Road To Offensive Security Certified Professional

Pentest Report

aymanrayan.kissami@gmail.com, OSID: XXXX

2022-06-09

Contents

1 Kenobi Pentensting Report

1.1 Introduction

The penetration test report contains all efforts that were conducted in order to get access to the machine . This report will be graded from a standpoint of correctness and fullness to all aspects of the Pentest. The purpose of this report is to ensure that the client has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Box. The Pentester is tasked with following methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report.

1.3 Requirements

The Pentester will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards this Box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal systems - the THINC.local domain. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the Box. During the testing, I had administrative level access to the system. The full box was successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.98.191(Kenobi) - Samba shares

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresse was:

Box IP

- 10.10.98.191

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **X** out of the **X** systems.

3.2.1 System IP:10.10.98.191

3.2.1.1 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.98.191	**TCP**:21,22,80,111,139,445,2049 **UDP**: 49228,52315,34095,52046

Nmap Scan Results:

=> since we saw a lot of classic samba ports open we will go ahead and perform a nomal scan first

```
└─# nmap -sC -sV -A 10.10.98.191
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-09 07:13 EDT
Nmap scan report for 10.10.98.191
Host is up (0.081s latency).
Not shown: 993 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         ProFTPD 1.3.5
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 b3:ad:83:41:49:e9:5d:16:8d:3b:0f:05:7b:e2:c0:ae (RSA)
|   256 f8:27:7d:64:29:97:e6:f8:65:54:65:22:f7:c8:1d:8a (ECDSA)
|_  256 5a:06:ed:eb:b6:56:7e:4c:01:dd:ea:bc:ba:fa:33:79 (ED25519)
80/tcp   open  http        Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/admin.html
|_http-title: Site doesn't have a title (text/html).
111/tcp  open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100003  2,3,4       2049/udp   nfs
|   100003  2,3,4       2049/udp6  nfs
|   100005  1,2,3      49228/udp6  mountd
|   100005  1,2,3      52315/udp   mountd
|   100005  1,2,3      52419/tcp   mountd
|   100005  1,2,3      52991/tcp6  mountd
|   100021  1,3,4      34095/udp6  nlockmgr
|   100021  1,3,4      34281/tcp   nlockmgr
|   100021  1,3,4      39001/tcp6  nlockmgr
|   100021  1,3,4      52046/udp   nlockmgr
|   100227  2,3         2049/tcp   nfs_acl
|   100227  2,3         2049/tcp6  nfs_acl
|   100227  2,3         2049/udp   nfs_acl
|_  100227  2,3         2049/udp6  nfs_acl
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
2049/tcp open  nfs_acl     2-3 (RPC #100227)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=7/9%OT=21%CT=1%CU=34385%PV=Y%DS=2%DC=T%G=Y%TM=62C962E8
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=107%TI=Z%CI=RI%II=I%TS=8)OPS
OS:(O1=M508ST11NW6%O2=M508ST11NW6%O3=M508NNT11NW6%O4=M508ST11NW6%O5=M508ST1
OS:1NW6%O6=M508ST11)WIN(W1=68DF%W2=68DF%W3=68DF%W4=68DF%W5=68DF%W6=68DF)ECN
OS:(R=Y%DF=Y%T=40%W=6903%O=M508NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=A
OS:S%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F
OS:=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%
OS:T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD
OS:=S)

Network Distance: 2 hops
Service Info: Host: KENOBI; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 3.1: Fast Scan

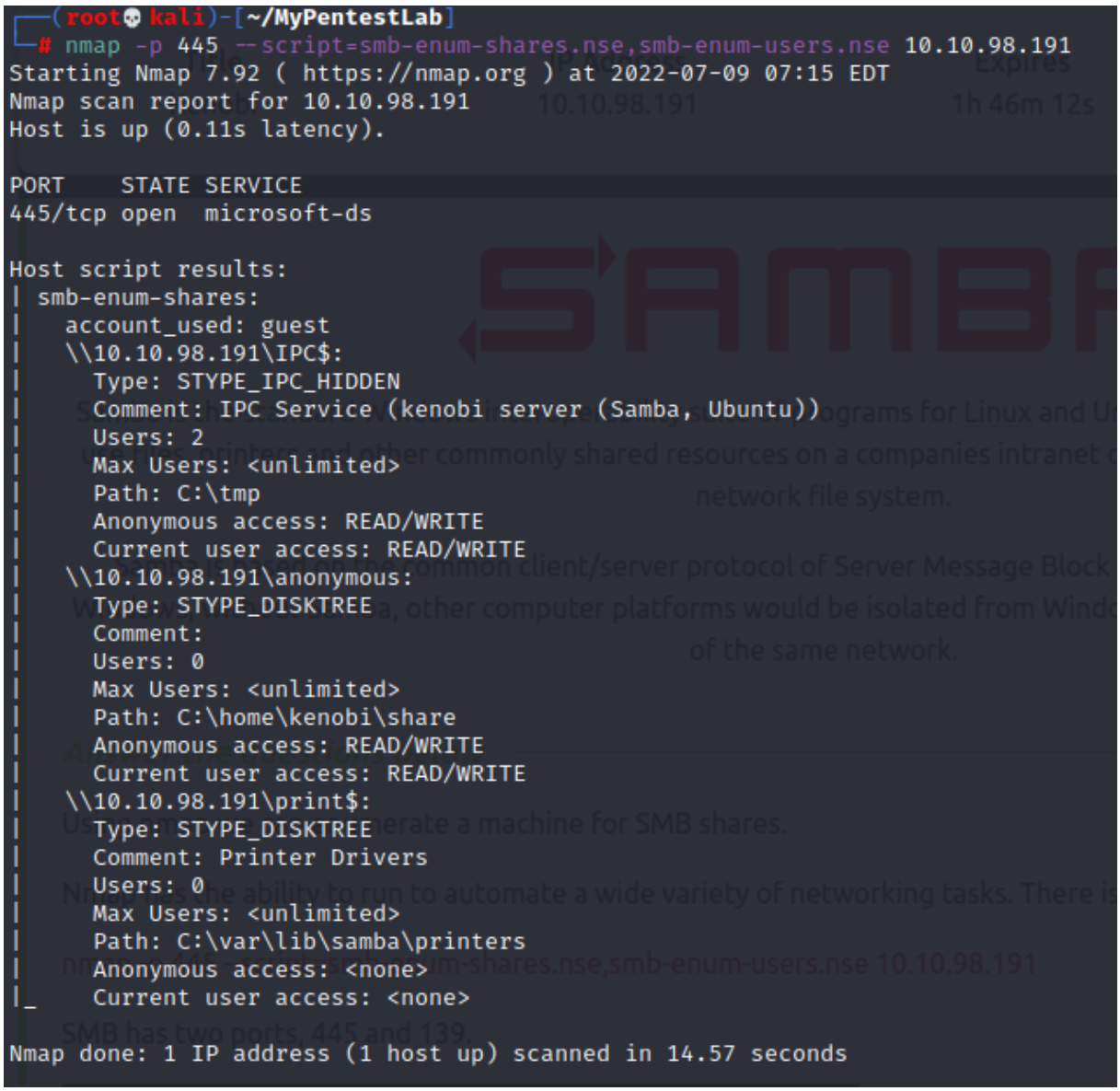


Figure 3.2: Deep scan

Samba

=> samba is a 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 often referred to as a network file system

=> samba is based on the common client server protocol of server message block SMB , its developed only for windows , without samba other computer plateforms would be isolated from windows machines even if they were part of the same network

