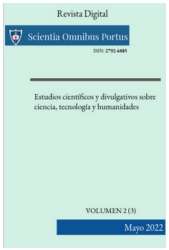
Scientia Omnibus Portus
ISSN: 2792-6885

https://iescelia.org/ojs



Revisión

Técnicas de enumeración de SMB

Esteban Sánchez, Pablo – IES Celia Viñas
Blanes Castro, Laura María – IES Al-Andalus
Correo para correspondencia: pabloesteban@iescelia.org

Resumen:

SMB es un protocolo para compartir recursos ampliamente extendido en redes locales y corporativas, presente de manera nativa en Windows y a través de Samba en otros sistemas. Es precisamente a consecuencia de esta presencia y de famosas vulnerabilidades descubiertas (aunque parcheadas), que se ha convertido en un objetivo prioritario de múltiples ataques informáticos. Para realizar un pentesting de SMB adecuado, se hace necesario disponer de toda la información posible. Es por ello que la **enumeración de SMB** es un habilidad fundamental para todo pentester. En este artículo, se hará una recopilación de algunas herramientas Open Source que permitirán al auditor recopilar información precisa. Es importante tener en cuenta que las herramientas aquí expuestas están pensadas para una auditoría de caja negra, en la que sólo tenemos acceso de red a las máquinas objetivo, pero no hemos accedido aún a ellas para una enumeración post-explotación.

Palabras clave: SMB, Samba, Ciberseguridad, Hacking Ético, Linux

Recibido el 11/04/2022
Aceptado el 15/04/2022

1. Introducción

SMB es el acrónimo de **Server Message Block**. Se trata de un protocolo (de Leeuw, 2006) para compartir recursos como archivos, impresoras y en general, cualquier recursos que deba ser usado o ser disponible por un servidor. En Windows está de manera nativa sin necesidad de instalar nada. Sin embargo, en Linux es un poco diferente. Para hacerlo funcionar necesitas instalar un servidor samba.
Antes de comenzar analizando herramientas, demos un repaso rápido a algunos conceptos.

1.1. Shares

En el ámbito de SMB, se llaman **shares** a los recursos expuestos por un servidor. Por defecto, Windows crea los siguientes shares:

- **IPC$**: Interprocess communication, para la comunicación entre procesos
- **ADMIN$**: para administrar remotamente el directorio %SystemRoot% (directorio de instalación de windows.
- **C$**: Unidad por defecto de un sistema Windows. Además, si el sistema objetivo tiene otras unidades con letra asignada (D:, E:, …) también aparecerán publicadas automáticamente.
- Si usas una impresora compartida, también aparecerá el recurso **print$**.

En **Windows**, podemos ver estos shares con el comando **net share**:

```
C:\Users\Administrador>net share

Nombre       Recurso                         Descripción

-------------------------------------------------------------------------------
C$           C:\                             Recurso predeterminado
IPC$                                         IPC remota
ADMIN$       C:\Windows                      Admin remota
Se ha completado el comando correctamente.
```

1.2. Recurso IPC$

Los recursos C$, D$, … asociados a directorios, son más fáciles de entender, pero no tanto el IPC$. El share o recurso compartido **IPC$ -** Inter process communication - (Han Liang, 2021) es un recurso especial para permitir comunicaciones entre procesos mediante un mecanismo llamado **named pipes** o **canalizaciones por nombre**.

Named Pipes
Técnica para permitir la comunicación entre 2 procesos. Consiste en el uso de un recurso compartido (una región de memoria tratada como un fichero) entre ambos, donde un proceso escribe los datos a compartir, y otro los lee. (Hawes, 2019)

1.3. Sesión Nula (Null Session)

En SMB, se conoce como **conexión de sesión nula** a la establecida para acceder a un recurso de red (normalmente IPC$, aunque también es válido para conectarse a un share convencional) sin suministrar ningún tipo de credenciales. Esto también se conoce como **acceso anónimo** o de **invitado**. Al usar esta sesión, Windows permite (dependiendo de la versión) realizar determinadas actividades, como enumerar los nombres de las cuentas de dominio y los recursos compartidos de red.

Una **sesión remota normal** ocurre cuando un usuario se loguea a una computadora remota usando las credenciales adecuadas (usuario/contraseña u otro mecanismo de autenticación aceptado, como relaciones de confianza en un dominio) para tener acceso a los recursos del sistema. Esto se lleva a cabo a través del protocolo SMB.

Una **sesión nula** entra en juego cuando un usuario hace una conexión a un sistema Windows sin credenciales (o mejor dicho, con usuario '' u contraseña '').

¿Cuál es el propósito de estas sesiones nulas?

- Que los dominios de confianza enumeren recursos
- Que los equipos no pertenecientes al dominio puedan autenticarse y enumerar usuarios.

Todo lo que se puede hacer con las sesiones nulas se ha ido recortando en las sucesivas versiones de **SMB** , de modo que, actualmente, es poca información la que puede recuperarse.

Las sesiones nulas estaban **habilitadas por defecto en NT y en 2000**, permitiendo a una persona cualquiera con acceso a la red listar usuarios, grupos, recursos compartidos, etc, … y todo sin suministrar credenciales. Hay que tener en cuenta que quedan aún muchas instalaciones de este tipo.

Posteriormente, en **XP** y **2003** se continuó permitiendo por defecto el establecimiento de sesiones nulas, pero se limitó la enumeración a las carpetas compartidas, salvaguardando información de usuarios y grupos.

Es ya, **a partir de Windows Vista y 2008**, que se endurecen las configuraciones por defecto, y es poco lo que se puede recuperar en estas versiones y superiores con una versión nula.

En definitiva, la sesión nula **no** permitiría el acceso sin restricciones a la máquina, pero **permitirá, en caso de versiones de Windows más antiguas o incorrectamente configuradas, una enumeración bastante extensa** que podría ayudar a un atacante y darle pistas suficientes para fases posteriores.

1.4. Puertos y servicios

Normalmente, hablamos de la presencia de SMB en una máquina si detectamos abiertos puertos como el **139 TCP** o el **445 TCP**. Sin embargo, esto puede resultar un poco confuso, ya que ¿Por qué existen 2?, ¿Por qué en ocasiones sólo está el 139 o el 445? Vamos a intentar aclararlo.

Simplificando mucho, y sin ánimo de ser exhaustivos, hoy día podemos encontrarnos 2 implementaciones de SMB (Ozturk, 2022):

1. **SMB sobre NetBIOS**: NetBIOS es un conjunto de protocolos de la capa se sesión OSI que proporcionan distintas utilidades para la comunicación de equipos en una LAN. Entre estos servicios encontramos el **Session Service** (**TCP 139**), el **Name Service** (**UDP 137**), … Para NetBIOS caben citar 2 implementaciones dependiendo de los protocolos subyacentes:
   a) **NetBIOS sobre TCP/IP**, también llamado **NBT** (NetBIOS over TCP/IP)
   b) **NetBIOS sobre NetBEUI**, obsoleto y que escapa del ámbito de este capítulo.
2. **SMB sobre TCP/IP** directamente: usa el puerto **445 TCP**. A partir de Windows 2000, Microsoft lanza esta versión de SMB, que no se apoya en NetBIOS, sino que usa directamente la pila TCP/IP.

