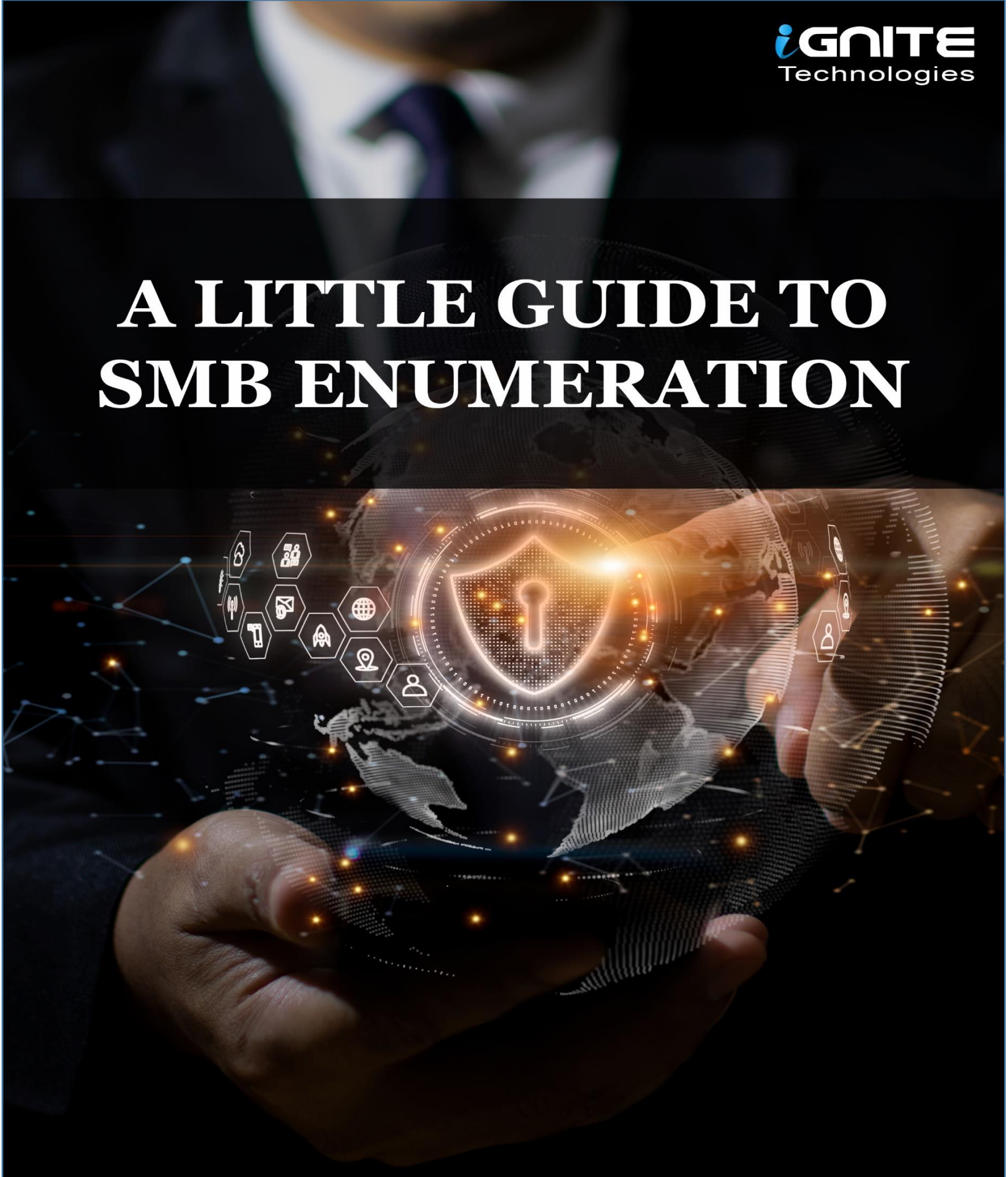
iGNITE Technologies

A LITTLE GUIDE TO SMB ENUMERATION



WWW.HACKINGARTICLES.IN

Contents

What is SMB?

SMB (Server Message Block) is the modernised concept of what was once known as the Common Internet File System. It works as an Application Layer Network Protocol. It is designed to be used as a file sharing protocol. Different applications on a system can read and write to files at the same time, as well as request services from a server located within a network.One of the interesting functionalities of SMB is that it can be run atop of the TCP/IP protocol or other network protocols. With the help of SMB, a user or any application or software that is authorised can access files or other resources on a remote server. Actions that can be performed include reading data, creating data, and updating data. Communication between clients and servers is done with the help of something called SMB client request.