```
nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.98.191
```

this command will enumerate the shares and users on the machine

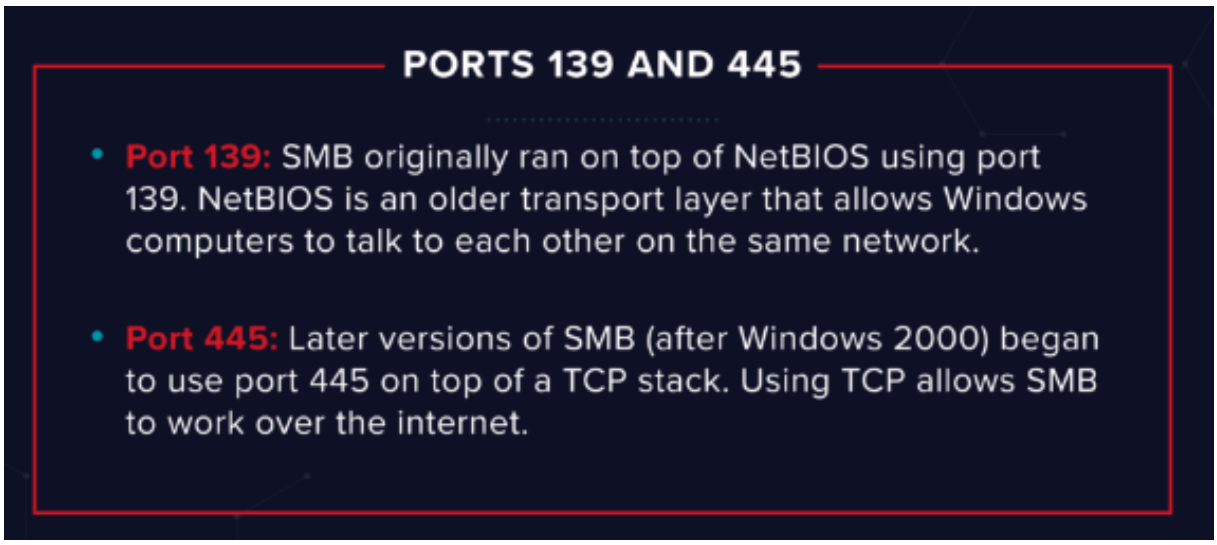


Figure 3.3: Samba Ports

–> let's inspect one of the shares

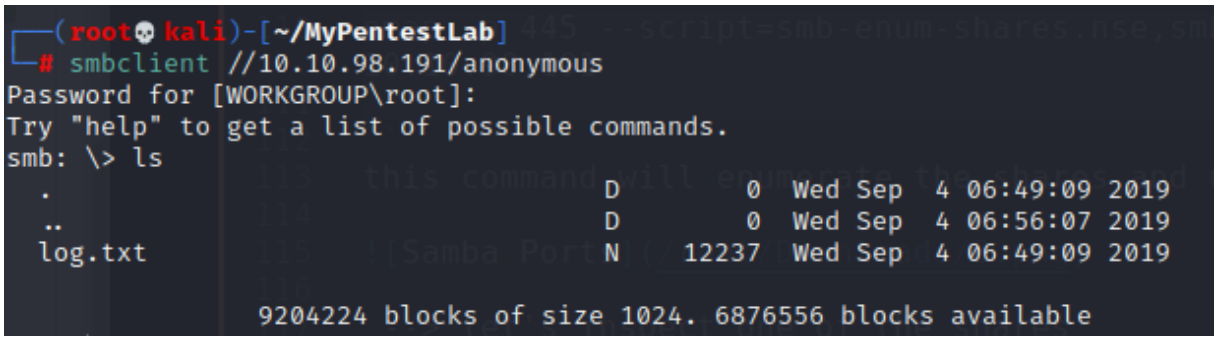


Figure 3.4: smbclient

–> now we can see that log.txt may be interesting , we can download it to our local machine using smbget

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

– this will recursively download the share , submit the username and password as nothing

Enum4linux

- we can use also enum4linux for further samba enumeration

