UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA
LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

TITULO O TEMA:

Auditoría de Seguridad en una Red Local

INTEGRANTES

N°	Apellidos	Nombres	Carnet	Participación
1	Guerrero Granada	Carlos Mauricio	GG01136478	
2	Menjívar Wiler	Riquelme Aimar	MW23371	
3	Pérez Juárez	Miguel Alexander	PJ23424	
4	Puentes Monge	Adiel Enoc	PM01136619	
5	Rodas Alvarado	Mario Enrique	RA23333	
6	Servellón Vázquez	Allan Bryan	SV2263	
			Total	100%

DOCENTE:

Rafael Antonio Díaz Palacios

Redes II

San Salvador, El Salvador, 22 de noviembre de 2024

Contenido

Introducción

En la actualidad el incremento de las amenazas cibernéticas exige que se implementen medidas de seguridad robustas para la protección de los sistemas informáticos

En el presente documento se presenta una guía práctica para realizar auditorías de seguridad en redes locales. Mediante el empleo de herramientas de escaneo de renombre como Nmap, Nessus y Metasploit, se busca identificar de las vulnerabilidades que podrían comprometer la integridad y confidencialidad de un sistema informático.

Se brinda información acerca de cada una de las herramientas mencionadas, así como su aplicación en el proyecto, detallando las configuraciones realizadas para su funcionamiento adecuado.

Además, se incluye un manual de usuario práctico que facilita la implementación de las técnicas descritas. A través de este trabajo se busca contribuir al fortalecimiento de las evaluaciones de seguridad informática en las organizaciones y tomar medidas correctivas para mitigar los riesgos identificados.

Objetivos

Objetivo General

Implementar un proceso de escaneo de vulnerabilidades en redes locales para identificar y mitigar riesgos de seguridad

Objetivos Específicos

- Investigar el funcionamiento y las características de herramientas como Nmap, Nessus y Metasploit
- Estudiar las mejores prácticas para identificar y mitigar vulnerabilidades en redes informáticas mediante herramientas de escaneo
- Realizar una auditoría de seguridad en una red local utilizando las herramientas seleccionadas identificando las vulnerabilidades presentes para su mitigación.

Marco Teórico

Una auditoría de red es un proceso fundamental para garantizar la fiabilidad, seguridad y eficiencia de un sistema informático en una organización. Este consiste en examinar los componentes que componen la red como pueden ser servidores, políticas de seguridad y prácticas operativas con la finalidad de encontrar vulnerabilidades o posibles riesgos.

Si bien sabemos en la actualidad en su mayoría las empresas y organizaciones mantienen su información de forma digital, mayormente se guarda todo en servidores ya sea de forma local o en la nube, pero desde el momento que están conectados a una red estos se vuelven vulnerables al no contar con la debida seguridad lo que puede dejar a la deriva información confidencial o personal en algunos casos hasta podría afectar las operaciones que realiza una organización.

¿Qué es Pentesting? Es una práctica mediante la cual se puede poner a prueba la reacción de una red ante un ataque informático. La práctica es novedosa y consiste en determinar el alcance de los fallos de seguridad de un sistema atacando diferentes entornos o sistemas permitiendo descubrir todas las áreas y encontrar esas vulnerabilidades que en un ataque real sería un gran problema para la organización.

Existen 3 formas al realizar un Pentesting:

- **White Box:** Es el ataque más completo en le que el auditor conoce todos los datos de la organización, como la estructura, contraseñas, IPs, Firewalls, etc.
- **Black Box:** Es el ataque más real en la que el auditor tiene pocos o casi nula información sobre la organización, por lo que se presenta como un ciberatacante

- **Grey Box:** Es una mezcla de los anteriores casos, el auditor tiene cierta información de la organización para el ataque, suele ser el más recomendado debido que de esta forma se pueden verificar más áreas de vulnerabilidad.

Entre las herramientas más comunes para auditoría de redes, podemos destacar las siguientes:

Nmap: Es la abreviatura de Network Mapper, tal como su nombre lo indica nos permite escanear y detectar dispositivos se están ejecutando en una red, así como direcciones IP y puertos abiertos. Esto le permite encontrar vulnerabilidades con mayor facilidad ya que reconoce los dispositivos (Servidores, enrutadores, dispositivos móviles) también servicios (servidores DNS, servidores web), así como el sistema operativo que se ejecuta en cada dispositivo. Evidentemente es una herramienta que gracias a sus características es considerada las más famosa y muchos profesionales de la materia la prefieren. Cabe mencionar que es gratuita.

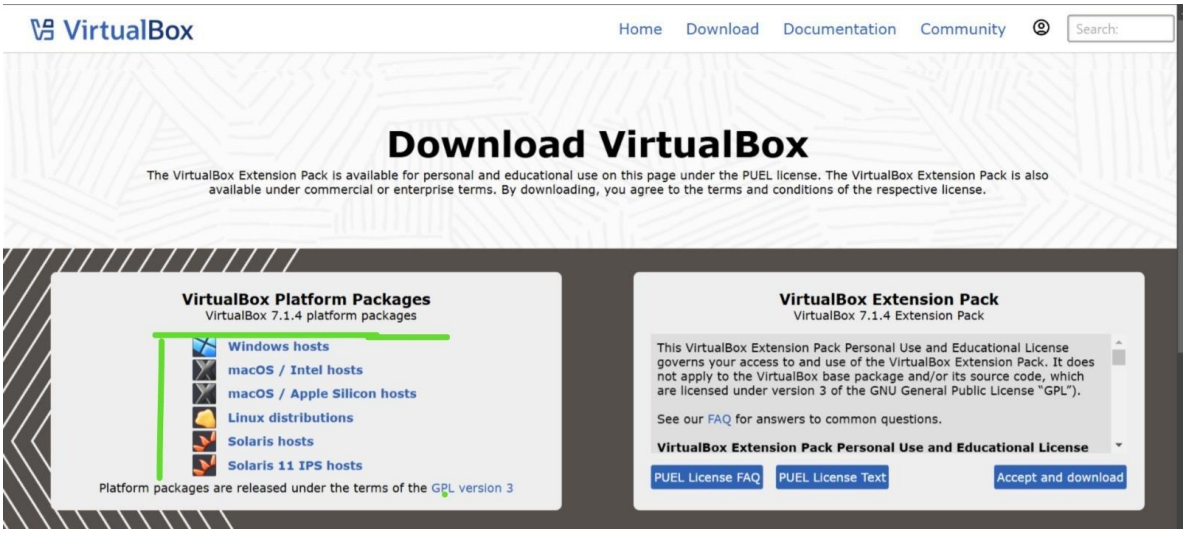
Nessus: Al igual que Nmap es un escáner de redes que de igual forma nos permite encontrar vulnerabilidades mediante la revisión servidores, dispositivos móviles, firewall y hasta routers. Una vez el escaneo finaliza Nessus lo presenta en un informe donde se detalla si encuentra vulnerabilidades, nivel de gravedad y brinda sugerencias para poder corregirlas; permite simular un ataque informático y eso brinda una mejor vista de la efectividad de las defensas de la organización. Es necesario mencionar que esta herramienta no es gratuita su precio ronda desde $4,000 a $6,000 anualmente.