SMB Working

The SMB Protocol delegates the client to communicate with other participants in the same network, allowing it to access files or services open to it on the network. In order for it to function, the other device also requires the implemented network protocol and receives and processes the respective client request using an SMB server application. Client computers using SMB connect to a supporting server using NetBIOS over TCP/IP, IPX/SPX, or NetBEUI. The initial establishment of the connection is required for exchanging information. Subsequent data transport is regulated by the provisions of the TCP protocol. SMB functions as a request-response or client-server protocol. Once the connection is established, the client computer or program can then open, read/write, and access files similar to the file system on a local computer.

SMB Versions

- CIFS: The old version of SMB, which was included in Microsoft Windows NT 4.0 in 1996.
- SMB 1.0 / SMB1: The version used in Windows 2000, Windows XP, Windows Server 2003 and Windows Server 2003 R2.
- SMB 2.0 / SMB2: This version used in Windows Vista and Windows Server 2008.
- SMB 2.1 / SMB2.1: This version used in Windows 7 and Windows Server 2008 R2.
- SMB 3.0 / SMB3: This version used in Windows 8 and Windows Server 2012.
- SMB 3.02 / SMB3: This version used in Windows 8.1 and Windows Server 2012 R2.
- SMB 3.1: This version used in Windows Server 2016 and Windows 10.

Presently, the latest version of SMB is SMB 3.1.1, which was introduced with Windows 10 and Windows Server 2016. This version supports AES 128 GCM encryption in addition to the AES 128 CCM encryption added in SMB3, and implements a pre-authentication integrity check using a SHA-512 hash. SMB 3.1.1 also makes secure negotiation mandatory when connecting to clients using SMB 2.x and higher.

SMB Security

The SMB protocol supports two levels of security. The first is the share level. The server is protected at this level, and each share has a password. On the client computer or user, you have to enter the password to access data or files saved under the specific share. This is the only security model available in the Core and Core plus SMG protocol definitions. User-level protection was later added to the SMB protocol. It is applied to individual files, and each share is based on specific user access rights. Once a server

authenticates the client, he/she is given a unique identification (UID) that is displayed upon accessing the server. The SMB protocol has supported individual security since LAN Manager 1.0 was implemented.

SMB Enumeration: Hostname

We will start the enumeration of the SMB by finding the hostname of the target machine. This can be done with various tools.

nmblookup

We started with the nmblookup tool. It is designed to make use of queries for the NetBIOS names and then map them to their subsequent IP addresses in a network. The options allow the name queries to be directed at a particular IP broadcast area or to a particular machine. All queries are done over UDP.

For unique names:

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

For group names:

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

nmblookup -A 192.168.1.17

```
┌──(root㉿kali)-[~]
└─# nmblookup -A 192.168.1.17
Looking up status of 192.168.1.17
        DESKTOP-ATNONJ9 <00> -         B <ACTIVE>
        DESKTOP-ATNONJ9 <20> -         B <ACTIVE>
        WORKGROUP       <00> - <GROUP> B <ACTIVE>
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>
        WORKGROUP       <1d> -         B <ACTIVE>
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>

        MAC Address = 00-0C-29-54-91-59
```

Here, we can see that we have enumerated the hostname to be DESKTOP-ATNONJ9.

nbtscan

Moving forward, we used nbtscan tool. NBTscan is a program for scanning IP networks for NetBIOS name information. It sends a NetBIOS status query to each address in the supplied range and lists the received information in human-readable form. For each responded host, it lists the IP address, NetBIOS computer name, logged-in user name, and MAC address (such as Ethernet).

nbtscan 192.168.1.17

```
┌──(root㉿kali)-[~]
└─# nbtscan 192.168.1.17
Doing NBT name scan for addresses from 192.168.1.17

IP address       NetBIOS Name     Server    User             MAC address
-------------------------------------------------------------------------
192.168.1.17     DESKTOP-ATNONJ9  <server>  <unknown>        00:0c:29:54:91:59
```

Here, we can see that we have enumerated the hostname to be DESKTOP-ATNONJ9.

nbstat NSE Script

This nmap script attempts to retrieve the target's NetBIOS names and MAC address. By default, the script displays the name of the computer and the logged-in user; if the verbosity is turned up, it displays all names the system thinks it owns. It also shows the flags that we studied in nmblookup tool.

nmap --script nbstat.nse 192.168.1.17

```
┌──(root㉿kali)-[~]
└─# nmap --script nbstat.nse 192.168.1.17
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-05 11:23 EST
Nmap scan report for 192.168.1.17
Host is up (0.00059s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:0C:29:54:91:59 (VMware)

Host script results:
| nbstat: NetBIOS name: DESKTOP-ATNONJ9, NetBIOS user: <unknown>, NetBIOS
| Names:
|   DESKTOP-ATNONJ9<00>  Flags: <unique><active>
|   DESKTOP-ATNONJ9<20>  Flags: <unique><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1e>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|_  \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
```