Veamos cómo se organizan sobre el modelo OSI:

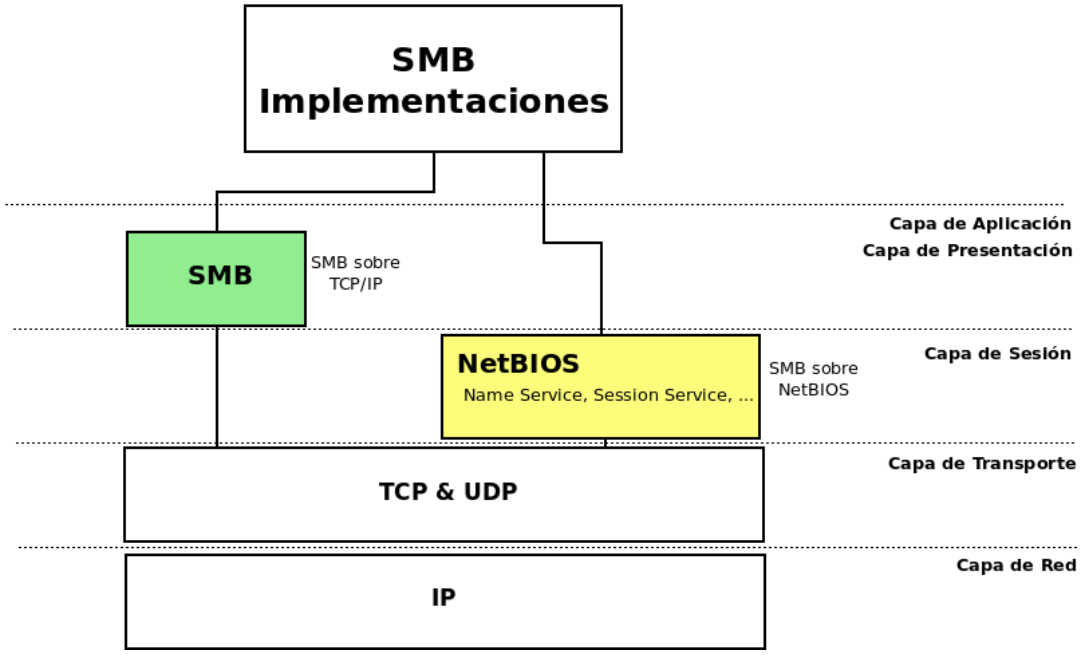


Figura 1: Principales implementaciones de SMB

Respecto a los puertos usados, y a modo de resumen:

- **SMB sobre TCP/IP**: Usa el puerto **TCP 445**.
- **SMB sobre NetBIOS sobre TCP/IP**: También conocido como NBT. Aquí tenemos una serie de protocolos de la capa de sesión, que usan distintos puertos. Los más comunes son:
  - **Session Service**: También conocido como NetBIOS-SSN. Escucha en el **TCP 139** y se encarga de crear una sesión entre 2 máquinas. Dicha sesión permite cosas como detección y corrección de errores, partición y reensamblaje de mensajes, …
  - **Name Service**: También conocido como NetBIOS-NS. Escucha en el **UDP 137**. Se trata de un servicio de nombres en el que se ha de registrar el nombre de las máquinas para poder hacer uso del resto de servicios NetBIOS.
  -

Importante! La enumeración que vamos a hacer aquí es previa a la fase de explotación, es decir, intentaremos sacar toda la información posible **sin haber comprometido el host destino**.

Una vez que se haya realizado la explotación, la cantidad de información a enumerar será significativamente mayor.

2. Enumeración desde Linux

En este apartado, nos vamos a centrar en la enumeración de SMB/Samba usando herramientas de Linux. Aunque se quedan fuera algunas interesantes, trataremos de hacer un repaso de las principales.

NOTA: Error común al trabajar con samba en Linux

Cannot connect to server. Error was NT_STATUS_CONNECTION_DISCONNECTED
Este error ocurre porque tienes herramientas modernas que son incapaces de comunicarse con protocolos inseguros y antiguos. Necesitas indicarle a tu demonio smbd que pruebe también versiones de SMB más antiguas.
1. Abre **/etc/samba/smb.conf**
2. En la sección **Global**, añade:
client min protocol = NT1
3. reinicia el servicio smbd
systemctl restart smdb
3bis. También puedes cambiar la opción al vuelo:
$ rpcclient -U "" -N 10.0.1.200 **--option "client min protocol = NT1"**

Las pruebas las haremos principalmente con la máquina vulnerable metasploitable2 de rapid7: https://docs.rapid7.com/metasploit/metasploitable-2/, que se trata de un Linux con Samba, y algunas con metasploitable3: https://github.com/rapid7/metasploitable3, que se trata de un Windows Server 2008 R2. También haremos alguna prueba puntual con Windows 10 y con Windows 2000.

Para ello, se ha montado un laboratorio en Virtual Box consistente en una red nat con la dirección de red **10.0.1.0/24** donde tenemos además, nuestra máquina atacante.

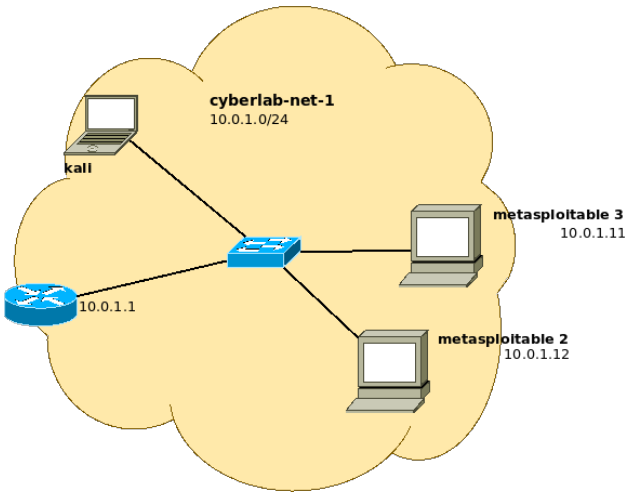


Figura 2: Esquema del laboratorio

2.1. Enum4Linux – Utilidad todo en uno

En una herramienta completa de enumeración escrita en perl. Se trata de una herramienta de 'botón gordo' que, si es posible, nos proporciona una gran cantidad de información de enumeración sobre el objetivo. Por eso, la vemos la primera. Básicamente es un wrapper de herramientas que veremos más tarde como smbclient, rpcclient, nmblookup, …

Muy recomendable consultar su ayuda:

```
$ enum4linux –h
```

Lo más común es lanzar el siguiente comando, que hace una enumeración completa:

```
$ enum4linux –a IP_OBJETIVO
```

Ese **-a** agrupa un conjunto de opciones de enum4linux, **-U -S -G -P -r -o -n -i** , siendo cada una de ellas:

- **-U** → obtener lista de usuarios
- **-S** → obtener lista de shares
- **-G** → obtener grupos y sus miembros
- **-P** → obtener política de contraseñas del sistema objetivo
- **-r** → enumerar usuarios recorriendo RIDs secuencialmente
- **-o** → Obtener información del sistema operativo
- **-n** → hace un nmblookup (similar a nbstat)
- **-i** → Obtiene información sobre impresoras

Enum4linux tratará de acceder mediante sesión nula o anónimamente. Un sistema bien protegido no permitirá este tipo de accesos, o la cantidad de información recopilada será muy escasa. Sin embargo, es oportuno realizarlo siempre que estemos ante un sistema SMB/Samba.