Metasploit: Considerada una herramienta de hacking ético permite evaluar la seguridad de un sistema informático, incluye una amplia variedad de módulos, exploits y payloads lo cual facilita la tarea al momento de realizar la auditoria, ya que, con la combinación de sus

módulos, que permite encontrar vulnerabilidades o programarle una acción en el sistema, sus exploits, aprovechando la vulnerabilidad para conseguir acceso al sistema informático, y sus payloads para ejecutar acciones en el sistema una vez ha ingresado al mismo. Como podemos notar es una herramienta muy completa y es gratuita al igual que Nmap, pero Metasploit a la vez que escanea la red permite ir ejecutando acciones lo cual hace más eficaz el vulnerar el sistema.

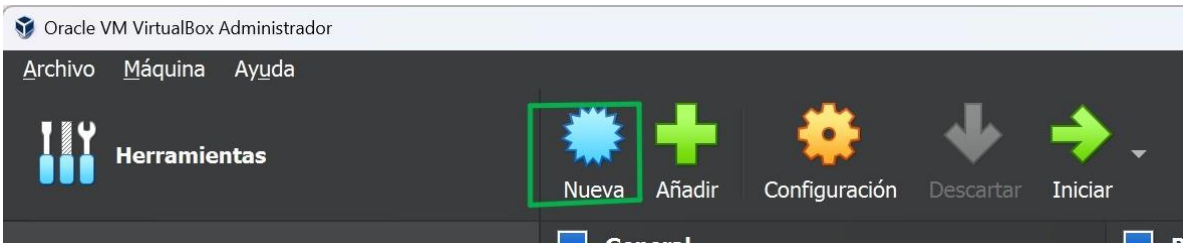
Implementación y pruebas

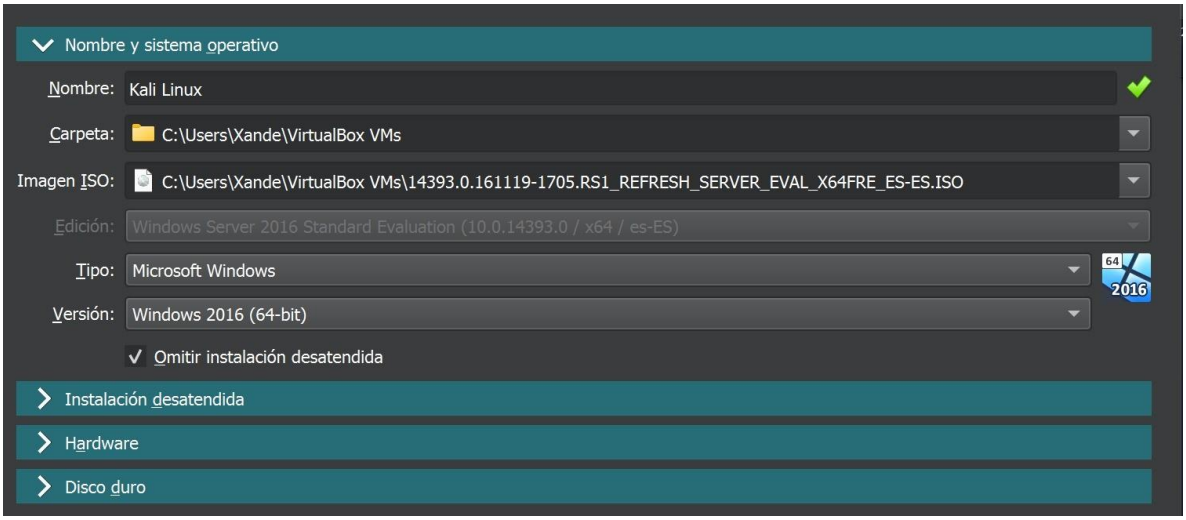
Se realiza primero la instalación de Virtual Box para la práctica, se descarga de acuerdo al sistema operativo en la computadora.



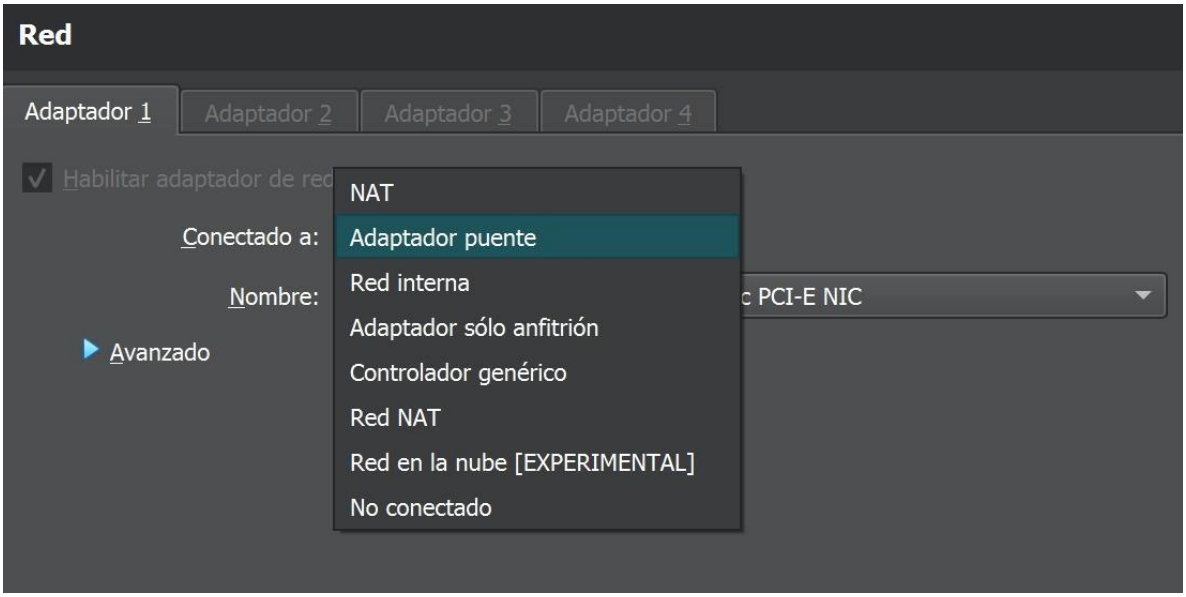
Se descarga la ISO de Kali Linux desde su sitio oficial https://www.kali.org/get-kali/#kali-platforms

Preparamos la máquina virtual para la instalación del Sistema Operativo Kali Linux