Here, we can see that we have enumerated the hostname to be DESKTOP-ATNONJ9.

nbtstat

This Windows command displays the NetBIOS over TCP/IP (NetBT) protocol statistics. It can read the NetBIOS name tables for both the local computer and remote computers. It can also read the NetBIOS name cache. This command allows a refresh of the NetBIOS name cache and the names registered with the Windows Internet Name Service (WINS). When used without any parameters, this command displays help information. This command is available only if the Internet Protocol (TCP/IP) protocol is installed as a component in the properties of a network adapter in Network Connections.

nbtstat -A 192.168.1.17

```
C:\Users\raj>nbtstat -A 192.168.1.17

Ethernet 2:
Node IpAddress: [192.168.56.1] Scope Id: []

    Host not found.

Ethernet 3:
Node IpAddress: [192.168.85.2] Scope Id: []

    Host not found.

VMware Network Adapter VMnet1:
Node IpAddress: [192.168.226.1] Scope Id: []

    Host not found.

VMware Network Adapter VMnet8:
Node IpAddress: [192.168.205.1] Scope Id: []

    Host not found.

Ethernet:
Node IpAddress: [192.168.1.3] Scope Id: []

           NetBIOS Remote Machine Name Table

       Name               Type         Status
    ---------------------------------------------
    DESKTOP-ATNONJ9<00>  UNIQUE      Registered
    DESKTOP-ATNONJ9<20>  UNIQUE      Registered
    WORKGROUP      <00>  GROUP       Registered
    WORKGROUP      <1E>  GROUP       Registered
    WORKGROUP      <1D>  UNIQUE      Registered
    ..__MSBROWSE__.<01>  GROUP       Registered

    MAC Address = 00-0C-29-54-91-59
```

Here, we can see that we have enumerated the hostname to be DESKTOP-ATNONJ9.

Ping

We can also use the ping command to detect the hostname of an SMB server or machine. The -a parameter specifies reverse name resolution to be performed on the destination IP address. If this is successful, ping displays the corresponding hostname.

```
ping -a 192.168.1.17
```

Here, we can see that we have enumerated the hostname to be DESKTOP-ATNONJ9.

smb-os-discovery NSE Script

```
C:\Users\raj>ping -a 192.168.1.17

Pinging DESKTOP-ATNONJ9 [192.168.1.17] with 32 bytes of data:
Reply from 192.168.1.17: bytes=32 time<1ms TTL=128
Reply from 192.168.1.17: bytes=32 time<1ms TTL=128
Reply from 192.168.1.17: bytes=32 time<1ms TTL=128
Reply from 192.168.1.17: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.17:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

This NSE script attempts to determine the operating system, computer name, domain, workgroup, and current time over the SMB protocol (ports 445 or 139). It is achieved by initiating a session with the anonymous account (or with a proper user account, if one is given; it likely doesn't make a difference); in response to the session starting, the server will send back all this information.

The following fields may be included in the output, depending on the circumstances (e.g., the workgroup name is mutually exclusive with domain and forest names) and the information available:

- OS
- Computer name
- Domain name
- Forest name
- FQDN
- NetBIOS computer name
- NetBIOS domain name
- Workgroup

- System time

nmap --script smb-os-discovery 192.168.1.17

```
┌──(root kali)-[~]
└─# nmap --script smb-os-discovery 192.168.1.17
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-05 12:03 EST
Nmap scan report for 192.168.1.17
Host is up (0.00069s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:0C:29:54:91:59 (VMware)

Host script results:
| smb-os-discovery:
|   OS: Windows 10 Pro 18362 (Windows 10 Pro 6.3)
|   OS CPE: cpe:/o:microsoft:windows_10::-
|   Computer name: DESKTOP-ATNONJ9
|   NetBIOS computer name: DESKTOP-ATNONJ9\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2021-03-05T09:03:24-08:00

Nmap done: 1 IP address (1 host up) scanned in 1.83 seconds
```

Here, we can see that we have enumerated the hostname to be DESKTOP-ATNONJ9.

SMB Enumeration: Share and Null Session

As we discussed earlier, SMB works on sharing files and resources. In order to transfer these files or resources, there are data streams that are called shares. There are public shares that are accessible to everyone on the network, and then there are user-specific shares. Let's enumerate these shares.

SMBMap

SMBMap allows users to enumerate samba share drives across an entire domain. List share drives, drive permissions, share contents, upload/download functionality, file name auto-download pattern matching, and even execute remote commands. This tool was designed with pen testing in mind and is intended to simplify searching for potentially sensitive data across large networks.

smbmap -H 192.168.1.40