```
enum4linux -a <ip>
```

– 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 <ip>
```

```
┌──(root💀kali)-[~/MyPentestLab]
└─# nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.98.191
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-09 07:47 EDT
Nmap scan report for 10.10.98.191
Host is up (0.17s latency).

PORT    STATE SERVICE
111/tcp open  rpcbind
| nfs-statfs:
|   Filesystem  1K-blocks  Used       Available  Use%  Maxfilesize  Maxlink
|_  /var        9204224.0  1837076.0  6876552.0  22%   16.0T        32000
| nfs-showmount:
|_  /var *
| nfs-ls: Volume /var
|   access: Read Lookup NoModify NoExtend NoDelete NoExecute
| PERMISSION  UID  GID  SIZE  TIME                 FILENAME
| rwxr-xr-x   0    0    4096  2019-09-04T08:53:24  .
| rwxr-xr-x   0    0    4096  2019-09-04T12:27:33  ..
| rwxr-xr-x   0    0    4096  2022-07-09T11:25:01  backups
| rwxr-xr-x   0    0    4096  2019-09-04T10:37:44  cache
| rwxrwxrwt   0    0    4096  2019-09-04T08:43:56  crash
| rwxrwsr-x   0    50   4096  2016-04-12T20:14:23  local
| rwxrwxrwx   0    0    9     2019-09-04T08:41:33  lock
| rwxrwxr-x   0    108  4096  2019-09-04T10:37:44  log
| rwxr-xr-x   0    0    4096  2019-01-29T23:27:41  snap
| rwxr-xr-x   0    0    4096  2019-09-04T08:53:24  www
|_
Nmap done: 1 IP address (1 host up) scanned in 1.91 seconds
```