Probemos en primer lugar contra **metasploitable 2**, un sistema linux que usa samba:

```
# enum4linux 10.0.1.12
```

A continuación, analizaremos por partes la salida devuelta por el comando.

Información básica del objetivo, y cómo se ha accedido (con sesión nula, con cuenta guest, ...)

```
 ============================
|    Target Information    |
 ============================
Target ........... 10.0.1.12
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none



 ================================================
|    Enumerating Workgroup/Domain on 10.0.1.12    |
 ================================================
[+] Got domain/workgroup name: WORKGROUP
```

Resultado de **nmblookup**:

```
 ========================================
|    Nbtstat Information for 10.0.1.12    |
 ========================================
Looking up status of 10.0.1.12
        METASPLOITABLE  <00> -         B <ACTIVE>  Workstation Service
        METASPLOITABLE  <03> -         B <ACTIVE>  Messenger Service
        METASPLOITABLE  <20> -         B <ACTIVE>  File Server Service
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>  Master Browser
        WORKGROUP       <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
        WORKGROUP       <1d> -         B <ACTIVE>  Master Browser
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>  Browser Service Elections

        MAC Address = 00-00-00-00-00-00
```

Básicamente, ejecuta un **nbtstat** sobre una máquina. Útil para enumerar dominios/estaciones de trabajo y MACS. NetBIOS funciona con la ayuda de sufijos que describen las características o capacidades del objetivo, y que significan lo siguiente:

Para nombres únicos:

- **00**: Workstation Service (workstation name)
- **03**: Windows Messenger service
- **06**: Remote Access Service
- **20**: File Service (also called Host Record)
- **21**: Remote Access Service client
- **1B**: Domain Master Browser – Primary Domain Controller for a domain
- **1D**: Master Browser

Para nombres de grupo:

- **00**: Workstation Service (workgroup/domain name)
- **1C**: Domain Controllers for a domain
- **1E**: Browser Service Elections

En el caso que acabamos de ver de metasploitable2, podemos decir, en base a los sufijos devueltos, lo siguiente:

- Su nombre NetBIOS es METASPLOITABLE (00)
- Tiene habilitado el 'Windows Messenger Service' (03), que permitía mandar mensajes entre estaciones a través de una LAN. Funcionaba a través del comando net send …
- Permite el uso compartido de archivos (20)

Pertenece al grupo 'WORKGROUP'

Obtención de **información del sistema operativo objetivo**

```
 ====================================
|    OS information on 10.0.1.12    |
 ====================================
Use of uninitialized value $os_info in concatenation (.) or string at ./enum4linux.pl line 464.
[+] Got OS info for 10.0.1.12 from smbclient:
[+] Got OS info for 10.0.1.12 from srvinfo:
```

```
        METASPLOITABLE Wk Sv PrQ Unx NT SNT metasploitable server (Samba 3.0.20-Debian)
        platform_id     :       500
        os version      :       4.9
        server type     :       0x9a03
```

Obtención de usuarios (resumimos la lista obtenida):

```
 ============================
|    Users on 10.0.1.12     |
 ============================
index: 0x1 RID: 0x3f2 acb: 0x00000011 Account: games    Name: games     Desc: (null)
index: 0x2 RID: 0x1f5 acb: 0x00000011 Account: nobody   Name: nobody    Desc: (null)
index: 0x3 RID: 0x4ba acb: 0x00000011 Account: bind     Name: (null)    Desc: (null)
index: 0x4 RID: 0x402 acb: 0x00000011 Account: proxy    Name: proxy     Desc: (null)
index: 0x5 RID: 0x4b4 acb: 0x00000011 Account: syslog   Name: (null)    Desc: (null)
index: 0x6 RID: 0xbba acb: 0x00000010 Account: user     Name: just a user,111,, Desc: (null)
index: 0x7 RID: 0x42a acb: 0x00000011 Account: www-data Name: www-data  Desc: (null)
index: 0x8 RID: 0x3e8 acb: 0x00000011 Account: root     Name: root      Desc: (null)
...
index: 0x18 RID: 0xbb8 acb: 0x00000010 Account: msfadmin Name: msfadmin,,,       Desc: (null)
index: 0x19 RID: 0x4c8 acb: 0x00000011 Account: telnetd Name: (null)    Desc: (null)
...

user:[games] rid:[0x3f2]
user:[nobody] rid:[0x1f5]
user:[bind] rid:[0x4ba]
user:[proxy] rid:[0x402]
user:[syslog] rid:[0x4b4]
user:[user] rid:[0xbba]
user:[www-data] rid:[0x42a]
user:[root] rid:[0x3e8]
...
user:[msfadmin] rid:[0xbb8]
user:[telnetd] rid:[0x4c8]
...
```

Como ves, entre otros tenemos root y msfadmin, cuyas contraseñas podremos obtener más adelante por fuerza bruta.

Listado de **shares** expuestos:

```
 =====================================
|    Share Enumeration on 10.0.1.12    |
 =====================================

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
```

```
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            METASPLOITABLE

[+] Attempting to map shares on 10.0.1.12
//10.0.1.12/print$      Mapping: DENIED, Listing: N/A
//10.0.1.12/tmp Mapping: OK, Listing: OK
//10.0.1.12/opt Mapping: DENIED, Listing: N/A
//10.0.1.12/IPC$        [E] Can't understand response:
NT_STATUS_NETWORK_ACCESS_DENIED listing \*
//10.0.1.12/ADMIN$      Mapping: DENIED, Listing: N/A
```

Política de contraseñas del objetivo:

```
 ================================================
|    Password Policy Information for 10.0.1.12    |
 ================================================
...
[+] Password Info for Domain: METASPLOITABLE

        [+] Minimum password length: 5
        [+] Password history length: None
        [+] Maximum password age: Not Set
        [+] Password Complexity Flags: 000000

                [+] Domain Refuse Password Change: 0
                [+] Domain Password Store Cleartext: 0
                [+] Domain Password Lockout Admins: 0
                [+] Domain Password No Clear Change: 0
                [+] Domain Password No Anon Change: 0
                [+] Domain Password Complex: 0

        [+] Minimum password age: None
        [+] Reset Account Lockout Counter: 30 minutes
        [+] Locked Account Duration: 30 minutes
```

```
        [+] Account Lockout Threshold: None
        [+] Forced Log off Time: Not Set


[+] Retieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 0
```

Grupos encontrados. No se ha encontrado nada en este caso.

```
 ============================
|    Groups on 10.0.1.12     |
 ============================

[+] Getting builtin groups:

[+] Getting builtin group memberships:

[+] Getting local groups:

[+] Getting local group memberships:

[+] Getting domain groups:

[+] Getting domain group memberships:
```

Probamos a obtener más usuarios iterando sobre su **RID**. Básicamente, se encuentra un **SID** (Security Identifier), que identifica de manera única objetos como cuentas de dominio, cuentas locales o grupos. Este security identifier tiene una parte común para todos loso objetos del mismo sistema, y una parte que cambia, el **RID** (Relative ID).