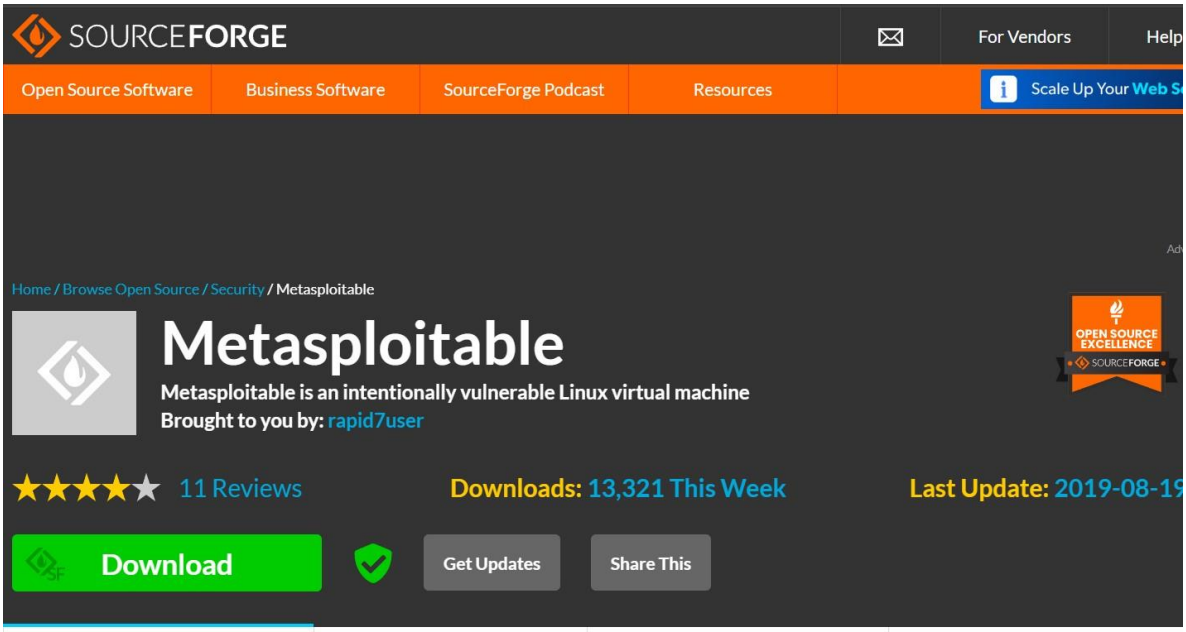
Configuramos el tipo de red utilizando Adaptador puente



Se inicia la máquina virtual y sigue las instrucciones en pantalla para instalar Kali Linux.

Para la simulación del ciber ataque descargamos Metasploit, es un sistema para poder simular ataques, tiene fallos y vulnerabilidades apropósito, con la finalidad de poder practicar en un entorno controlado.

Se realiza la descarga desde el siguiente enlace https://sourceforge.net/projects/metasploitable/



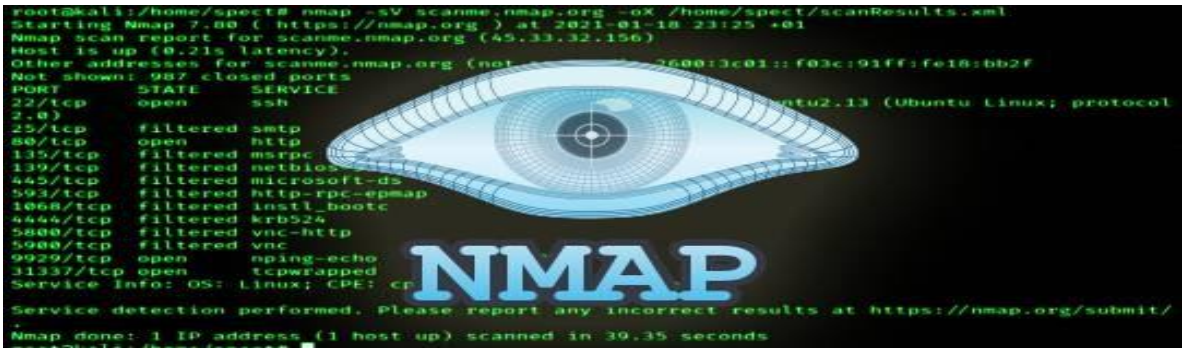
Realizamos el mismo procedimiento de instalación que la máquina virtual de Kali Linux y estamos listo para comenzar la auditoria.

Herramientas a utilizar para realizar el Pentesting

Kali Linux ya tiene integradas muchas herramientas para el hacking ético, como por ejemplo Nmap y Metasploit**.**

NMAP

Herramienta de código abierto para exploración de red y auditoría de seguridad. Se diseñó para analizar rápidamente grandes redes, aunque funciona muy bien contra equipos individuales.