Figure 3.5: rpc

- we can see that the mount is /var directory

– other way to do this is using showmount

```
showmount -e <ip>
```

PROFTP

- ProFTP is an opensource ftp server compatible with unix and windows it also been vuln in the past software versions

```
nc <ip> 21 //to connect to the proftp machine
```

- since the ProFTPd is an older version , we can use searchsploit to look for an exploit

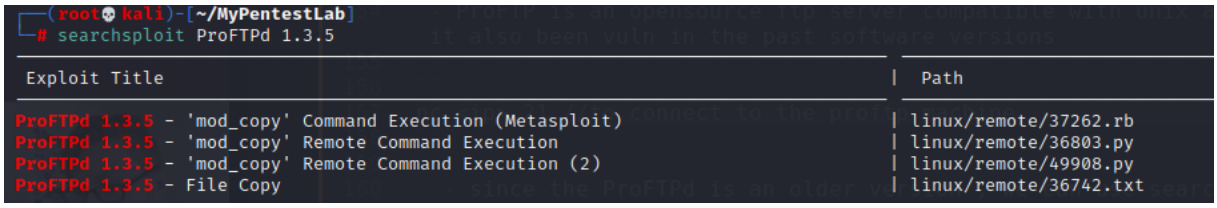


Figure 3.6: searchsploit

Vulnerability Explanation:

- 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 levrage these commands to copy files from any part of the file system to a chosen destination
- so we can potentially pull that ssh key into a location we can read and access

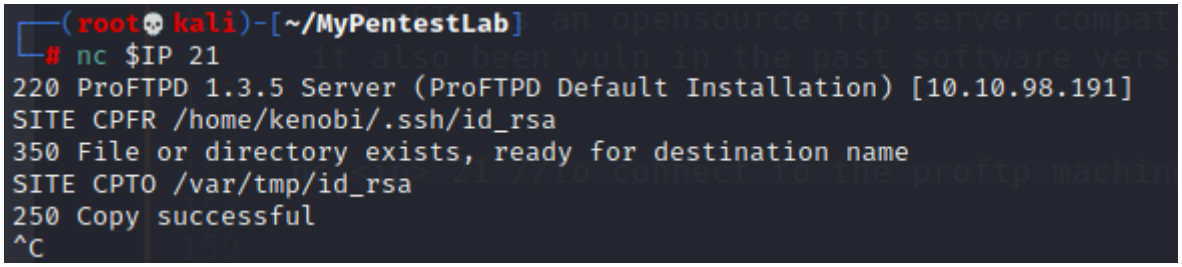
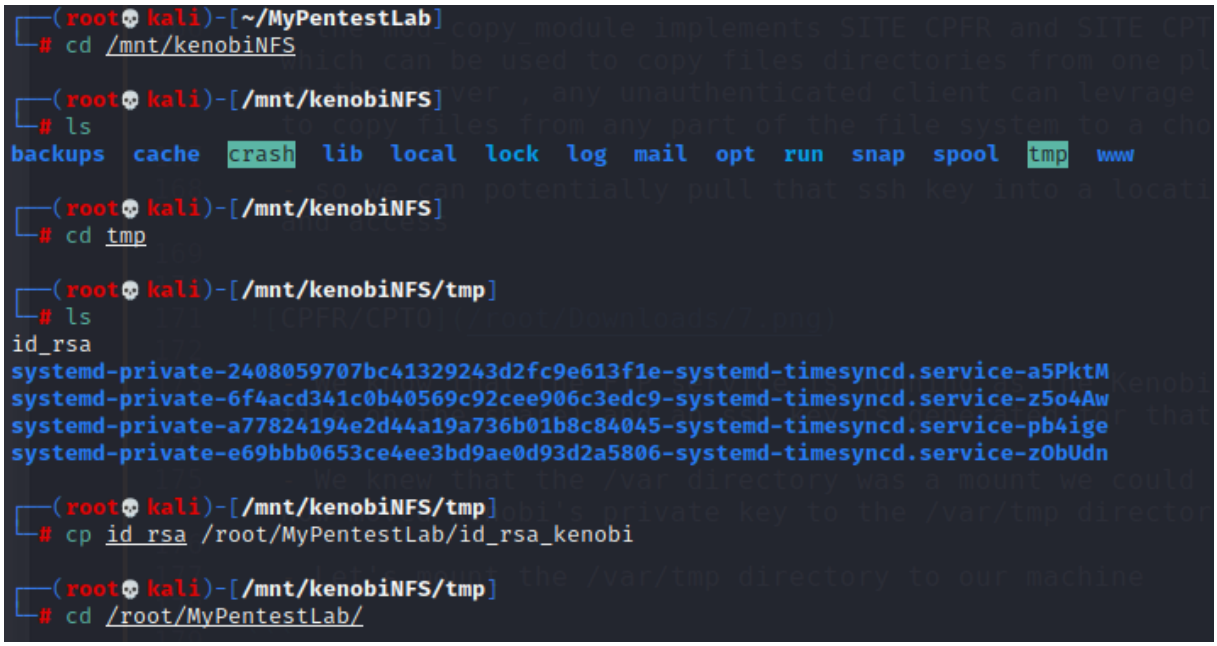