```
┌──(root kali)-[~]
└─# smbmap -H 192.168.1.40
[+] Guest session       IP: 192.168.1.40:445    Name: 192.168.1.40
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        guest                                                   READ, WRITE
        IPC$                                                    NO ACCESS       IPC Service (ubuntu
```

Here we see that the target machine has some shares. There is a share by the name of the guest. That must be a public share. Let's enumerate a user-specific share using the credentials for that user. We are enumerating the shares for the user raj as shown in the image below.

smbmap -H 192.168.1.17 -u raj -p 123

```
┌──(root kali)-[~]
└─# smbmap -H 192.168.1.17 -u raj -p 123
[+] IP: 192.168.1.17:445        Name: 192.168.1.17
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    READ ONLY       Remote IPC
        share                                                   READ, WRITE
        Users                                                   READ ONLY
```

smbclient

Samba client with an "FTP-like" interface is smbclient.It is a useful tool to test connectivity to a Windows share. It can be used to transfer files, or to look at share names. In addition, it has a nifty ability to 'tar' (backup) and restore files from a server to a client and vice versa. We enumerated the target machine and found the guest share using the SMBClient directly. Then we connect to the guest share and see that there is a text file named file.txt. We can download it using the get command.

smbclient -L 192.168.1.40
smbclient //192.168.1.40/guest
get file.txt

```
┌──(root㉿kali)-[~]
└─# smbclient -L 192.168.1.40
Enter WORKGROUP\root's password:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        guest           Disk
        IPC$            IPC       IPC Service (ubuntu server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available

┌──(root㉿kali)-[~]
└─# smbclient //192.168.1.40/guest
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Mar  5 13:01:46 2021
  ..                                  D        0  Fri Mar  5 13:00:54 2021
  file.txt                            N       25  Fri Mar  5 13:01:17 2021

                20509264 blocks of size 1024. 15451640 blocks available
smb: \> get file.txt
getting file \file.txt of size 25 as file.txt (6.1 KiloBytes/sec) (average 6.1
smb: \> exit
```

Now we enumerate the user-specific share. We connect to the SMB as user raj and find a share by the name of 'share'. We reconfigured the smbclient command to access the share and we see that we find a file named raj.txt. Again, we can download this file as well as use the get command.

```
smbclient -L 192.168.1.17 -U raj%123
smbclient //192.168.1.17/share -U raj%123
get raj.txt
```

```
┌──(root㉿kali)-[~]
└─# smbclient -L 192.168.1.17 -U raj%123

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        share           Disk
        Users           Disk
SMB1 disabled -- no workgroup available

┌──(root㉿kali)-[~]
└─# smbclient //192.168.1.17/share -U raj%123
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Mar  5 11:51:13 2021
  ..                                  D        0  Fri Mar  5 11:51:13 2021
  raj.txt                             A        4  Fri Mar  5 10:22:51 2021

                15563263 blocks of size 4096. 9657922 blocks available
smb: \> get raj.txt
getting file \raj.txt of size 4 as raj.txt (0.4 KiloBytes/sec) (average 0.4 KiloBytes/sec)
smb: \>
```

smb-enum-shares NSE Script

This NSE script attempts to list shares using the srvsvc.NetShareEnumAll MSRPC function and retrieve more information about them using srvsvc. NetShareGetInfo. If access to those functions is denied, a list of common share names is checked. Calling NetShareGetInfo requires an administrator account on all versions of Windows up to 2003, as well as Windows Vista, Windows 7, and Windows 10, if UAC is turned off. Even if NetShareEnumAll is restricted, attempting to connect to a share will always reveal its existence. So, if NetShareEnumAll fails, a pre-generated list of shares, based on a large test network, is used. If any of those succeed, they are recorded. After a list of shares is found, the script attempts to connect to each of them anonymously, which divides them into "anonymous," for shares that the NULL user can connect to, or "restricted," for shares that require a user account.

```
nmap --script smb-enum-shares  -p139,445 192.168.1.17
```

```
┌──(root㉿kali)-[~]
└─# nmap --script smb-enum-shares -p139,445 192.168.1.17
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-05 11:59 EST
Nmap scan report for 192.168.1.17
Host is up (0.00054s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:54:91:59 (VMware)

Host script results:
| smb-enum-shares:
|   note: ERROR: Enumerating shares failed, guessing at common ones (NT_STAT
|   account_used: <blank>
|   \\192.168.1.17\ADMIN$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\192.168.1.17\C$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\192.168.1.17\IPC$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: READ
|   \\192.168.1.17\SHARE:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\192.168.1.17\USERS:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|_    Anonymous access: <none>

Nmap done: 1 IP address (1 host up) scanned in 1.18 seconds
```

Here, we can see that we have the shares listed although the Access is Denied the existence of the share is confirmed.

Net view

Displays a list of domains, computers or resources that are being shared by the specified computer. Used without parameters, net view displays a list of computers in your current domain. This time we are on the Windows machine. We used the net view with the /all parameter to list all the shares on the target machine.