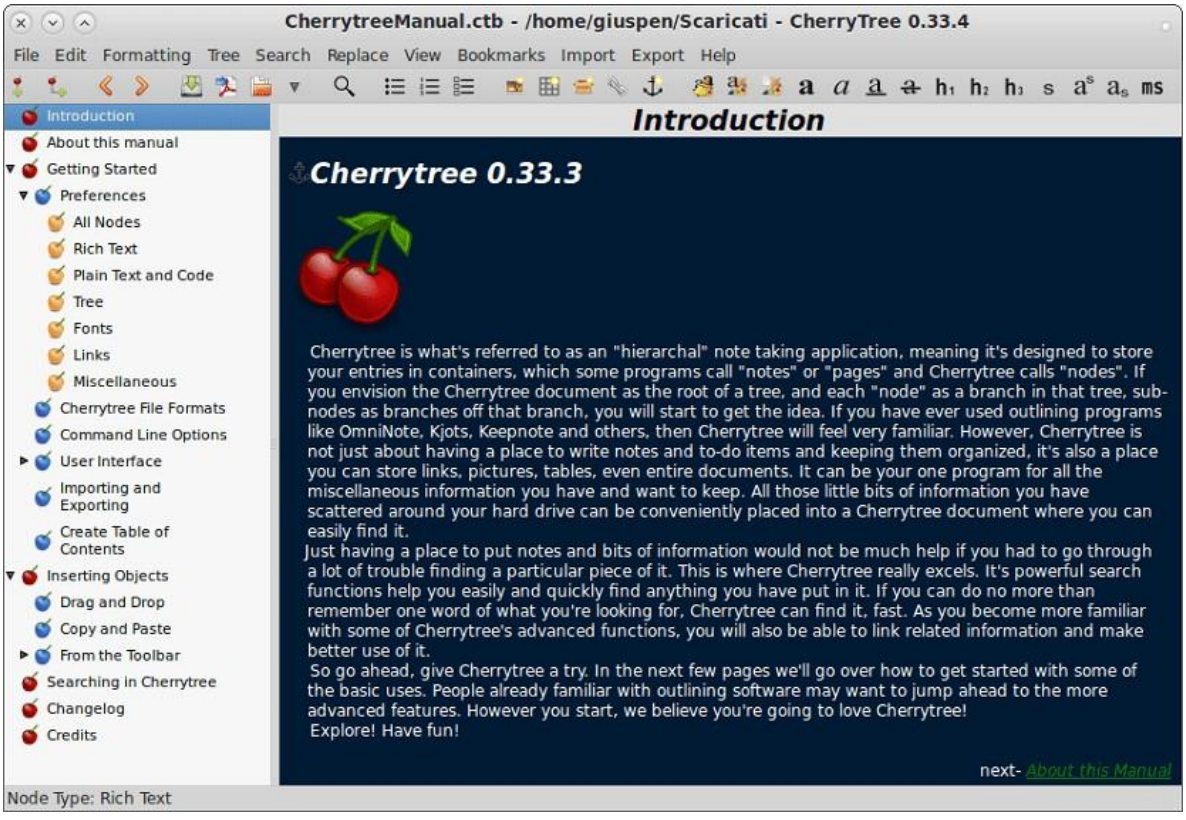
METASPLOIT

Herramienta muy poderosa que pueden usar tanto los cibercriminales como los hackers éticos para investigar vulnerabilidades sistemáticas en redes y servidores. Como es un marco de código abierto, se puede personalizar fácilmente y usar con la mayoría de los sistemas operativos.

Mediante Kali Linux podemos acceder a este mediante el siguiente comando:

CHERRYTREE

Es una herramienta muy práctica que nos permitirá tomar notas y guardar cualquier tipo de texto al que podremos acceder para revisarlo o seguir tomando notas en cualquier momento. Esta herramienta la usaremos para documentar nuestra auditoria paso a paso.



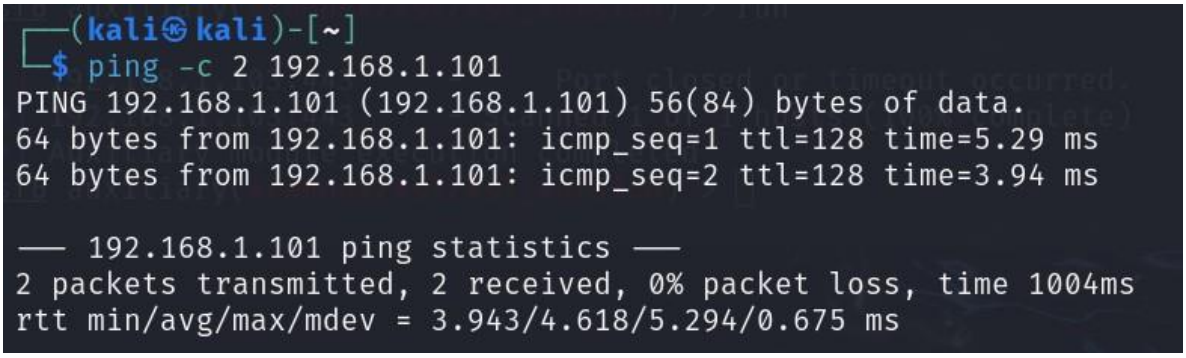
Instalación

Con este comando se inicializar la descarga de nuestro paquete de Cherrytree

Implementando NMAP

Escaneando puertos TCP

El siguiente comando escaneara todos los puertos tcp del dispositivo de la dirección 192.168.1.101



¿Qué podemos observar?

Host activo: El host está encendido y responde a las solicitudes de Nmap

Puertos abiertos: Se han detectado varios puertos TCP abiertos.

- **135/tcp: msrpc** (Microsoft RPC): Este servicio es utilizado por muchos componentes de Windows para comunicarse entre sí.
- **139/tcp: NetBIOS**: NetBIOS es un protocolo de red utilizado para compartir recursos en redes pequeñas.
- **445/tcp: microsoft-ds** (Microsoft Directory Services): Este servicio proporciona un directorio centralizado para almacenar información sobre usuarios, equipos y otros objetos de la red.
- **5357/tcp: wsdapi** (Web Services for Devices API): Este servicio se utiliza para descubrir y administrar dispositivos en una red.

Implicaciones de seguridad

La presencia de estos puertos abiertos y servicios puede indicar posibles vulnerabilidades:

- **MSRPC:** Es vulnerable a diversos ataques, incluyendo ataques de día cero.
- **NetBIOS:** Los servicios NetBIOS pueden ser explotados para realizar ataques de denegación de servicio y ataques de fuerza bruta.
- **SMB (Server Message Block):** El protocolo SMB, que utiliza el puerto 445, es vulnerable a múltiples exploits, como EternalBlue y SMBGhost.

Recomendaciones:

Actualizar el sistema operativo y las aplicaciones: Es fundamental mantener el sistema operativo y todas las aplicaciones instaladas actualizadas con los últimos parches de seguridad.

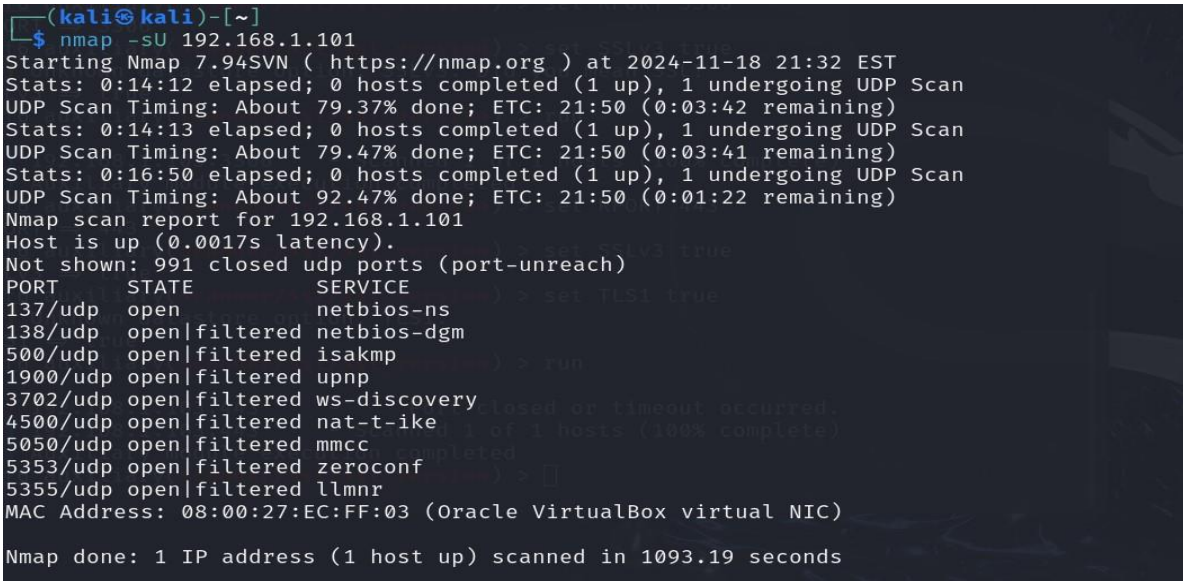
Bloquear puertos innecesarios: Si algunos de los puertos abiertos no son necesarios, se recomienda bloquearlos para reducir la superficie de ataque.