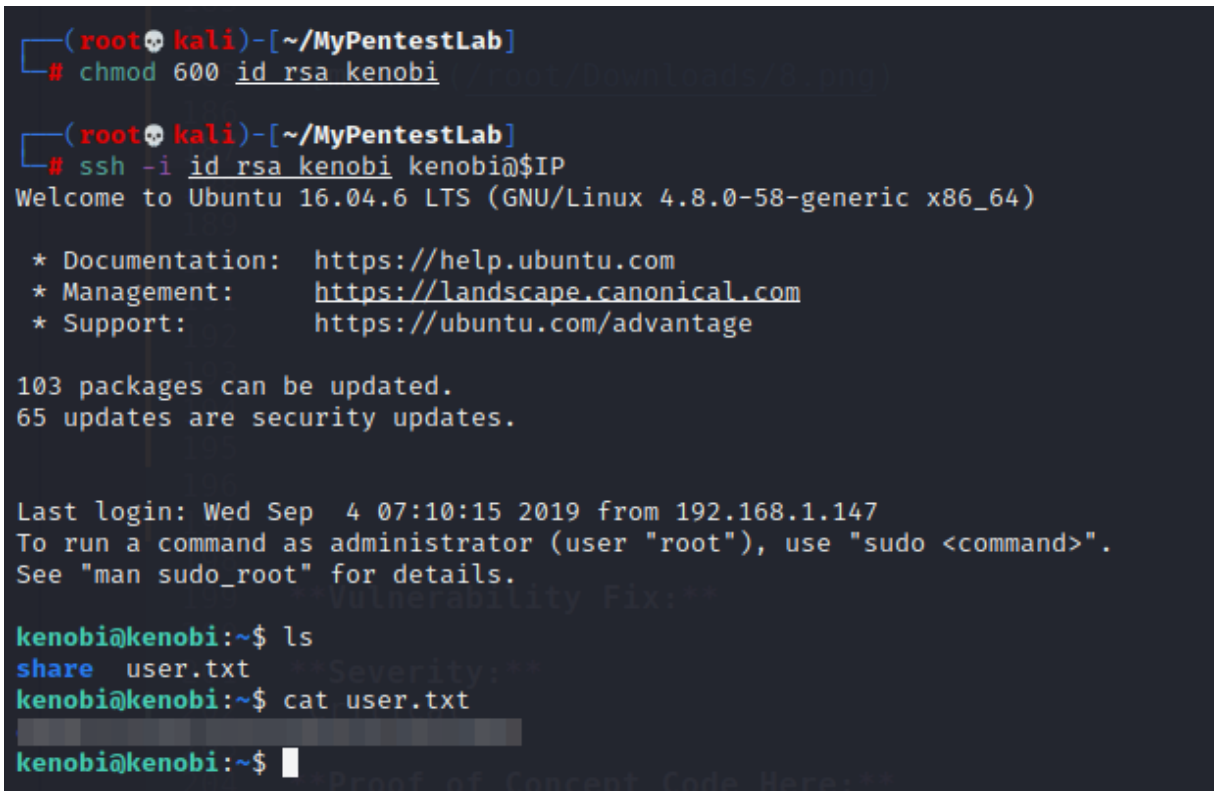
Figure 3.7: CPFR/CPTO

- 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 knew that the /var directory was a mount we could see. So we've now moved Kenobi'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
```





– and we have access to the machine let's see how to get root

Vulnerability Fix:

Severity: moderate

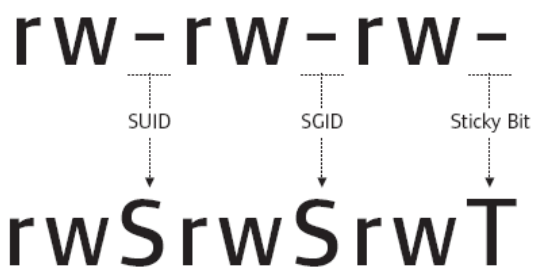
Proof of Concept Code Here:

Local.txt Proof Screenshot

Local.txt Contents

3.2.1.2 Privilege Escalation

Additional Priv Esc info



Lets first understand what what SUID, SGID and Sticky Bits are.

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.

Figure 3.8: privesc

- SUID bits can be dangerous some binaries such as passwd need to run with elevated privs(cz its resetting ur pass on the system) however other custom files 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
```

