Microsoft Windows Vista|2008|7
OS CPE: cpe:/o:microsoft:windows_vista::sp1 cpe:/o:microsoft:windows_vista::sp2 cpe:/o:microsoft:windows_server_2008::sp2 cpe:/o:microsoft:windows_7
OS details: Microsoft Windows Vista SP1 - SP2, Windows Server 2008 SP2, or Windows 7

7.9. Fase de descubrimiento de vulnerabilidades

Objetivo: Escanear las vulnerabilidades que tiene un sistema operativo, bien sea, Windows o Linux. En este caso particular se enfoca en el último.

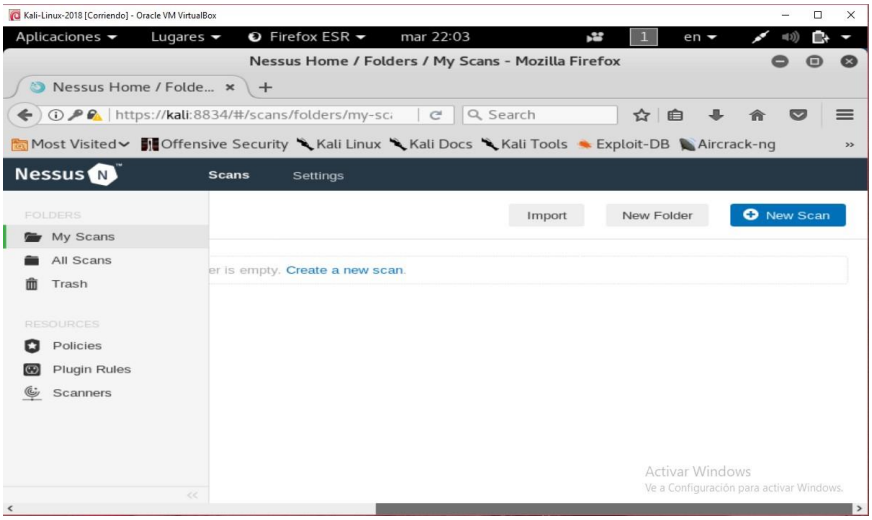
7.9.1. Instalación de la herramienta Nessus en el sistema operativo Kali Linux.

Para lo anterior, es necesario descargar (www.tenable.com/downloads/nessus) la última versión de la aplicación Nessus e instalarla en el sistema operativo Kali Linux. El archivo descargado tiene el siguiente nombre:

Nessus-8.0.1-debian6_i386.deb

Por lo anterior, es necesario ejecutar el comando # dpkg -i Nessus-8.0.1-debian6_i386.deb en una terminal para la instalación de la aplicación Nessus. Una vez instalado se debe ejecutar una vez más el siguiente comando para poder iniciar la aplicación, a saber; /etc/init.d/nessusd start. Después de iniciado Nessus, es necesario configurarlo, abriendo el navegador de internet (preinstalado con Kali Linux) y se debe ingresar la siguiente dirección: https://kali:8834

Ahora bien, lo anterior genera una alerta de seguridad y se debe confirmar la excepción de seguridad que muestra el navegador. Una vez se da inicio en la aplicación, se debe crear Usuario y Contraseña (de su preferencia) para comenzar a descargar los *pluggins* que se necesiten. Cuando se terminen de descargar y compilar los *pluggins* permitrá el ingreso a la aplicación y queda habilitada la interfaz para su uso, tal y como se presenta en la siguiente imagen.



7.9.2. Selección tipo de plantilla para escaneo

En la aplicación Nessus contiene unas plantillas preconfiguradas categorizadas por su utilidad o enfoque para escanear vulnerabilidades, que son: a. *basic network scann,* b. *Intel amt secutiry bypass* c. *malware scann* d. *bad look detección* entre otras.