```
net view \\192.168.1.17 /All
```

```
C:\Users\raj>net view \\192.168.1.17 /All
Shared resources at \\192.168.1.17


Share name  Type  Used as  Comment

-------------------------------------------------------------------------------
ADMIN$      Disk           Remote Admin
C$          Disk           Default share
IPC$        IPC            Remote IPC
share       Disk
Users       Disk
The command completed successfully.


C:\Users\raj>net use \\192.168.1.17\share
The command completed successfully.


C:\Users\raj>net use
New connections will be remembered.


Status       Local     Remote                    Network

-------------------------------------------------------------------------------
OK                     \\192.168.1.17\share      Microsoft Windows Network
Disconnected           \\192.168.1.16\IPC$       Microsoft Windows Network
The command completed successfully.


C:\Users\raj>copy \\192.168.1.17\share\raj.txt
        1 file(s) copied.

C:\Users\raj>
```

Then we changed the command by adding the share, and we were able to read the contents of that share. Now, using the copy command, we can download the file from the share.

Metasploit: smb_enumshares

The smb_enumshares module enumerates any SMB shares that are available on a remote system. It requires the IP Address of the target server or machine, followed by a set of credentials that can be used to access the share.

```
use auxiliary/scanner/smb/smb_enumshares
set rhosts 192.168.1.17
set smbuser raj
set smbpass 123
exploit
```

```
msf6 > use auxiliary/scanner/smb/smb_enumshares
msf6 auxiliary(scanner/smb/smb_enumshares) > set rhosts 192.168.1.17
rhosts => 192.168.1.17
msf6 auxiliary(scanner/smb/smb_enumshares) > set smbuser raj
smbuser => raj
msf6 auxiliary(scanner/smb/smb_enumshares) > set smbpass 123
smbpass => 123
msf6 auxiliary(scanner/smb/smb_enumshares) > exploit

[-] 192.168.1.17:139      - Login Failed: Unable to negotiate SMB1 with the remote host: Not a
[!] 192.168.1.17:445      - peer_native_os is only available with SMB1 (current version: SMB3)
[!] 192.168.1.17:445      - peer_native_lm is only available with SMB1 (current version: SMB3)
[+] 192.168.1.17:445      - ADMIN$ - (DISK) Remote Admin
[+] 192.168.1.17:445      - C$ - (DISK) Default share
[+] 192.168.1.17:445      - IPC$ - (IPC) Remote IPC
[+] 192.168.1.17:445      - share - (DISK)
[+] 192.168.1.17:445      - Users - (DISK)
[*] 192.168.1.17:         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_enumshares) >
```

CrackMapExec

CrackMapExec (a.k.a CME) is a post-exploitation tool that helps automate assessing the security of large Active Directory networks. Built with stealth in mind, CME follows the concept of "Living off the Land": abusing built-in Active Directory features/protocols to achieve its functionality and allowing it to evade most endpoint protection/IDS/IPS solutions. CrackMapExec can map the network hosts, generate relay lists, enumerate shares and access, enumerate active sessions, enumerate disks, enumerate logged on users, enumerate domain users, enumerate users by bruteforcing RID, enumerate domain groups, enumerate local groups, etc.