En nuestro caso, un ejemplo de SID sería:

```
S-1-5-21-1042354039-2475377354-766472396-500
```

Siendo la parte común a todos los objetos:

```
S-1-5-21-1042354039-2475377354-766472396
```

Y el RID:

```
500
```

500 suele ser el RID del administrador built-in en todas las instalaciones. La técnica consiste en ir iterando de 1 en 1 a partir de 500: 501, 502, …, e ir obteniendo SIDs válidos de cuentas, grupos, ...

Resumimos la salida:

```
 ============================================================
|    Users on 10.0.1.12 via RID cycling (RIDS: 500-550,1000-1050)    |
 ============================================================
[I] Found new SID: S-1-5-21-1042354039-2475377354-766472396
[+] Enumerating users using SID S-1-5-21-1042354039-2475377354-766472396 and logon username '', password ''
S-1-5-21-1042354039-2475377354-766472396-500 METASPLOITABLE\Administrator (Local User)
S-1-5-21-1042354039-2475377354-766472396-501 METASPLOITABLE\nobody (Local User)
S-1-5-21-1042354039-2475377354-766472396-502 *unknown*\*unknown* (8)
...
S-1-5-21-1042354039-2475377354-766472396-512 METASPLOITABLE\Domain Admins (Domain Group)
S-1-5-21-1042354039-2475377354-766472396-513 METASPLOITABLE\Domain Users (Domain Group)
S-1-5-21-1042354039-2475377354-766472396-514 METASPLOITABLE\Domain Guests (Domain Group)
S-1-5-21-1042354039-2475377354-766472396-515 *unknown*\*unknown* (8)
...
S-1-5-21-1042354039-2475377354-766472396-1000 METASPLOITABLE\root (Local User)
S-1-5-21-1042354039-2475377354-766472396-1001 METASPLOITABLE\root (Domain Group)
…
```

Obtenemos información de **posibles impresoras compartidas**:

```
 ========================================
|    Getting printer info for 10.0.1.12    |
 ========================================
No printers returned.
```

Recuerda lo que comentamos anteriormente. Enum4linux intenta una **conexión anónima o sesión nula**. Dependiendo de la versión del sistema operativo objetivo (en el caso de windows) o de samba, la información obtenida será mayor o menor.

En sistemas bien securizados y actualizados, la información proporcionada está restringida. Sólo obtendríamos cosas como el grupo de trabajo, el nbtstat, y poco más.

Para una información más completa, debemos probar con un usuario del sistema. Para ello, enum4linux pone a nuestra disposición las opciones **-u** (para especificar un nombre de usuario) y **-p** (para especificar una contraseña).

```
$ enum4linux -a -u vagrant -p vagrant 10.0.1.11
```

La cantidad de información también dependerá de si ese usuario tiene privilegios de administración o no (la diferencia entre enumerar el listado de shares o proporcionar detalles de cada share).

2.2. NMAP

Nmap es una herramienta muy versátil que debe formar parte del inventario básico de cualquier pentester. En este apartado, haremos un repaso de cómo podemos usarla para enumerar samba/smb.

2.2.1. Enumeración rápida con scripts por defecto -sC

Esta debería ser nuestra primera opción en una enumeración. El parámetro -sC, lanza scripts de la categoría *default*. Scripts rápidos y considerados seguros.

```
$ sudo nmap -sC -p 139,445 -sV 10.0.1.11 10.0.1.12
```

Por ser más exhaustivos, incluiremos en un apartado posterior una lista detallada de scripts que pueden sernos de utilidad
Analicemos su salida:

Metasploitable 2 (10.0.1.12):

```
PORT    STATE SERVICE     VERSION
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
MAC Address: 08:00:27:56:C9:96 (Oracle VirtualBox virtual NIC)


Host script results:
|_clock-skew: mean: 2h30m00s, deviation: 3h32m08s, median: 0s
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2021-02-12T01:20:41-05:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)
```

Conclusiones:

- Metasploitable2 usa samba.
- Metasploitable2 usa samba 3.0.20-Debian. (Seguramente tenga vulnerabilidades pero hay que ver si está parcheada)
- La negociación de SMBv2 (*smb2-time*) ha fallado, de lo que podríamos deducir que el dialecto de SMB usado es SMBv1.

2.2.2. Usando scripts de nmap específicos

Los siguiente scripts no son tan seguros como los de la categoría *default*. Hay que lanzarlos con cautela.

2.2.2.1. smb-protocols

Trata de obtener la lista de versiones de SMB soportados por el objetivo.
Las versiones 1 y 2 son vulnerables y esto puede proporcionarnos información interesante.

Por ejemplo, lanzado contra metasploitable 2:

```
# nmap --script smb-protocols -sV -p139,445 10.0.1.12
```

Nos devuelve:

```
PORT STATE SERVICE
139/tcp open netbios-ssn
445/tcp open microsoft-ds
MAC Address: 08:00:27:56:C9:96 (Oracle VirtualBox virtual NIC)
```

```
Host script results:
| smb-protocols:
| dialects:
|_ NT LM 0.12 (SMBv1) [dangerous, but default]
```

Lo que confirma lo que dedujimos antes con el script por defecto **smv2-time**.

2.2.2.2. smb-os-discovery

Se usa para enumerar el sistema operativo del objetivo, permitiendo además obtener otra información relevante como: el nombre de la máquina, el nombre del dominio o el grupo de trabajo, el nombre **NETBIOS**, …

Este script devolverá información valiosa si **SMBv1** está activado. SMBv1 permite obtener información sin autenticación. Esta característica ha sido explotada durante muchos años ya que SMBv1 sigue activado en sistemas operativos modernos por motivos compatibilidad, incluso aunque la última versión de Windows que puede sólo negociar SMBv1 fue Windows Server 2003.