7.9.3. Configuración del escaneo

Debe responder al siguiente paso a paso:

a. Opción My Scans (Seleccionar la plantilla *malvare scann*)

b. Configuración de la plantilla seleccionada. Generar nombre y descripción y la IP de la máquina que se va a scannear.
c. Guardar la configuración
d. Ir a la sección principal y se selecciona la plantilla que se acabó de configurar.
e. Opción desplegable **MORE** se selecciona la opción *launch.*
f. Luego de ejecutar *launch,* se muestra la plantilla en estado de ejecución
g. Una vez termine el scanner el estado de esta plantilla cambia a "completado".
h. En la pestaña de "vulnerabilidades" se enumeran los huecos de seguridad encontrados en el sistema objetivo.
i. Click al registro de escaneo para ver el resultado de vulnerabilidades encontradas. Las vulnerabilidades se clasifican según su criticidad en un color distinto así: 1. Rojo (vulnerabilidades críticas) 2. Naranja: (vulnerabilidades altas) 3. Amarillo: (vulnerabilidades medias) 4. Verde: (vulnerabilidades bajas) y Azul: (Resultado de tipo informativo). Si se desea conocer el detalle de cada vulnerabilidad encontrada, se debe dar click sobre la vulnerabilidad respectiva.

7.10. Fase de ataque

Objetivo: Ejecutar *metasploid framework* para aprovecharse de una vulnerabilidad conocida del Sistema Operativo Windows Server 2008.

Aprovechando los resultados obtenidos de la herramienta Nmap y Nessus, se puede identificar por donde se puede atacar al sistema Operativo objetivo. Para este caso, se va atacar a Windows 2008 server por el puerto 445 que se encuentra abierto, y que expone un servicio llamado Microsoft -DS. Para aprovechar esa vulnerabilidad se va a usar un Sploit que pueda atacar el puerto descrito anteriormente.

7.10.1. Pasos

a. Iniciar la consola de metasploit framework, para ello, se debe ejecutar el comando "msfconsole". Para esta guía se va a utilizar el *sploit* "ms09_050_smb2_negotiate_func_index". En una terminal se debe ejecutar el siguiente comando "use exploit/Windows/smb/ ms09_050_smb2_negotiate_func_index".
b. Cargar *payload* para ello se debe ejecutar el comando "set payload Windows/meterpreter/reverse_tcp".

c. Agregar la IP de la máquina objetivo y del puerto que se quiere atacar. Para esta guía se va a usar el equipo con la IP 192.168.0.7 y el puerto 445.
d. Agregar la IP de la máquina atacante que para esta guía es 192.168.0.11 y el puerto por donde se va a realizar el ataque es el 4000.
e. Una vez se ha realizado la configuración anterior se debe ejecutar el comando “exploit” para iniciar el ataque.

7.10.2. Resultados

El ataque originalmente pretende iniciar sesión en la máquina victima con el fin de tener acceso tanto a la información como a la ejecución de comandos. El ataque puede cumplir dos objetivos: el primero descrito anteriormente como el acceso al equipo víctima, el segundo objetivo es generar una denegación de servicios en el equipo objetivo ocasionando un reinicio inesperado provocando un error en la ejecución del sistema operativo. Se evidencia en el equipo víctima los efectos del ataque desde la máquina con el sistema Kali Linux. Inesperadamente el sistema operativo Windows Server 2008 se bloquea y se genera un reinicio en el sistema.

8. Conclusiones

A partir de las prácticas y ejercicios realizados en la detección de vulnerabilidades y analizar los ataques, se refiere a continuación una síntesis de cada una de las herramientas así;

Nmap

Con la presente herramienta se identifica y se permite conocer los equipos que se encuentran en una red, y reconocer entre ellos los que no pertenezcan a la misma red de equipos y que probablemente sean personas mal intencionadas que quieran adquirir información o manipular equipos de la misma red.
Por otra parte, permite igualmente conocer qué puertos se encuentran abiertos y qué tipo de servicios se exponen por cada uno de estos.