```
crackmapexec smb 192.168.1.40 -u 'raj' -p '123' --shares
```

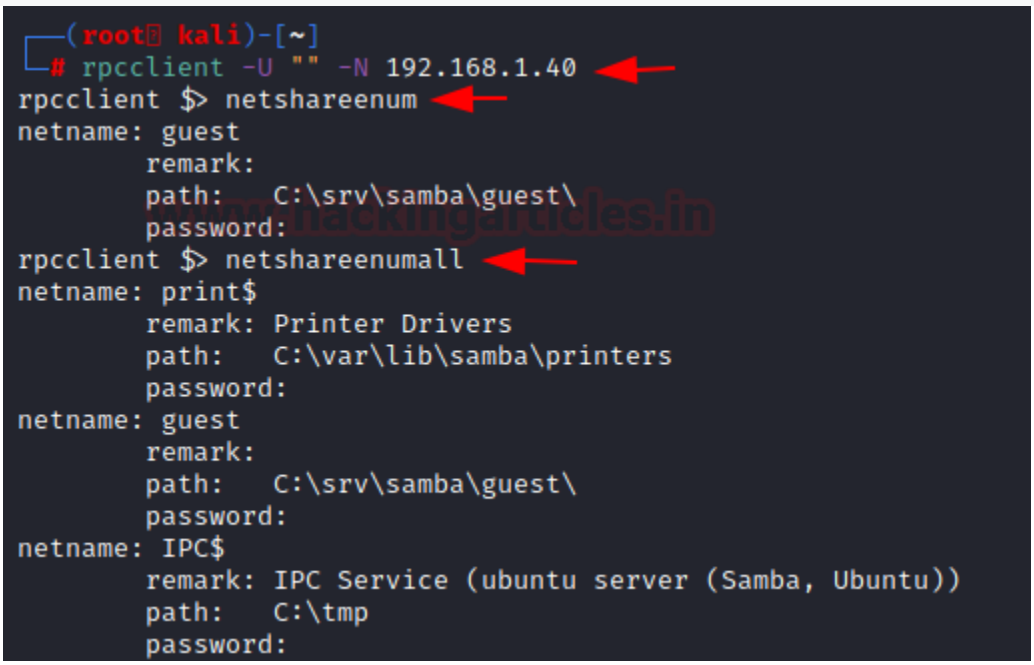
```
┌──(root㉿kali)-[~]
└─# crackmapexec smb 192.168.1.40 -u '' -p '' --shares
SMB         192.168.1.40    445    UBUNTU           [*] Windows 6.1 (name:UBUNTU) (domain:) (signing:False) (SMBv1:True)
SMB         192.168.1.40    445    UBUNTU           [+] \:
SMB         192.168.1.40    445    UBUNTU           [+] Enumerated shares
SMB         192.168.1.40    445    UBUNTU           Share           Permissions     Remark
SMB         192.168.1.40    445    UBUNTU           -----           -----------     ------
SMB         192.168.1.40    445    UBUNTU           print$                          Printer Drivers
SMB         192.168.1.40    445    UBUNTU           guest           READ,WRITE
SMB         192.168.1.40    445    UBUNTU           IPC$                            IPC Service (ubuntu server (Samba, Ubuntu))
```

Here, we can see different shares and the permissions that are allowed on that particular share.

rpcclient

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 workstations. We will be using it to enumerate the users on the SMB shares using the option of netshareenum, as shown in the image below.

```
rpcclient -U "" -N 192.168.1.40
netshareenum
netshareenumall
```



SMB Enumeration: Vulnerability Scanning

Enumerate an SMB server in order to compromise it. We need to enumerate and find possible vulnerabilities that can be used to exploit the server. In order to do this in an optimised method, we can perform a vulnerability scan. There might be multiple tools to perform this kind of scanning, but here we will be focusing on this NSE script.

smb-vuln NSE Script

Nmap in the past used to have a script by the name of smb-check-vulns. It is used to scan the target server for various vulnerabilities, such as:

- conficker
- cve2009-3103
- ms06-025
- ms07-029

- regsvc-dos
- ms08-067

Then the script was divided into single vulnerability checks that could be run individually, such as smb-vuln-ms08-067. Hence, to check all SMB vulnerabilities available in the Nmap Scripting Engine, we use the * with the script.

nmap --script smb-vuln* 192.168.1.16

```
┌──(root kali)-[~]
└─# nmap --script smb-vuln* 192.168.1.16
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-05 14:33 EST
Nmap scan report for 192.168.1.16
Host is up (0.00061s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:0C:29:5C:69:16 (VMware)