Veámoslo contra metasploitable 2:

```
# nmap --script smb-os-discovery -p139,445 -sV 10.0.1.12
```

Siendo parte de la salida:

```
PORT    STATE SERVICE     VERSION
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
MAC Address: 08:00:27:56:C9:96 (Oracle VirtualBox virtual NIC)

Host script results:
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
```

Si lo probamos con un Windows 10, vemos que no es posible obtener información:

```
NSE: Starting smb-os-discovery against 10.0.1.38.
NSE: [smb-os-discovery 10.0.1.38] SMB: Added account '' to account list
NSE: [smb-os-discovery 10.0.1.38] SMB: Added account 'guest' to account list
NSE: [smb-os-discovery 10.0.1.38] Couldn't negotiate a SMBv1 connection:SMB: Failed to
receive bytes: ERROR
NSE: Finished smb-os-discovery against 10.0.1.38.
```

2.2.2.3. smb-enum-shares

Enumera los shares de SMB expuestos por el objetivo. Además, si no es posible obtenerlos, intentará obtenerlos por fuerza bruta basada en un diccionario de nombres de shares comunes.

Probémoslo contra **metasploitable 2**:

```
# nmap --script smb-enum-shares -p139,445 -sV 10.0.1.12
```

Obteniendo:

```
PORT    STATE SERVICE     VERSION
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 08:00:27:56:C9:96 (Oracle VirtualBox virtual NIC)

Host script results:
| smb-enum-shares:
|   account_used: <blank>
|   \\10.0.1.12\ADMIN$:
|     Type: STYPE_IPC
|     Comment: IPC Service (metasploitable server (Samba 3.0.20-Debian))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: <none>
|   \\10.0.1.12\IPC$:
|     Type: STYPE_IPC
|     Comment: IPC Service (metasploitable server (Samba 3.0.20-Debian))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|   \\10.0.1.12\opt:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: <none>
|   \\10.0.1.12\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
```

```
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|   \\10.0.1.12\tmp:
|     Type: STYPE_DISKTREE
|     Comment: oh noes!
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|_    Anonymous access: READ/WRITE
```

El script funciona anónimamente hasta Windows 2000, y a partir de éste, requiere una cuenta con **privilegios de usuario** para listar los shares, y de **administrador** para obtener más detalle sobre ellos, sino, se recurrirá a obtenerlos por fuerza bruta.

A continuación, probamos con un Windows 2008, donde la lista de shares se obtiene por fuerza bruta. Necesitamos una cuenta con permisos administrativos para obtener más detalles sobre los mismos. Probando anónimamente, obtenemos:

```
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE:
cpe:/o:microsoft:windows

Host script results:
| smb-enum-shares:
| note: ERROR: Enumerating shares failed, guessing at common ones
(NT_STATUS_ACCESS_DENIED)
| account_used: <blank>
| \\10.0.1.11\ADMIN$:
| warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
| Anonymous access: <none>
| \\10.0.1.11\C$:
| warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
| Anonymous access: <none>
| \\10.0.1.11\IPC$:
| warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
| Anonymous access: READ
| \\10.0.1.11\USERS:
| warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|_ Anonymous access: <none>
```

Sin embargo, podemos suministrar los parámetros **smbusername** y **smbpassword** para ejecutarlo con credenciales:

```
...
Host script results:
| smb-enum-shares:
|   account_used: vagrant
|   \\10.0.1.11\ADMIN$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Remote Admin
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\Windows
```

```
|     Anonymous access: <none>
|     Current user access: READ/WRITE
|   \\10.0.1.11\C$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Default share
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\
|     Anonymous access: <none>
|     Current user access: READ/WRITE
|   \\10.0.1.11\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: Remote IPC
|     Users: 1
|     Max Users: <unlimited>
|     Path:
|     Anonymous access: READ
|     Current user access: READ/WRITE
...
```

2.2.2.4. smb-enum-users

Trata de enumerar todos los usuarios del sistema objetivo usando 2 técnicas: la Enumeración SAMR, y fuerza bruta LSA.

Si el objetivo permite la enumeración de usuarios anónima, el listado obtenido se incluirá entre los resultados. Ten en cuenta que en los sistemas posteriores a Windows 2000 es necesario proporcionar credenciales válidas, ya que, incluso habiendo sesión nula, la lista de usuarios está restringida.
Probemos contra **metasploitable 2**:

```
# nmap --script smb-enum-users -p139,445 10.0.1.12
```

Obteniendo, entre otras cosas:

```
Host script results:
| smb-enum-users:
|   METASPLOITABLE\backup (RID: 1068)
|     Full name:   backup
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\bin (RID: 1004)
|     Full name:   bin
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\bind (RID: 1210)
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\daemon (RID: 1002)
|     Full name:   daemon
```

```
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\dhcp (RID: 1202)
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\distccd (RID: 1222)
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\ftp (RID: 1214)
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\games (RID: 1010)
…
```

De nuevo, en un Windows 2008 sin proporcionar credenciales:

```
...
Host script results:
| smb-enum-users:
|_ ERROR: Access denied while trying to enumerate users; except against Windows 2000,
Guest or better is typically required
...
```

Mientras que si lo ejecutamos para este mismo sistema usando una cuenta de administrador:

```
# nmap --script smb-enum-users -p139,445 -sV --script-args
smbusername=vagrant,smbpassword=vagrant 10.0.1.11
```

Obtenemos:

```
...
Host script results:
| smb-enum-users:
|   METASPLOITABLE3\Administrator (RID: 500)
|     Description: Built-in account for administering the computer/domain
|     Flags:       Normal user account
|   METASPLOITABLE3\anakin_skywalker (RID: 1011)
|     Flags:       Normal user account
|   METASPLOITABLE3\artoo_detoo (RID: 1007)
|     Flags:       Normal user account
|   METASPLOITABLE3\ben_kenobi (RID: 1009)
|     Flags:       Normal user account
…
```

2.2.2.5. Vulnerabilidades en SMB

Además, existen scripts que buscan vulnerabilidades conocidas. Estos serían, entre otros:

- smb-vuln-conficker
- smb-vuln-cve-2017-7494
- smb-vuln-cve2009-3103
- smb-vuln-ms06-025
- smb-vuln-ms07-029
- smb-vuln-ms08-067
- smb-vuln-ms10-054
- smb-vuln-ms10-061
- smb-vuln-ms17-010
- smb-vuln-regsvc-dos
- smb-vuln-webexec

Por ejemplo, el script **smb-vuln-ms08-067** comprueba si el objetivo es vulnerable a la vulnerabilidad de ejecución de código remoto **MS08-067**, también conocida como **netapi** o **CVE-2008-4250**, que afecta a Windows 2000, XP y Windows Server 2003.

Otro ejemplo destacable es el **smb-vuln-ms17-010**, que detecta la vulnerabilidad **EternalBlue**, **ms17-010**, o **CVE-2017-0143**.