Implementar un firewall: Un firewall puede ayudar a proteger la red al filtrar el tráfico entrante y saliente. Utilizar un IDS: Un sistema de detección de intrusiones puede ayudar a detectar y responder a ataques en tiempo real.

Realizar auditorías de seguridad regulares: Es importante realizar auditorías de seguridad periódicas para identificar y corregir cualquier vulnerabilidad.

Escaneando puertos UDP

El siguiente comando escaneara todos los puertos UDP del dispositivo. Muchos servicios críticos como DNS, DHCP y TFTP utilizan UDP. Un escaneo UDP permitirá identificar estos servicios y evaluar posibles vulnerabilidades.



Posibles implicaciones de seguridad:

La presencia de ciertos puertos abiertos y servicios puede indicar posibles vulnerabilidades:

- UPnP (Universal Plug and Play): Este servicio puede ser utilizado por atacantes para acceder a la red interna.
- WS-Discovery: Este protocolo puede ser explotado para descubrir dispositivos en la red y lanzar ataques. NetBIOS: Los servicios NetBIOS pueden ser vulnerables a diversos ataques.

Recomendaciones

1. Investigar los servicios: Para evaluar el riesgo asociado a los servicios detectados, es necesario investigar a fondo cada uno de ellos y buscar cualquier vulnerabilidad conocida.
2. Bloquear puertos innecesarios: Si algunos de los puertos abiertos no son necesarios, se recomienda bloquearlos para reducir la superficie de ataque.
3. Mantener los sistemas actualizados: Es fundamental mantener todos los sistemas operativos y aplicaciones actualizados con los últimos parches de seguridad.
4. Implementar un firewall: Un firewall puede ayudar a proteger la red al filtrar el tráfico entrante y saliente.

```
┌──(kali㉿kali)-[~]
└─$ nmap --script "vuln" -p135 192.168.150.29
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 21:05 EST
Nmap scan report for 192.168.150.29
Host is up (0.0021s latency).

PORT    STATE SERVICE
135/tcp open  msrpc
MAC Address: 08:00:27:EC:FF:03 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 25.15 seconds
┌──(kali㉿kali)-[~]
└─$
```

```
┌──(kali㉿kali)-[~]
└─$ nmap --script "vuln" -p445 192.168.150.29
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 21:07 EST
Nmap scan report for 192.168.150.29
Host is up (0.0036s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 08:00:27:EC:FF:03 (Oracle VirtualBox virtual NIC)

Host script results:
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ER
ROR
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_smb-vuln-ms10-054: false

Nmap done: 1 IP address (1 host up) scanned in 15.56 seconds
```

Vulnerabilidades:

samba-vuln-cve-2012-1182: No se pudo establecer una conexión para verificar esta vulnerabilidad. smb-vuln-ms10-061: Tampoco se pudo establecer una conexión para verificar esta vulnerabilidad. smb-vuln-ms10-054: Se determinó que esta vulnerabilidad no está presente.

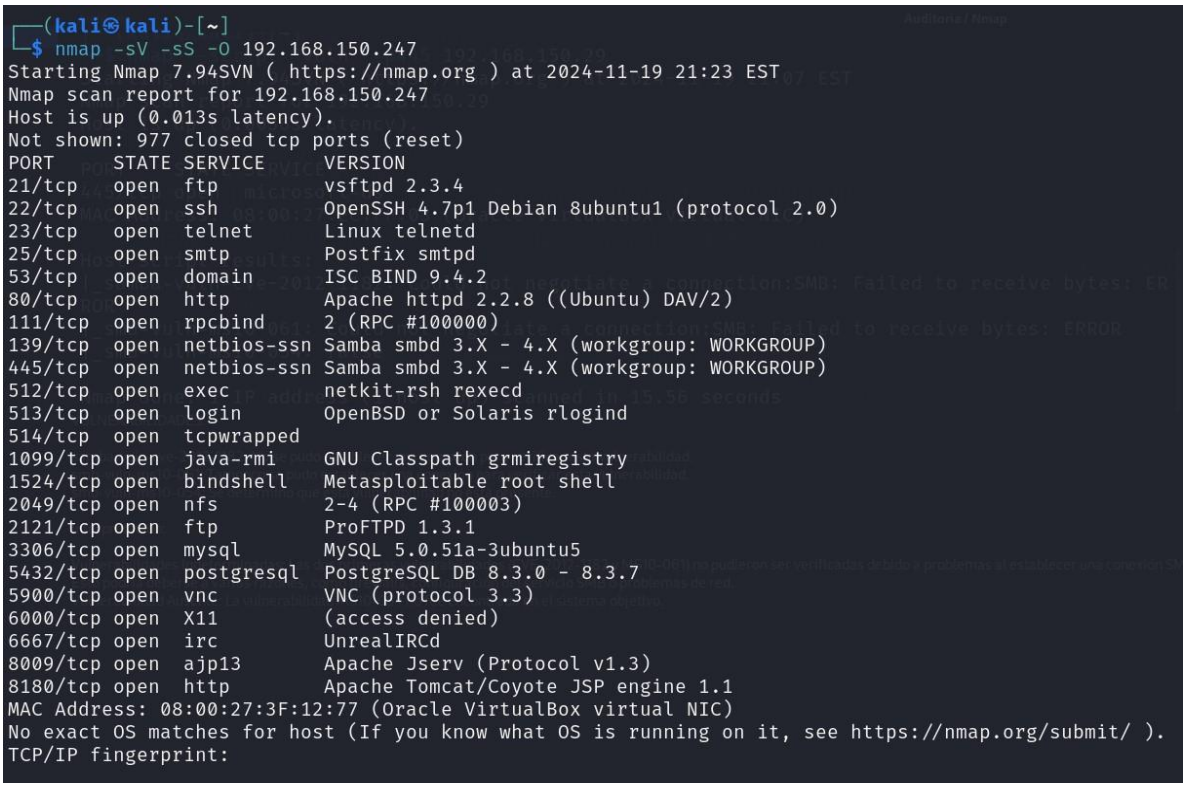
Interpretación:

Vulnerabilidades Indeterminadas: Las dos primeras vulnerabilidades (CVE-2012-1182 y MS10-061) no pudieron ser verificadas debido a problemas al establecer una conexión SMB.

Esto podría deberse a varias razones, como firewalls, configuración del servicio SMB o problemas de red. Vulnerabilidad Ausente: La vulnerabilidad MS10-054 no fue encontrada en el sistema objetivo.

Se intento vulnerar todos puertos abiertos no encontrando vulnerabilidades a las cuales explotar.