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
| 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-wan
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
```

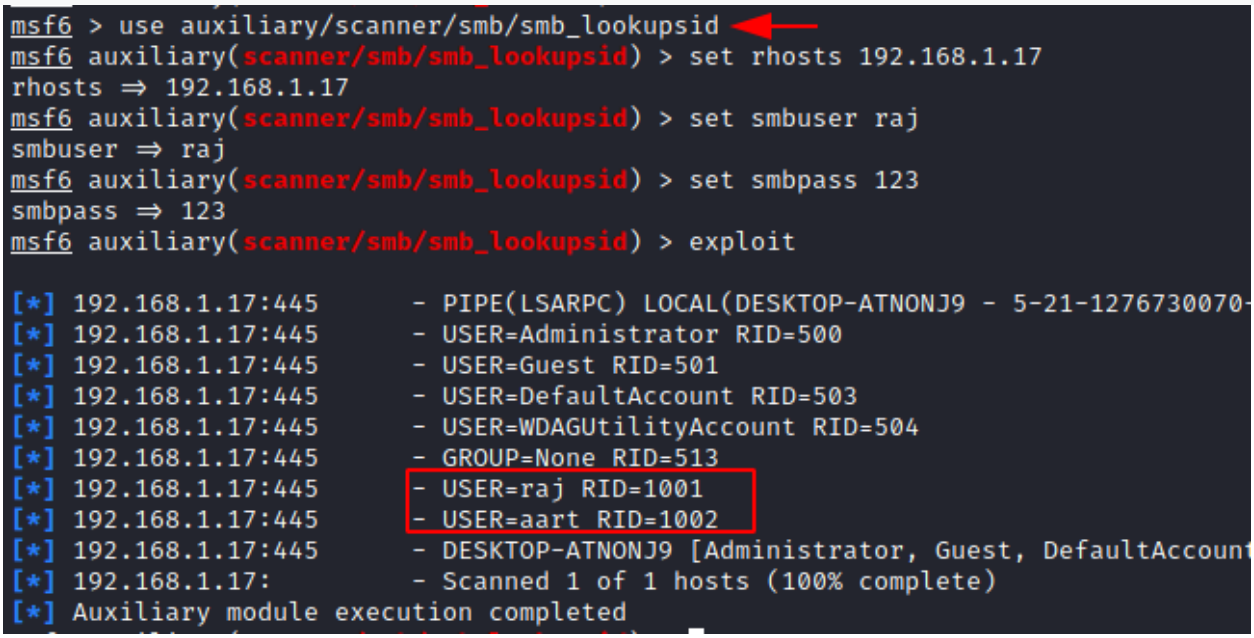
SMB Enumeration: Users

In a Windows environment, each user is assigned a unique identifier called a Security ID, or SID, which is used to control access to various resources like files, registry keys, network shares, etc. Hence, the SID of a user shouldn't be compromised.

smb_lookupsid

The smb_lookupsid module brute-forces SID lookups on a range of targets to determine what local users exist in the system. Knowing what users exist on a system can greatly speed up any further brute-force logon attempts later on.

```
use auxiliary/scanner/smb/smb_lookupsid
set rhosts 192.168.1.17
set smbuser raj
set smbpass 123
exploit
```



Here, we can see that through enumerating SMB we have extracted two users: raj and aarti.

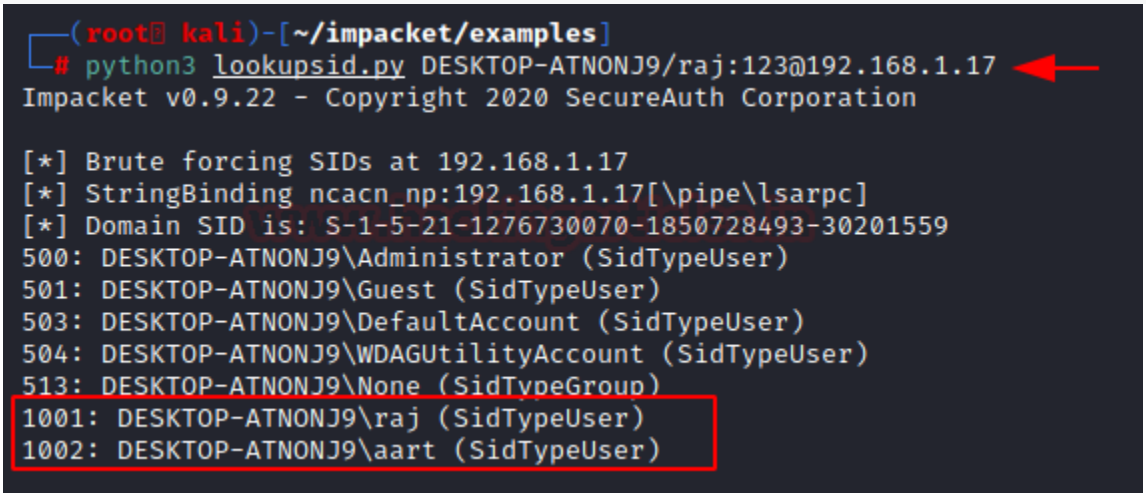
Impacket: Lookupsid

A Security Identifier (SID) is a unique value of variable length that is used to identify a user account. Through a SID User Enumeration, we can extract information about which users exist and their data. The Lookupsid script can enumerate both local and domain users. There is a Metasploit module too for this attack. If you are planning on injecting a target server with a golden or silver ticket, then one of the things that is required is the SID of the 500 user. Lookupsid.py can be used in that scenario. When we provide the following parameters to the Lookupsid in such a format as shown below.
Requirements:

- Domain
- Username

- Password/Password Hash
- Target IP Address

```
python3 lookupsid.py DESKTOP-ATNONJ9/raj:123@192.168.1.17
```



SMB Enumeration: Enum4Linux

Enum4linux is a tool that is designed to detect and extract data or enumerate from Windows and Linux operating systems, including SMB hosts that are on a network. Enum4linux can discover the following:

- Domain and group membership
- User listings
- Shares on a device (drives and folders)
- Password policies on a target
- The operating system of a remote target

We start to normal scan using enum4linux. It extracts the RID Range, Usernames, Workgroup, Nbtstat Information, Sessions, SID Information, OS Information.