Por ejemplo, si queremos comprobar si una máquina es vulnerable a **EternalBlue**, ejecutamos:

```
# nmap --script smb-vuln-ms17-010 -p 139,445 <target>
```

Si lo ejecutamos contra metasploitable3:

```
# nmap --script smb-vuln-ms17-010 -p139,445 -sV 10.0.1.11
…
Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
 https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
…
```

2.3. Herramientas específicas

2.3.1. nbtscan – busca de servidores de nombres NetBIOS

- Este comando escanea redes o hosts discretos, en busca de información de nombres NetBIOS abiertos en una red. Funcionalmente, es muy parecido a nbtstat de windows.
- Es un primer paso para encontrar cosas compartidas.
- Funciona contra IPs individuales o contra subredes.

```
# nbtscan 10.0.1.0/24
Doing NBT name scan for addresses from 10.0.1.0/24

IP address       NetBIOS Name     Server    User             MAC address
------------------------------------------------------------------------------
10.0.1.0        Sendto failed: Permission denied
10.0.1.11        METASPLOITABLE3  <server>  <unknown>        08:00:27:cd:bc:7e
10.0.1.12        METASPLOITABLE   <server>  METASPLOITABLE   00:00:00:00:00:00
10.0.1.255      Sendto failed: Permission denied
```

- La columna *user* se corresponde con el usuario actualmente logueado en el sistema (si no hay ninguno, te pone el nombre de la máquina)., Si usas **-v**, obtienes la tabla de nombres netbios completa para cada hosts (lo que hace nmblookup o nbtstat en Windows)

2.3.2. nmblookup - datos de NetBIOS

Usada para consultar **nombres NetBIOS** y mapearlas a direcciones IP en una red. Comentamos ya, cuando hablamos de enum4linux, como NetBIOS utiliza sufijos para caracterizar las capacidades de una estación de trabajo. Estos sufijos son unos números con distinto significado si se aplican a nivel individual o de grupo:

Para nombres únicos:

- **00**: Workstation Service (workstation name)
- **03**: Windows Messenger service
- **06**: Remote Access Service
- **20**: File Service (also called Host Record)
- **21**: Remote Access Service client
- **1B**: Domain Master Browser – Primary Domain Controller for a domain
- **1D**: Master Browser

Para nombres de grupo:

- **00**: Workstation Service (workgroup/domain name)
- **1C**: Domain Controllers for a domain
- **1E**: Browser Service Elections

Para metasploitable2, sería tal y como se describe a continuación, y sus resultados ya los comentamos anteriormente:

```
$ nmblookup -A 10.0.1.12
```

También es posible consultar, en lugar de un host, un grupo de trabajo o dominio. Sin -S, sólo te devuelve la IP del miembro del grupo:

```
$ nmblookup -S WORKGROUP
```

```
10.0.1.11 WORKGROUP<00>
Looking up status of 10.0.1.11
        METASPLOITABLE3 <00> -         B <ACTIVE>
        WORKGROUP       <00> - <GROUP> B <ACTIVE>
        METASPLOITABLE3 <20> -         B <ACTIVE>

        MAC Address = 08-00-27-CD-BC-7E

10.0.1.12 WORKGROUP<00>
Looking up status of 10.0.1.12
        METASPLOITABLE  <00> -         B <ACTIVE>
        METASPLOITABLE  <03> -         B <ACTIVE>
        METASPLOITABLE  <20> -         B <ACTIVE>
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>
        WORKGROUP       <00> - <GROUP> B <ACTIVE>
        WORKGROUP       <1d> -         B <ACTIVE>
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>

        MAC Address = 00-00-00-00-00-00
```

2.3.4. smbmap – enumerar recursos compartidos

Este comando te permite enumerar los shares de un host o de un grupo de trabajo o dominio.
Lista unidades compartidas, permisos sobre los mismos, y si se puede: contenidos.
Entre sus parámetros:

- **-H**: para indicar la IP del host
- **-P**: para indicar el puerto SMB (por defecto es 445)
- **-v**: nos proporciona información sobre la versión del SO.

Veamos su uso con sesión nula:

Metasploitable2 (10.0.1.12)

```
# smbmap -H 10.0.1.12 -R
[+] IP: 10.0.1.12:445   Name: 10.0.1.12
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        tmp                                                     READ, WRITE     oh noes!
        .\tmp\*
        dr--r--r--                0 Fri Feb 12 02:15:23 2021    .
        dw--w--w--                0 Thu Dec 24 05:43:29 2020    ..
        dr--r--r--                0 Fri Feb 12 01:49:38 2021    .ICE-unix
        fw--w--w--                0 Fri Feb 12 00:50:02 2021    4539.jsvc_up
        dr--r--r--                0 Fri Feb 12 00:49:45 2021    .X11-unix
        fw--w--w--               11 Fri Feb 12 00:49:45 2021    .X0-lock
        .\tmp\.X11-unix\*
```

```
        dr--r--r--                0 Fri Feb 12 00:49:45 2021    .
        dr--r--r--                0 Fri Feb 12 02:15:23 2021    ..
        fr--r--r--                0 Fri Feb 12 00:49:45 2021    X0
        opt                                                     NO ACCESS
        IPC$                                                      NO ACCESS       IPC Service (metasploitable
server (Samba 3.0.20-Debian))
        ADMIN$                                                    NO ACCESS       IPC Service (metasploitable
server (Samba 3.0.20-Debian))
```

Sin **-R** no explora directorios.

En caso de que no se permita sesión nula, o deseemos acceder con credenciales:

```
# smbmap -H 10.0.1.11 -u vagrant -p vagrant
```

Este comando ofrece infinitas posibilidades, incluso ejecutar comandos remotos con el parámetro -x. Aquí sólo hemos visto unas pinceladas de su uso.

2.3.5. smbclient – conexión con servidor SMB

smbclient es un cliente que puede comunicarse con un servidor SMB/CIFS y ofrece multitud de posibilidades relacionadas con el sistema de ficheros al que nos conectamos. Permite, entre otras bajar y subir ficheros del servidor, listar directorios remotos, …

Normalmente, se usa en 2 fases: Primero, con el parámetro -L para **listar los shares** (elementos compartidos) de una ip. Y luego, nos conectamos a un share concreto.