Segunda auditoria a un sistema Metasploitable



```
OS:SCAN(V=7.94SVN%E=4%D=11/19%OT=21%CT=1%CU=38698%PV=Y%DS=1%DC=D%G=Y%M=0800
OS:27%TM=673D481F%P=x86_64-pc-linux-gnu)SEQ(SP=CA%GCD=1%ISR=D1%TI=Z%CI=Z%II
OS:=I%TS=5)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7
OS:%O5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=16A0%W2=16A0%W3=16A0%W4=16A0%W5=16A0%
OS:W6=16A0)ECN(R=Y%DF=Y%T=40%W=16D0%O=M5B4NNSNW7%CC=N%Q=)T1(R=Y%DF=Y%T=40%S
OS:=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=Y%T=40%W=16A0%S=O%A=S+%F=AS%O=M5B4
OS:ST11NW7%RD=0%Q=)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%
OS:T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD
OS:=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL
OS:=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe
:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 27.28 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ nmap --script "vuln" -p21 192.168.150.247
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 21:31 EST
Nmap scan report for 192.168.150.247
Host is up (0.023s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2011-2523  BID:48539
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
|     References:
|       https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_
234_backdoor.rb
|       https://www.securityfocus.com/bid/48539
|       http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
MAC Address: 08:00:27:3F:12:77 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 11.91 seconds
```

Vulnerabilidad critica:

Se ha detectado una vulnerabilidad crítica en el servidor FTP vsftpd, específicamente la CVE-2011-2523. Esta vulnerabilidad permite a un atacante remoto ejecutar comandos con privilegios de root:

- **Exploitable:** Nmap indica claramente que esta vulnerabilidad es exploitable, lo que significa que un atacante podría aprovecharla para obtener acceso completo al sistema.
- **vsftpd vulnerable:** La versión de vsftpd que está ejecutando el servidor (2.3.4) es vulnerable a una falla de seguridad que permite a un atacante remoto ejecutar comandos con privilegios de root.

Riesgo Elevado: Esta vulnerabilidad representa un riesgo extremadamente alto para la seguridad del sistema, ya que un atacante podría tomar el control completo del servidor.

Recomendaciones:

Acción Inmediata: Es crucial abordar esta vulnerabilidad lo antes posible. Un atacante podría aprovecharla para robar datos, instalar malware o incluso tomar el control completo del sistema.

Actualizar vsftpd: La primera medida a tomar es actualizar vsftpd a una versión que no sea vulnerable a CVE-2011-2523. Consulta la documentación oficial de vsftpd para obtener instrucciones específicas sobre cómo actualizar.

Deshabilitar FTP: Si no es estrictamente necesario, considera deshabilitar el servicio FTP por completo, ya que existen protocolos más seguros para transferir archivos (como SFTP o FTPS).

Aplicar Parches de Seguridad: Asegúrate de que todos los sistemas estén actualizados con los últimos parches de seguridad.

Implementar un Firewall: Configura un firewall para bloquear el acceso no autorizado al puerto 21.

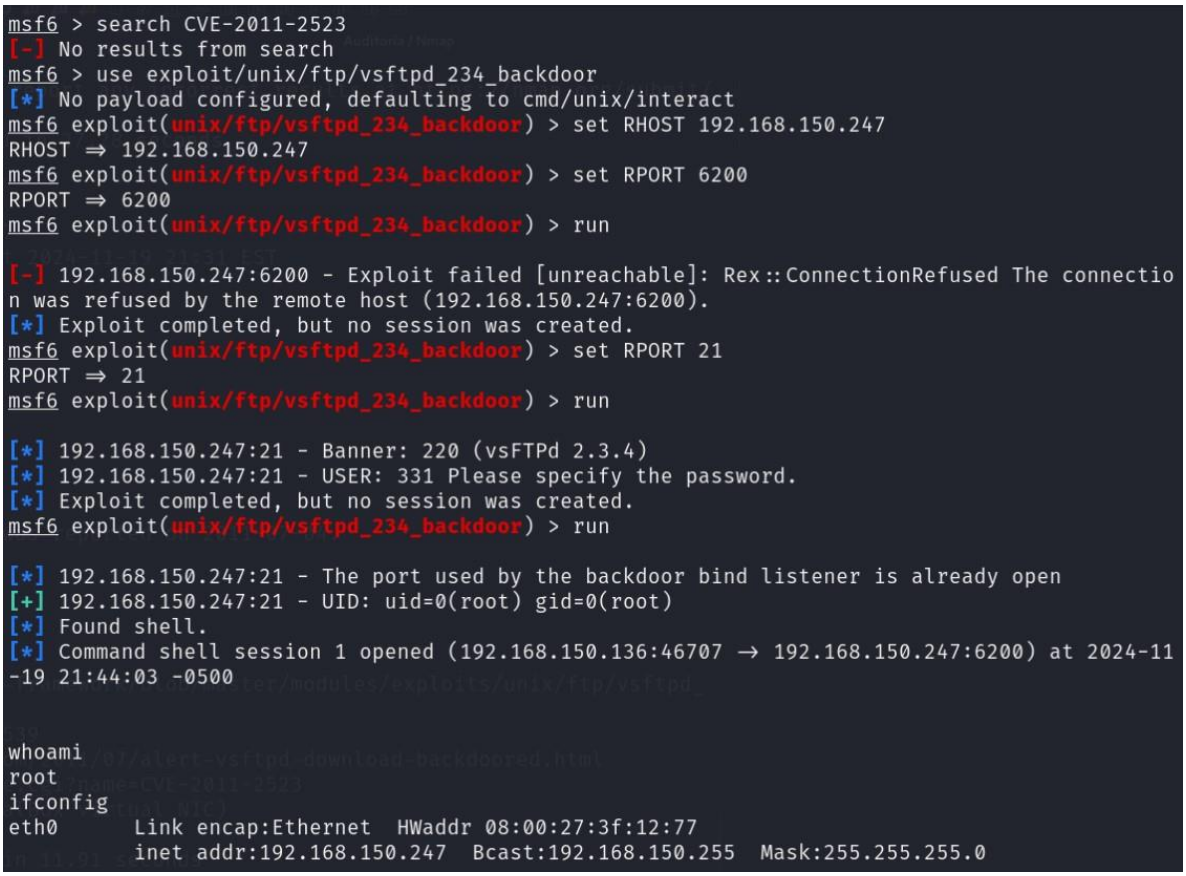
Monitorear Actividad: Utiliza herramientas de monitoreo de seguridad para detectar cualquier actividad sospechosa en el sistema.

Importante

Ética: La explotación de sistemas sin autorización es ilegal y puede tener graves consecuencias legales. Solo realiza estas acciones en entornos de laboratorio autorizados y con fines educativos.

Seguridad: Siempre ten en cuenta que, al obtener acceso a un sistema, estás asumiendo un riesgo. Protege tu propia identidad y evita dejar rastro.