```
enum4linux 192.168.1.40
```

```
┌──(root㉿kali)-[~]
└─# enum4linux 192.168.1.40
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on

 ==========================
|    Target Information    |
 ==========================
Target ........... 192.168.1.40
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ============================================================
|    Enumerating Workgroup/Domain on 192.168.1.40            |
 ============================================================
[+] Got domain/workgroup name: WORKGROUP

 =====================================================
|    Nbtstat Information for 192.168.1.40             |
 =====================================================
Looking up status of 192.168.1.40
        UBUNTU          <00> -         B <ACTIVE>  Workstation Service
        UBUNTU          <03> -         B <ACTIVE>  Messenger Service
        UBUNTU          <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

 ==================================================
|    Session Check on 192.168.1.40                 |
 ==================================================
[+] Server 192.168.1.40 allows sessions using username '', password ''

 ===================================================
|    Getting domain SID for 192.168.1.40            |
 ===================================================
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

 ==================================================
|    OS information on 192.168.1.40                |
 ==================================================
Use of uninitialized value $os_info in concatenation (.) or string at ./enum4linux.pl
[+] Got OS info for 192.168.1.40 from smbclient:
[+] Got OS info for 192.168.1.40 from srvinfo:
        UBUNTU         Wk Sv PrQ Unx NT SNT ubuntu server (Samba, Ubuntu)
        platform_id     :       500
        os version      :       6.1
        server type     :       0x809a03
```

We see that it has also extracted the two uses based on the SID. These two users are privs and ignite. This user's information was extracted through communicating via the SMB channels by the enum4linux script.

```
S-1-5-32-1050 *unknown*\*unknown* (8)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\privs (Local User)
S-1-22-1-1001 Unix User\ignite (Local User)
[+] Enumerating users using SID S-1-5-21-894636310-4219792968-1492264695 and l
S-1-5-21-894636310-4219792968-1492264695-500 *unknown*\*unknown* (8)
S-1-5-21-894636310-4219792968-1492264695-501 UBUNTU\nobody (Local User)
S-1-5-21-894636310-4219792968-1492264695-502 *unknown*\*unknown* (8)
S-1-5-21-894636310-4219792968-1492264695-503 *unknown*\*unknown* (8)
S-1-5-21-894636310-4219792968-1492264695-504 *unknown*\*unknown* (8)
S-1-5-21-894636310-4219792968-1492264695-505 *unknown*\*unknown* (8)
S-1-5-21-894636310-4219792968-1492264695-506 *unknown*\*unknown* (8)
```

Finally, we have the Share Enumeration, which had the guest share that we enumerated earlier. Then we see that it tried to enumerate inside the print share and IPC but was restricted. Then we have the Password Policy Information regarding the users on the system. It enumerates if the password was changed recently or if it has never been changed. It also tells us the complexity and other details regarding users and the operating system of the target system.

```
 ═══════════════════════════════════════════
|    Share Enumeration on 192.168.1.40      |
 ═══════════════════════════════════════════

        Sharename       Type      Comment
        ─────────       ────      ───────
        print$          Disk      Printer Drivers
        guest           Disk
        IPC$            IPC       IPC Service (ubuntu server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available

[+] Attempting to map shares on 192.168.1.40
//192.168.1.40/print$   Mapping: DENIED, Listing: N/A
//192.168.1.40/guest    Mapping: OK, Listing: OK
//192.168.1.40/IPC$     [E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*

 ═══════════════════════════════════════════════════
|    Password Policy Information for 192.168.1.40   |
 ═══════════════════════════════════════════════════


[+] Attaching to 192.168.1.40 using a NULL share

[+] Trying protocol 139/SMB ...

[+] Found domain(s):

        [+] UBUNTU
        [+] Builtin

[+] Password Info for Domain: UBUNTU

        [+] Minimum password length: 5
        [+] Password history length: None
        [+] Maximum password age: 37 days 6 hours 21 minutes
        [+] 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: 37 days 6 hours 21 minutes
```

Conclusion

In this discussion, we understood the various scripts and tools that can be used to enumerate the SMB/MSRPC services on a target system. Enumeration is the key step in order to compromise and to defend your system and network. Be sure to safeguard your SMB service.