1. Para listar los shares:

```
# smbclient -L 10.0.1.12
Enter WORKGROUP\root's password:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
```

2. Nos conectamos a un share concreto:

```
# smbclient //10.0.1.12/tmp
Enter WORKGROUP\root's password:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Fri Feb 12 02:45:11 2021
```

```
  ..                                  DR        0  Thu Dec 24 05:43:30 2020
  .ICE-unix                           DH        0  Fri Feb 12 01:49:39 2021
  4539.jsvc_up                         R        0  Fri Feb 12 00:50:03 2021
  ...
```

A partir de aquí tienes bastantes posibilidades: puedes ver una lista completa de comandos ejecutando **help**:

```
smb: \> help
?              allinfo        altname        archive        backup
blocksize      cancel         case_sensitive cd             chmod
chown          close          del            deltree        dir
du             echo           exit           get            getfacl
geteas         hardlink       help           history        iosize
lcd            link           lock           lowercase      ls
l              mask           md             mget           mkdir
more           mput           newer          notify         open
posix          posix_encrypt  posix_open     posix_mkdir    posix_rmdir
posix_unlink   posix_whoami   print          prompt         put
pwd            q              queue          quit           readlink
rd             recurse        reget          rename         reput
rm             rmdir          showacls       setea          setmode
scopy          stat           symlink        tar            tarmode
timeout        translate      unlock         volume         vuid
wdel           logon          listconnect    showconnect    tcon
tdis           tid            utimes         logoff         ..
!
smb: \>
```

Por ejemplo, si deseamos subir un fichero de nuestra máquina local. Podemos subirlo a metasploitable 2 así:

```
smb: \> put prueba.txt
putting file prueba.txt as \prueba.txt (0.1 kb/s) (average 0.1 kb/s)
smb: \> ls
  .                                   D        0  Mon Feb 12 03:30:40 2021
  ..                                 DR        0  Thu Dec 24 05:43:30 2020
  .ICE-unix                          DH        0  Mon Feb 12 02:21:28 2021
  .X11-unix                          DH        0  Mon Feb 12 01:21:37 2021
  .X0-lock                           HR       11  Mon Feb 12 01:21:37 2021
  4526.jsvc_up                        R        0  Mon Feb 12 01:21:54 2021
  prueba.txt                          A        3  Mon Feb 12 03:30:40 2021
```

Como veremos más adelante, esto es un vector de ataque muy importante, ya que se abre una vía para la subida de payloads maliciosos, que puede ser explotado entre otras formas, con vulnerabilidades File Inclusion.

Si necesitamos acceder con credenciales a un share:

```
# smbclient //10.0.1.11/C$ -U=vagrant%vagrant
Try "help" to get a list of possible commands.
smb: \> dir
  $Recycle.Bin                        DHS        0  Mon Jul 13 22:34:39 2009
  Boot                                DHS        0  Sun Jul 19 06:07:30 2020
  bootmgr                            AHSR   383786  Sat Nov 20 22:24:02 2010
```

2.3.6. rpcclient – ejecución de comandos

Utilidad para probar MS-RPC en samba. MS-RPC es la implementación de Microsoft del estándar DCE-RPC (llamada a procedimiento remoto). Sirve para la comunicación entre procesos remotos. Permite a un programa ejecutar una subrutina o un procedimiento en otro equipo de una red como si estuvieran disponibles de forma local.

Según la ayuda del comando:

> **rpcclient** is a utility initially developed to test MS-RPC functionality in Samba itself. It has undergone several stages of development and stability. Many system administrators have now written scripts around it to manage Windows NT clients from their UNIX workstation.

Podemos usar rpcclient para abrir una sesión SMB autenticada a una máquina objetivo ejecutando el siguiente comando en la máquina donde hemos usado una sesión nula (introduciendo el username “”). Esto sólo está disponible en SMB1:

en metasploitable2:

```
# rpcclient -U ""  //10.0.1.12
Enter WORKGROUP\'s password:
rpcclient $>
```

NOTA: deja la contraseña del grupo en blanco, y si pones -N no te pide contraseña. A partir de aquí, existen multitud de posibilidades. Puedes hacerte una idea usando el comando help:

```
# rpcclient $> help
---------------         ----------------------
         MDSSVC
fetch_properties               Fetch connection properties
fetch_attributes               Fetch attributes for a CNID
---------------         ----------------------
        CLUSAPI
clusapi_open_cluster           Open cluster
clusapi_get_cluster_name               Get cluster name
clusapi_get_cluster_version            Get cluster version
clusapi_get_quorum_resource            Get quorum resource
clusapi_create_enum            Create enum query
...
```

Entre otras muchas cosas, listar usuarios, grupos, crear usuarios, resetear contraseñas, … todo dependerá de los permisos que tenga el usuario con el que se accede.

Si hiciera falta acceder con credenciales:

```
# rpcclient –U vagrant%vagrant  //10.0.1.11
rpcclient $>
```

Algunas de las operaciones de enumeración que podemos hacer, son:

1. enumerar usuarios de un dominio o de un host:

```
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[anakin_skywalker] rid:[0x3f3]
user:[artoo_detoo] rid:[0x3ef]
user:[ben_kenobi] rid:[0x3f1]
user:[boba_fett] rid:[0x3f6]
user:[chewbacca] rid:[0x3f9]
...
user:[vagrant] rid:[0x3e8]
```

enumerar grupos:

```
rpcclient $> enumdomgroups
group:[None] rid:[0x201]
```

Obtener info de un grupo:

```
rpcclient $> querygroup 0x201
       Group Name:     None
        Description:    Ordinary users
        Group Attribute:7
        Num Members:20
```

Enumerar usuarios de un grupo:

```
rpcclient $> querygroupmem 0x201
        rid:[0x1f4] attr:[0x7]
        rid:[0x1f5] attr:[0x7]
        rid:[0x3e8] attr:[0x7]
        rid:[0x3e9] attr:[0x7]
        rid:[0x3ea] attr:[0x7]
        ….
```

Consultar información de un usuario a partir de su RID:

Por ejemplo, vamos a consultar el usuario vagrant con RID 0x3e8

```
rpcclient $> queryuser 0x3e8
       User Name   :   vagrant
        Full Name   :   vagrant
```

```
        Home Drive  :
        Dir Drive   :
        Profile Path:
        Logon Script:
        Description :   Vagrant User
        Workstations:
        Comment     :
        Remote Dial :
        Logon Time               :      Fri, 12 Feb 2021 05:04:21 EST
        Logoff Time              :      Wed, 31 Dec 1969 19:00:00 EST
        Kickoff Time             :      Wed, 31 Dec 1969 19:00:00 EST
        Password last set Time   :      Sun, 19 Jul 2020 05:09:00 EDT
        Password can change Time :      Sun, 19 Jul 2020 05:09:00 EDT
        Password must change Time:      Wed, 13 Sep 30828 22:48:05 EDT
        unknown_2[0..31]...
        user_rid :      0x3e8
        group_rid:      0x201
        acb_info :      0x00000210
        fields_present: 0x00ffffff
        logon_divs:     168
        bad_password_count:     0x00000000
        logon_count:    0x00000021
        padding1[0..7]...
        logon_hrs[0..21]...
```

Puedes sacar incluso más información sobre las **políticas de contraseñas general**:

```
rpcclient $> getdompwinfo
min_password_length: 0
password_properties: 0x00000000
```

E información sobre las políticas establecidas para usuarios concretos. Consultemos en este caso, otra vez, el usuario vagrant (rid = 0x3e8)

```
rpcclient $> getusrdompwinfo  0x3e8
    &info: struct samr_PwInfo
        min_password_length      : 0x0000 (0)
        password_properties      : 0x00000000 (0)
               0: DOMAIN_PASSWORD_COMPLEX
               0: DOMAIN_PASSWORD_NO_ANON_CHANGE
               0: DOMAIN_PASSWORD_NO_CLEAR_CHANGE
               0: DOMAIN_PASSWORD_LOCKOUT_ADMINS
               0: DOMAIN_PASSWORD_STORE_CLEARTEXT
               0: DOMAIN_REFUSE_PASSWORD_CHANGE
```