Implementando Metasploit



Explicación de los parámetros:

1. **msfconsole:** Inicia la interfaz de línea de comandos de Metasploit.
2. **use exploit/unix/ftp/vsftpd_234_backdoor:** Selecciona el módulo de explotación específico para la vulnerabilidad CVE-2011-2523
3. **set RHOST <IP_del_objetivo>:** Reemplaza <IP_del_objetivo> con la dirección IP del sistema vulnerable.
4. **set RPORT 6200**: El puerto 6200 es el puerto por defecto utilizado por el backdoor.
5. **run:** Ejecuta el exploit.

Dirección IP y Puertos: Se muestra la dirección IP del sistema atacado (192.168.150.247) y los puertos involucrados en la conexión (21 para el servicio FTP y 6200 para el backdoor).

Credenciales: Se indica que la sesión se obtuvo con los privilegios de root (UID=0 y GID=0), lo que significa que el atacante tiene control total sobre el sistema.

Fecha y Hora: Se registra la fecha y hora en que se estableció la sesión.

Cronograma de Actividades

N	Actividad	Octubre				Noviembre			
		Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4
1	Designación del tema del proyecto								
2	Asignación de tareas a cada integrante								
3	Investigación teórica sobre las herramientas								
4	Entrega de primer avance								
5	Implementación de NMAP								
6	Implementación de Metasploit								
7	Finalización de la documentación								
8	Entrega y defensa del proyecto final								

Presupuesto

Concepto	Descripción	Costo Estimado (USD)	Observaciones
Personal	Estudiantes	0	
Herramientas	VirtualBox	Gratuito	Software de virtualización libre.
	Kali Linux	Gratuito	Distribución Linux enfocada en pruebas de penetración.
	Metasploit, Nmap,	Gratuito	Herramientas de hacking ético de código abierto.
Infraestructura	Conexión a internet	60 por mes	Dependerá del consumo de datos y del proveedor de internet.
	Electricidad	100 por mes	
Otros	Imprevistos	10% del total	Para cubrir gastos no contemplados.
Total		$160.00	

Bibliografía

Estefanía Domínguez de la Iglesia. (2020, 26 de febrero). ¿Qué es el Pentesting? Campus Internacional de Ciberseguridad. https://www.campusciberseguridad.com/blog/item/139-que-es-el-pentesting

BlackeyeB. (2023, 23 de abril). Qué es Nmap y cómo usarlo: Un tutorial para la mejor herramienta de escaneo de todos los tiempos. freeCodeCamp.org. https://www.freecodecamp.org/espanol/news/que-es-nmap-y-como-usarlo-un-tutorial-para-la-mejor-herramienta-de-escaneo-de-todos-los-tiempos/

Vyas, K. (2023, 22 de marzo). Nessus vs. Nmap Vulnerability Scans: Which is Best for You? | Datamation. Datamation. https://www.datamation.com/security/nessus-vs-nmap

Sepulveda, M. (2023, 23 de febrero). Que es Nessus y como utilizarlo. Ciberseguridad club. https://ciberseguridad.club/que-es-nessus-y-como-utilizarlo/

Sepulveda, M. (2023b, 25 de febrero). Metasploit Herramienta de hacking ético - El Club de la Ciberseguridad. El Club de la Ciberseguridad. https://ciberseguridad.club/metasploit-herramienta-de-hacking-etico/

Anexos

Guía de Comandos para Auditoría de Seguridad en Red Local

1. **Reconocimiento Inicial con Nmap**
   ### Escaneo básico de red
   bash
   # Escaneo básico de la red
   nmap -sn 192.168.1.0/24 # Descubre hosts activos en la red

   # Escaneo completo de puertos
   nmap -sS -sV -O 192.168.1.0/24 # Escaneo sigiloso con detección de servicios y SO

   # Escaneo detallado de un host específico
   nmap -A -T4 -p- 192.168.1.x # Escaneo agresivo de todos los puertos

   ### Detección de vulnerabilidades con Nmap
   bash
   # Escaneo de vulnerabilidades
   nmap --script vuln 192.168.1.x # Ejecuta scripts de vulnerabilidades

   # Escaneo de malware
   nmap --script malware 192.168.1.x # Busca indicadores de malware

   # Fuerza bruta de servicios
   nmap --script brute 192.168.1.x # Intenta fuerza bruta en servicios comunes

2. **Análisis con Nessus**
   1. Acceder a Nessus (típicamente en https://localhost:8834)
   2. Crear un nuevo escaneo
      - Basic Network Scan para escaneo general
      - Advanced Scan para configuración detallada

   ### Configuraciones recomendadas:

   Name: Audit_[fecha]
   Targets: 192.168.1.0/24
   Policy: Internal Network Scan

3. **Pruebas de Penetración con Metasploit**
   ### Iniciar Metasploit
   bash
   # Iniciar la consola de Metasploit
   msfconsole

   # Configuración básica
   workspace -a red_local # Crear nuevo espacio de trabajo

   ### Comandos básicos de reconocimiento
   bash
   # Escaneo de servicios
   db_nmap -sS -sV 192.168.1.0/24

   # Búsqueda de exploits
   search type:exploit platform:windows

   # Uso de un exploit
   use exploit/windows/smb/ms17_010_eternalblue
   set RHOSTS 192.168.1.x
   exploit

4. **Análisis de Tráfico con Wireshark**
   ### Captura de tráfico
   bash
   # Desde terminal (tshark)
   tshark -i eth0 -w captura.pcap # Captura tráfico en archivo

   ### Filtros útiles en Wireshark

   # Filtros comunes
   ip.addr == 192.168.1.x # Tráfico de un host específico
   http # Solo tráfico HTTP
   tcp.port == 80 # Tráfico en puerto específico

5. **Evaluación de Contraseñas**
   ### John The Ripper
   bash
   # Crackeo básico de contraseñas
   john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

```
# Modo incremental
john --incremental hash.txt
```

6. **Reportes y Documentación**

```
### Crear reporte con documentación
bash
# Exportar resultados de Nmap
nmap -sV 192.168.1.0/24 -oA reporte_nmap

# Convertir a formato HTML
xsltproc reporte_nmap.xml -o reporte_nmap.html
```

7. **Verificación de Configuraciones**

```
### Revisión de servicios activos
bash
# Linux
netstat -tuln    # Muestra puertos abiertos
ps aux    # Procesos en ejecución

# Windows
netstat -ano    # Puertos y conexiones
tasklist    # Procesos activos
```

Buenas Prácticas de Seguridad

1. Documentar todos los hallazgos
2. Realizar capturas de pantalla de vulnerabilidades
3. Mantener un registro de todas las pruebas realizadas
4. Verificar permisos antes de cada prueba
5. Hacer copias de seguridad antes de modificar configuraciones
6. Seguir una metodología estructurada (OSSTMM, PTES, etc.)