Ahora bien, desde el lado de un usuario mal intencionado o atacante, se puede obtener información relevante de los equipos que conforman una red, lo cual, está información genera ventajas pues aporta debilidades que podría explotar para poder ingresar a este sistema y obtener la información.

Nessus

Desde la perspectiva de la seguridad informática, esta herramienta permite conocer las debilidades o vulnerabilidades de un sistema, que puedan ser aprovechadas por un atacante. Igualmente, esta herramienta permite conocer cuáles son los puntos débiles de un sistema que deben ser fortalecidos, es decir, puede funcionar como una herramienta de prevención.

Metasploid Framework

Se evidenció la potencia que tiene la herramienta para aprovechar vulnerabilidades dentro de un Sistema Operativo. En este tipo de ejecuciones se debe realizar solo en un ambiente controlado, puesto que, puede ocasionar daños en los sistemas en un ambiente de producción.

9. Recomendaciones

Este ejercicio de hacking ético pretende orientar de forma básica sobre la forma adecuada en la que se debe realizar las pruebas de seguridad sobre una red o un equipo en específico y resaltar la importancia de conocer las debilidades de una red o sistema y como poder fortalecerlo a partir de las vulnerabilidades conocidas.

10. Resultados

Teniendo en cuenta que el presente documento es un trabajo monográfico y no un proyecto de investigación, no se presentan metodologías trabajadas ni aspectos metodológicos. Por ende, como resultado se puede indicar la realización del Manual Básico de pruebas de Intrusión, que se presenta en el capítulo III del presente documento. Sin embargo, se presenta a continuación para tal fin se expone a continuación las principales actividades que se desarrollaron para lograr los objetivos y específicos y a su vez el objetivo general del proyecto. Las principales actividades fueron las siguientes:

a. Se realizó minería textual empezando por las herramientas de pentesting, en las que se incluyen Nessus, Metasploit y Nmap.
b. Se realizó una comparativa en la instalación, manejo y ejecución de las herramientas mencionadas anteriormente.
c. Se hizo una creación de laboratorio controlado para la aplicación de las pruebas de cada una de las herramientas.
d. En el mismo laboratorio, se hizo la ejecución de cada una de las herramientas de pentesting seleccionadas.
e. Basada en las experiencias anteriores de aplicación, se construye un manual con los pasos ejecutados para realizar las pruebas de intrusión.

11. Discusión

Inicialmente, en este manual se presenta una metodología básica para realizar pruebas de intrusión dentro de una red de equipos informáticos, con el fin de validar el nivel de seguridad de un sistema operativo según la versión que se está validando. A manera propositiva, en el manual se puede ampliar con el fin de validar la seguridad en las aplicaciones, bien sea en web o en dispositivos móviles, ampliando los pasos ejecutados para cubrir aspectos adicionales de la seguridad como sesiones de usuario y seguridad en la comunicación con otras aplicaciones.

12. Bibliografía

Alonso Cebrián, J., Díaz Sáez, V., Guzmán Sacristán, A., Laguna Durán, P., & Martín Bailón, A. (2018). Auditoría y desarrollo seguro. Retrieved from https://www.exabyteinformatica.com/uoc/Informatica/Seguridad_en_bases_de_datos/Seguridad_en_bases_de_datos_(Modulo_4).pdf

Barreto Cuitiva, J. (2018). *Diseño de manual de diagnóstico y prevención de vulnerabilidades en redes de datos para pymes* (Especialización). Universidad Nacional Abierta y a Distancia UNAD.

Benavides Arias, A., & Velásquez Mayorga, J. (2015). *Prueba de intrusión al sistema operativo Windows Server 2003 de una empresa del sector financiero* (Especialización). Universidad Nacional Abierta y a Distancia.

Blanco López, A. (2018). *Explotación de sistemas Windows y pentesting* (Magister). Universitat Oberta de Catalunya.