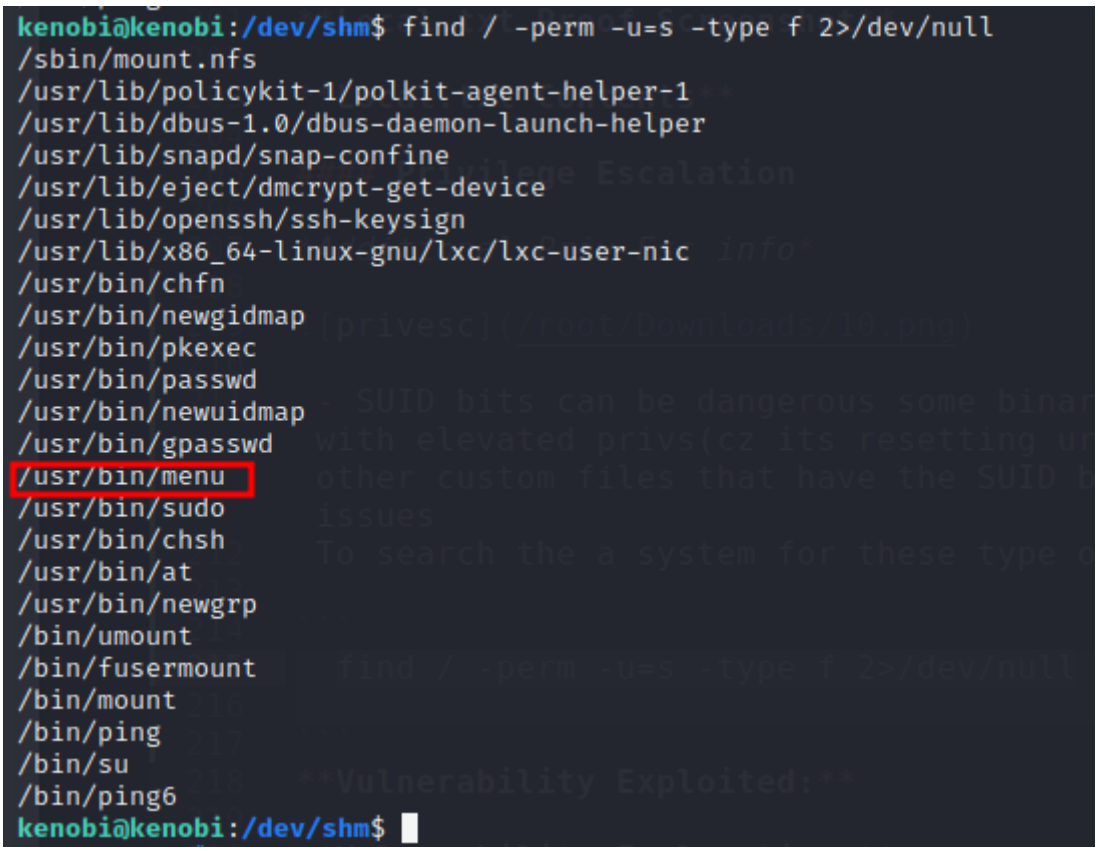


Figure 3.9: privesc

- the file menu seems a little odd

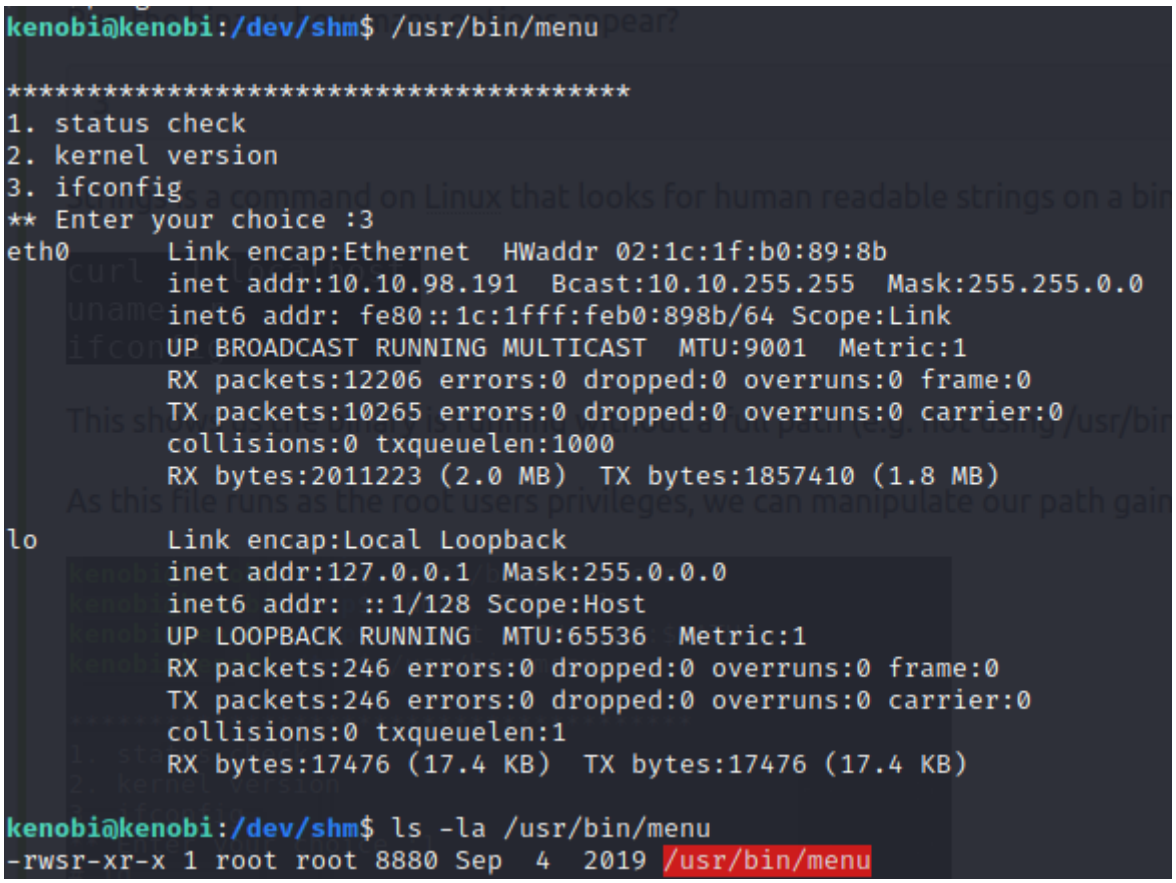


Figure 3.10: privesc

- strings command looks for human readable chars in the binary

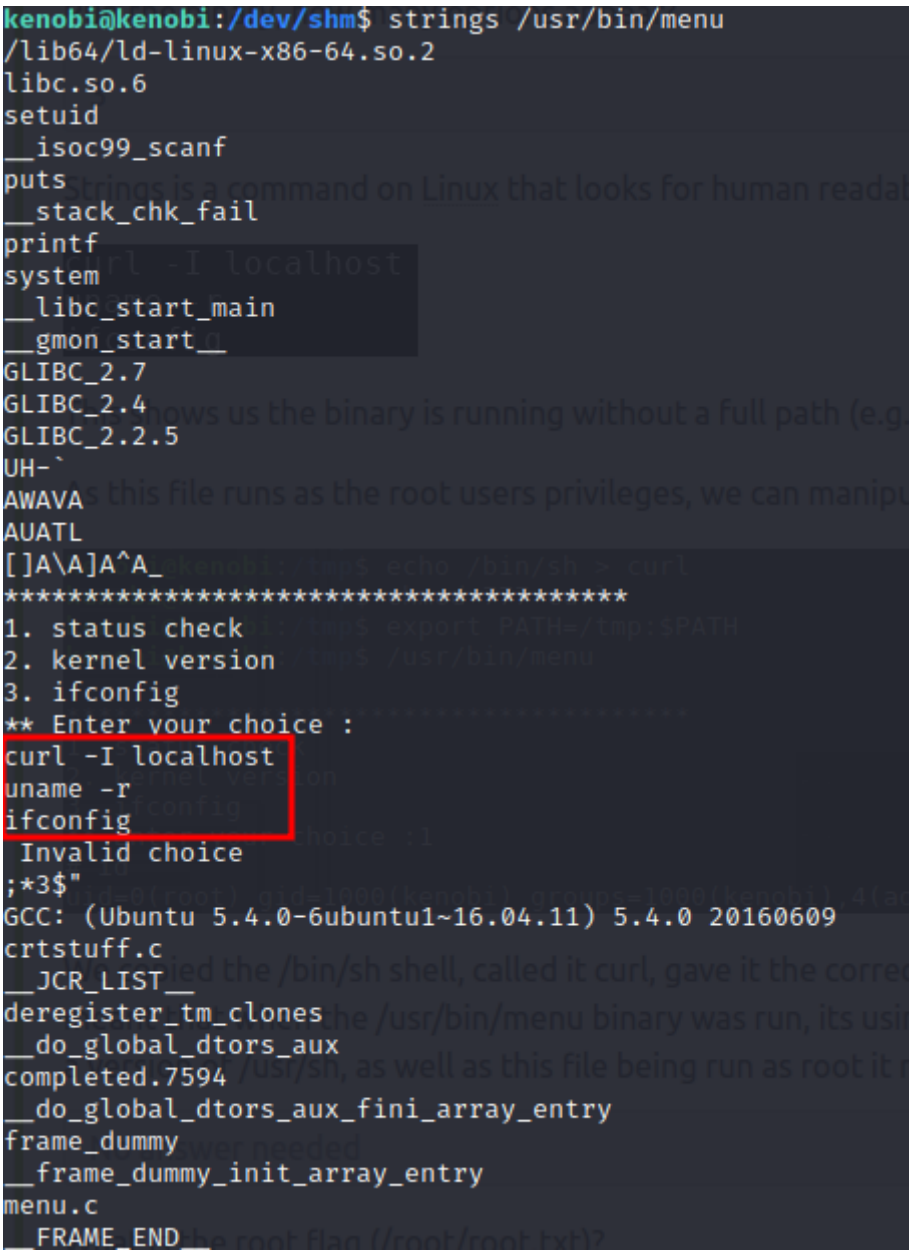


Figure 3.11: privesc

- we can see that the binary is running without a full path so we can abuse this