Notas Importantes

- Siempre obtener autorización antes de realizar pruebas
- Documentar cada paso realizado
- Mantener un entorno de pruebas controlado
- Evitar modificaciones permanentes en sistemas críticos
- Reportar inmediatamente vulnerabilidades críticas
- Mantener la confidencialidad de los hallazgos

Documentación Técnica

1. Especificaciones técnicas

1.1. Arquitectura del Sistema

Sistema de Auditoría
- Módulo de Reconocimiento (Nmap)
- Módulo de Análisis de Vulnerabilidades (Nessus)
- Módulo de Penetración (Metasploit)
- Módulo de Análisis de Tráfico (Wireshark)

1.2. Requisitos del Sistema

Requisitos mínimos de Hardware

- Procesador: Intel Core i5 o AMD equivalente
- RAM: 8GB
- Almacenamiento: 50GB SSD
- NIC: Tarjeta de red compatible con modo promiscuo

Requisitos recomendados de Hardware

- Procesador: Intel Core i7 o AMD Ryzen 7}
- RAM: 16GB
- Almacenamiento: 100GB SSD
- NIC: Intel PRO/1000 PT o similar

Software

- SO: Kali Linux 2023.1 o superior
- Kernel: 5.15 o superior
- Python: 3.9+
- PostgreSQL: 13+
- Java: OpenJDK 11+

2. Especificaciones de los Componentes

2.1. Módulo de Reconocimiento (Nmap)

Características Técnicas

yaml
Versión: 7.93+
Protocolos: TCP, UDP, SCTP
Métodos de Escaneo:
- SYN scan (-sS)
- TCP connect scan (-sT)
- UDP scan (-sU)
- Version detection (-sV)

Configuración de Red

bash

Configuración de interfaz

MTU: 1500

Modo: Promiscuo

Buffer de captura: 65535

2.2. Módulo de Análisis de Vulnerabilidades (Nessus)

Especificaciones

yaml

Versión: Nessus Professional 10+

Base de datos: SQLite

Puertos: 8834 (HTTPS)

API: REST v2.0

Estructura de Plugins

Plugins

- Configuración (.nasl)
- Credenciales (.inc)
- Políticas (.policy)

2.3. Módulo de Penetración (Metasploit)

Arquitectura

Framework

- Rex (Core)
- MSF::Core
- MSF::Base
- Interfaces

Base de datos

sql

-- Estructura PostgreSQL

CREATE DATABASE msf;

CREATE USER msf_user WITH PASSWORD 'password';

GRANT ALL PRIVILEGES ON DATABASE msf TO msf_user;

2.4. Módulo de Análisis de Tráfico (Wireshark)

Especificaciones Técnicas

yaml

Versión: 4.0+

Librerías: libpcap/WinPcap

Formatos: pcap, pcapng

Filtros: BPF, display filters

3. **Protocolos y Comunicación**
    3.1. **Protocolos de Red**

    yaml
    Protocolos Soportados:
      - TCP/IP (v4/v6)
      - UDP
      - ICMP
      - ARP
      - DNS
      - HTTP/HTTPS
      - SMB
      - SSH

    3.2. **Puertos y Servicios**

    yaml
    Puertos Default:
      - 22: SSH
      - 80/443: HTTP/HTTPS
      - 445: SMB
      - 3306: MySQL
      - 8834: Nessus

4. **Seguridad y Autenticación**
    4.1. **Mecanismos de Seguridad**

    yaml
    Encriptación:
      - TLS 1.3
      - AES-256-GCM
      - RSA 4096
    Autenticación:
      - OAuth 2.0
      - Certificados X.509
      - HMAC

    4.2. **Políticas de Seguridad**

    bash
    # Configuración de Firewall
    iptables -A INPUT -p tcp --dport 8834 -j ACCEPT
    iptables -A INPUT -p tcp --dport 22 -j ACCEPT

5. APIs e Interfaces

5.1. API de Nessus

```
json
{
  "endpoint": "/scans",
  "method": "POST",
  "headers": {
    "X-ApiKeys": "accessKey=xxx;secretKey=xxx",
    "Content-Type": "application/json"
  }
}
```

5.2. API de Metasploit

```
ruby
require 'msfrpc-client'
client = Msf::RPC::Client.new(
  host: '127.0.0.1',
  port: 55553,
  user: 'msf',
  pass: 'password'
)
```

6. Gestión de Datos

6.1. Estructura de Datos

```
sql
-- Esquema de base de datos
CREATE TABLE scans (
  id SERIAL PRIMARY KEY,
  target_ip VARCHAR (15),
  scan_type VARCHAR (50),
  start_time TIMESTAMP,
  end_time TIMESTAMP,
  status VARCHAR (20)
);

CREATE TABLE vulnerabilities (
  id SERIAL PRIMARY KEY,
  scan_id INTEGER REFERENCES scans(id),
  severity INTEGER,
  description TEXT,
  solution TEXT
);
```

6.2. Formato de Informes

```
xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<report>
  <scan_info>
    <target>192.168.1.0/24</target>
    <timestamp>2024-03-16T10:00:00Z</timestamp>
  </scan_info>
  <findings>
    <vulnerability>
      <severity>High</severity>
      <description>SQL Injection</description>
      <affected_host>192.168.1.100</affected_host>
    </vulnerability>
  </findings>
</report>
```

7. Rendimiento y Optimización

7.1. Configuraciones de Rendimiento

```
yaml
Nmap:
  max_parallelism: 100
  timing_template: T4
  host_timeout: 30m

Nessus:
  max_hosts: 10
  max_checks: 5
  timeout: 3600

Metasploit:
  thread_count: 10
  connect_timeout: 30
  read_timeout: 30
```

7.2. Monitoreo del Sistema

```
bash
# Métricas de rendimiento
CPU_THRESHOLD=80
MEM_THRESHOLD=90
DISK_THRESHOLD=85
```

8. Recuperación y Respaldo

8.1. Procedimientos de Backup

bash

```
#!/bin/bash
# Backup de configuraciones y datos
backup_dir="/backup/$(date +%Y%m%d)"
mkdir -p $backup_dir

# Backup de bases de datos
pg_dump msf > $backup_dir/msf.sql
sqlite3 nessus.db .dump > $backup_dir/nessus.sql

# Backup de configuraciones
cp -r /etc/nessus $backup_dir/
cp -r ~/.msf4 $backup_dir/
```

8.2. Plan de Recuperación

yaml

```
Procedimientos:
  1. Restauración de BD
     - psql msf < backup/msf.sql
  2. Restauración de configs:
     - cp – r  ~ /.msf4 $backup_dir/
  3. Reinicio de servicios
     - systemctl restart nessusd
     - systemctl restart postgresql
```

9. Mantenimiento

9.1. Actualizaciones

bash

```
# Actualización de componentes
apt update
apt upgrade nmap wireshark metasploit-framework
/opt/nessus/sbin/nessuscli update
```

9.2. Limpieza

sql

```
-- Limpieza de datos antiguos
DELETE FROM scans WHERE start_time < NOW() - INTERVAL '90
days';
DELETE FROM vulnerabilities WHERE scan_id NOT IN (SELECT id
FROM scans);
```