Bortnik, S. (2013). Pruebas de penetración para principiantes: 5 herramientas para empezar. *Seguridad, Cultura De Prevención Para TI*, (251). Retrieved from https://revista.seguridad.unam.mx/numero-18/pruebas-de-penetracion-para-principiantes-5-herramientas-para-empezar

Catoira, Fernando. (2013). “Pruebas de penetración para principiantes: explotando una vulnerabilidad con metaexploid framework”. Revista: Seguridad. Cultura de prevención para TI. No. 19. (en línea) Disponible en: https://revista.seguridad.unam.mx/sites/revista.seguridad.unam.mx/files/revistas/pdf/num19_seguridad.pdf

Deraison, R., & Gula, R. (2018). Using Nessus to Detect Wireless Access Points. Retrieved from http://docs.huihoo.com/nessus/wap-id-nessus.pdf

Docs.tenable.com. (2019). *Security Warnings (Nessus)*. (En línea) Disponible en: https://docs.tenable.com/nessus/8_2/Content/SecurityWarnings.htm [Accesado 10 marzo 2019].

Gomes Martinelo, C., & Bellezi, M. (2014). Análisis de vulnerabilidades con OpenVas y Nessus. *Tecnologías, Infraestructura E Software*, (v 3 n 1), 34-44.

Guevara, A., Hacking ético: mítos y realidades. (2018). *Seguridad, Cultura De Prevención Para TI*, (12), 8. Retrieved from http://www.ru.tic.unam.mx/bitstream/handle/123456789/1761/63.pdf?sequence=1&isAllowed=y

Guía de referencia de Nmap (Página de manual) |. (2018). Retrieved from https://nmap.org/man/es/index.html#man-description

Retrieved from http://www.revistatis.dc.ufscar.br/index.php/revista/article/view/74/68

Hernández Saucedo, J. (2018). Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web. Retrieved from http://www.redalyc.org/html/5122/512251501005/

Kanclirz, J., & Baskin, B. (2008). *Netcat power tools* (pp. 3-4). Burlington, MA: Syngress Pub.

Mendoza, M. (2018). Cómo utilizar OpenVAS para la evaluación de vulnerabilidades. Retrieved from https://www.welivesecurity.com/la-es/2014/11/18/como-utilizar-openvas-evaluacion-vulnerabilidades/

Metasploit.help.rapid7.com. (2019). *Metasploit Framework*. (en línea) Disponible en: https://metasploit.help.rapid7.com/docs/msf-overview [Accesado 9 marzo 2019].

Metasploit.help.rapid7.com. (2019). *Working with Payloads*. (En línea) Disponible en: https://metasploit.help.rapid7.com/docs/working-with-payloads [Accesado 9 marzo 2019].

Rouse, M., ¿Qué es Prueba de penetración (pen test)? - Definición en WhatIs.com. (2018). Retrieved from https://searchdatacenter.techtarget.com/es/definicion/Prueba-de-penetracion-pen-test

Ramírez Restrepo, J., & Ávila Pardo, W. (2018). *Escaneo de vulnerabilidades al servidor principal de la empresa caso de estudio* (Especialización). Universidad Nacional Abierta y a Distancia.

Tools.kali.org. (2018). Wfuzz Package Description. (en línea). Disponible en: https://tools.kali.org/web-applications/wfuzz [Accesado 9 marzo 2019]

Veloz, J., Alcivar, A., Salvatierra, G., & Silva, C. (2017). Ethical Hacking, una metodología para descubrir fallas de seguridad en sistemas informáticos mediante la herramienta KALI-LINUX. *Revista De Tecnología De La Informática Y Las Comunicaciones*, (1), 3-4. Retrieved from https://revistas.utm.edu.ec/index.php/Informaticaysistemas/article/view/194/156

Wolfgang, M. (2018). Host Discovery with nmap. Retrieved from https://havel.mojeservery.cz/wp-content/uploads/2015/10/nmap-discovery-howto-2002.pdf