En este caso concreto, no hemos obtenido nada especialmente útil, pero en otros casos, podríamos sacar el mínimo tamaño de la contraseña, si si se bloquea o no con cierto número de intentos, etc, construirnos un diccionario apropiado con *crunch* para un ataque por fuerza bruta. Y probar las combinaciones con smbclient o rpcclient por ejemplo:

```
# rpcclient –U “bobba_fet%aaa” –c “getusername;quit” //10.0.1.11
Cannot connect to server. Error was NT_STATUS_LOGON_FAILURE
# rpcclient –U “bobba_fet%password_ok” –c “getusername;quit” //10.0.1.11
Accoint Name Bobba Fett, Authority Name:
```

En lo anterior, le diríamos a rpcclient que ejecute 2 comandos: getusername y quit, para salir. Sabiendo esto, podrías hacerte construir un bucle que lea las contraseñas de un fichero y las vaya probando.

En el estudio de Thyer (2020), podemos ver aún más posibilidades de ésta herramienta.

2.4. Enumeración con metasploit

A su vez, metasploit también proporciona buenos módulos que nos permiten obtener información de SMB. Vamos a destacar algunos:

2.4.1. Extraer información de SMB

Módulo: auxiliary/scanner/smb/smb_version

Probémoslo con la máquina metasploitable2:

```
msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > set RHOST 10.0.1.12
RHOST => 10.0.1.12
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 10.0.1.12:445         - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[*] 10.0.1.12:445         -   Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 10.0.1.12:            - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Nos ha detectado la versión 1 de SMB. Aunque de manera implícita: sólo se puede negociar una versión y no tiene dialecto SMB favorito.

Si probamos con un Windows 2000:

```
[*] 10.0.1.21:445         - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[+] 10.0.1.21:445         -   Host is running Windows 2000 (language:Spanish) (name:JOHNREDYSERV)
(workgroup:WORKGROUP)
[*] 10.0.1.21:            - Scanned 1 of 1 hosts (100% complete)
```

Si probamos con Windows Server 2008 R2:

```
[*] 10.0.1.11:445               - SMB Detected (versions:1, 2) (preferred dialect:SMB 2.1)
(signatures:optional) (uptime:21s) (guid:{6ac3b828-b681-41c9-9f13-f9616b49d9a2}) (authentication
domain:METASPLOITABLE3)
[+] 10.0.1.11:445             -    Host is running Windows 2008 R2 Standard SP1 (build:7601)
(name:METASPLOITABLE3) (workgroup:WORKGROUP)
```

Es decir, soporta las versiones 1 y 2, pero la preferida es la 2.1

Otros scripts interesantes de metasploit están bajo auxiliary/scanner/smb. A saber:

- **smb_enumusers:** enumera usuarios del objetivo
- **smb_enumshares:** Enumera shares del objetivo
- **smb_ms17_010:** detecta si se es vulnerable a EternalBlue
- **smb_lookupsid:** obtiene usuarios recorriendo el SID

En cualquier caso, se recomienda ver los disponibles con la búsqueda:

```
search auxiliary/scanner/smb/
```

3. Recursos de interés

Antes de finalizar, conviene destacar algunas guías rápidas y cheat sheets, que sintetizan gran parte de la información ofrecida en este artículo:

- **Cheat sheet de herramientas para enumeración SMB**: https://highon.coffee/blog/penetration-testing-tools-cheat-sheet/#fingerprint-smb-version
- **Hoja de cálculo con comandos básicos para enumeración SMB:** https://docs.google.com/spreadsheets/d/1F9wUdEJv22HdqhSn6hy-QVtS7eumgZWYYrD-OSi6JOc/edit#gid=2080645025

4. Conclusiones

Como se comentó al principio, una buena enumeración que permita recavar toda la información posible de un objetivo es clave en un test de penetración. Siendo SMB tan extendido, resulta fundamental contar con las herramientas adecuadas, así como alternativas para obtener la misma información, ya que como se ha visto, según qué condiciones, unas son más efectivas que otras.

Se ha hecho una recopilación de utilidades Open Source en Linux y probado con máquinas vulnerables conocidas que permiten que dicho artículo pueda ser puesto en práctica de manera sencilla.

Referencias

Chandel, R. (2022). *A Little Guide to SMB Enumeration - Hacking Articles*. Hacking Articles. Retrieved 9 April 2022, from https://www.hackingarticles.in/a-little-guide-to-smb-enumeration/.

Hawes, R. (2019). *Part I: The Fundamentals of Windows Named Pipes*. VerSprite Cybersecurity Consulting Services. Retrieved 9 April 2022, from https://versprite.com/blog/security-research/microsoft-windows-pipes-intro/.

Izhar, A. (2021). *Nmap SMB Scripts and SMB Enumeration Step-By-Step Pentesting Guide*. Learn InfoSec | Cyber Security Made Easy | InfoSecAdemy. Retrieved 9 April 2022, from https://www.infosecademy.com/nmap-smb-scripts-enumeration/.

Liang, H. (2021). *Comportamiento de la sesión nula y del recurso compartido IPC$ - Windows Server*. Docs.microsoft.com. Retrieved 9 April 2022, from https://docs.microsoft.com/es-es/troubleshoot/windows-server/networking/inter-process-communication-share-null-session.

Ozturk, O. (2022). *What are SMB Ports*. Medium. Retrieved 9 April 2022, from https://medium.com/codex/what-are-smb-ports-1459040b089c.

Polop, C. (2022). *139,445 - Pentesting SMB - HackTricks*. Book.hacktricks.xyz. Retrieved 9 April 2022, from https://book.hacktricks.xyz/pentesting/pentesting-smb.

Thyer, J. (2020). *Password Spraying & Other Fun with RPCCLIENT - Black Hills Information Security*. Black Hills Information Security. Retrieved 9 April 2022, from https://www.blackhillsinfosec.com/password-spraying-other-fun-with-rpcclient/.

Biografía de los autores

Pablo Esteban Sánchez en Ingeniero en Informática por la Universidad de Almería. Actualmente es Profesor de Enseñanza Secundaria de la Junta de Andalucía en el IES Celia Viñas (Almería). Antes de ser docente ha estado trabajado en el sector privado como desarrollador durante varios años hasta finales de 2016.

Laura María Blanes Castro es Ingeniera Técnica en Informática de Gestión por la Universidad de Almería. Trabaja como docente en la especialidad de Sistemas y Aplicaciones Informáticas. Y antes de eso, en el sector privado en empresas relacionadas con eLearning principalmente. Su centro educativo actual es el IES Al-Andalus (Almería)