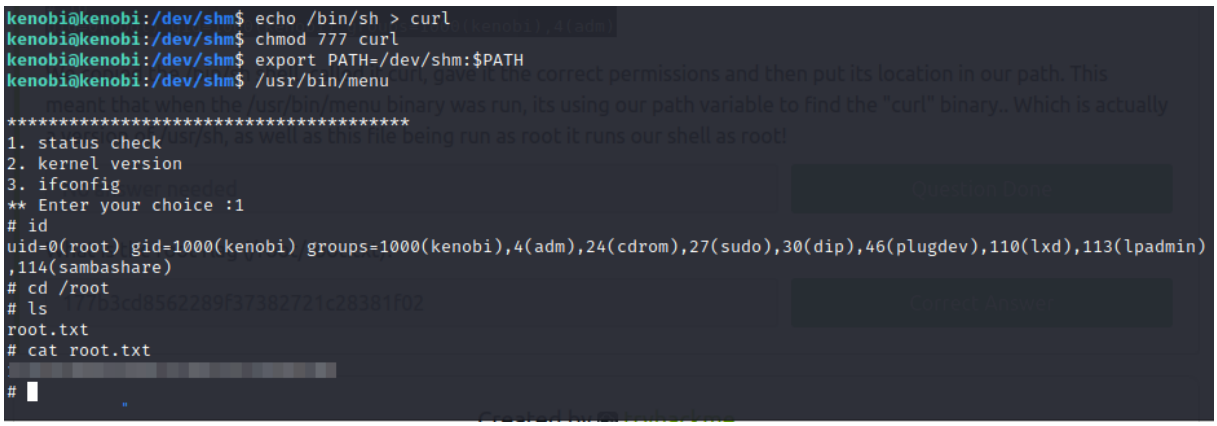


Figure 3.12: privesc

Vulnerability Exploited:

Vulnerability Explanation:

Vulnerability Fix:

Severity:

Exploit Code:

Proof Screenshot Here:

Proof.txt Contents:

3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

3.4 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which

can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the exam network was completed, I removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4 Additional Items

4.1 Appendix - Proof and Local Contents:

4.2 Appendix - Metasploit/Meterpreter Usage

4.3 Appendix - Completed Buffer Overflow Code