Introductory Notes on Metasploit

Below is general information about Metasploit that may be helpful to you:

- Metasploit is one of many tools that can be used for tunneling and other pen testing capabilities.
- Metasploit is written in a common language, Ruby.
- Metasploit comes built-in with many different types of exploits.
- Metasploit also includes a database capability; the database can also accept external input from various devices so information gathered can be read and used by Metasploit.

Metasploit Architecture

Metasploit is organized by directories. The main directories are defined below.

LIBRARIES	The libraries are Rex, MSF Core, and MSF Base. Rex is the custom ruby library that handles most of the low level tasks such as socket and protocol manipulation and text transformations. The MSF Core library is the basic API which handles the interactions between the various modules. The MSF Base library provides implementation of some default sessions, and is also a wrapper for some framework core functions that make the various tasks easier to manage.
INTERFACES	The interfaces are how the user interacts with the framework. There are several that can be used. In this course, we will focus our use on the CLI (command line interface) and Console.
TOOLS	This directory contains various Metasploit tools.
PLUGINS	The plugins directory contains programs or scripts that can be loaded at run-time. Some of these plugins are useful to interface with other programs such as Nessus, Nexpose, or an external database.
DATA	The data directory is unofficially a catch-all directory for files or programs that don't seem to fit in the other categories and include 3rd party developed tools. According to Metasploit, this directory contains editable files.
DOCUMENTATION	The documentation directory contains documentation for the framework.
SCRIPTS	The scripts directory contains Metasploit and externally developed scripts for Meterpreter and PowerShell, among other capabilities.
MODULES	The modules directory contains the actual Metasploit modules.

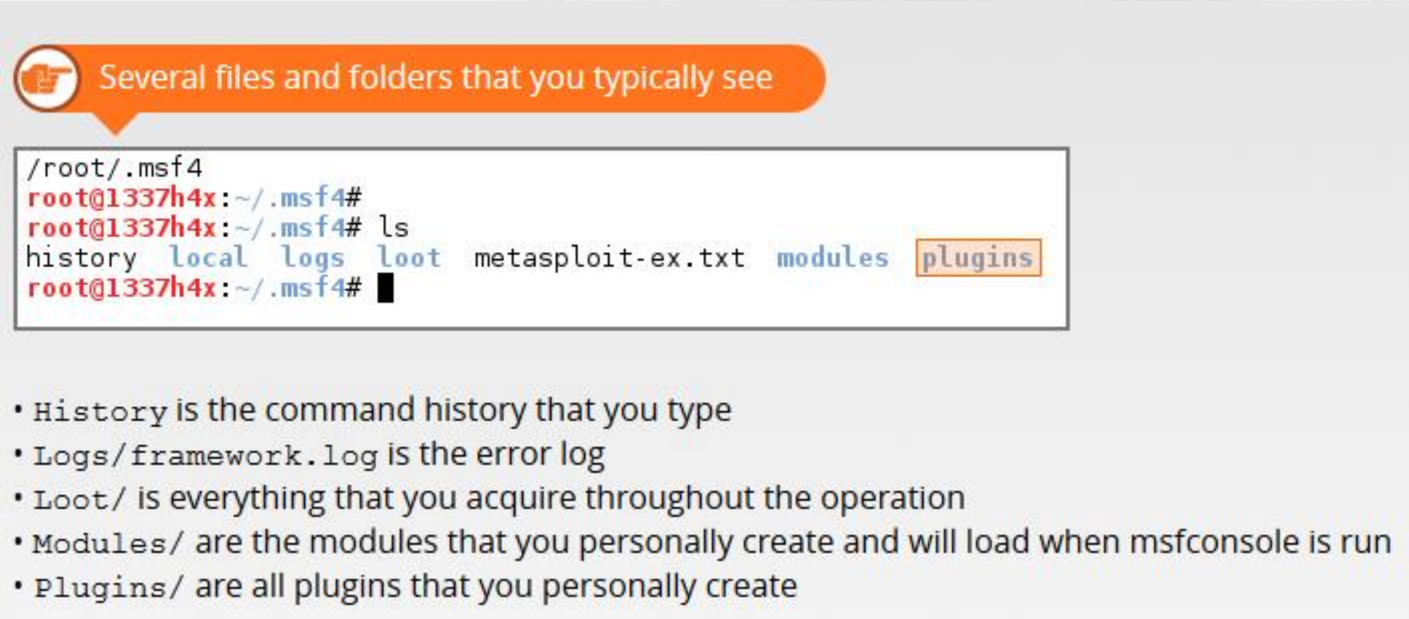
Metasploit Architecture: Modules

Below are types of modules:

PAYLOADS	These modules contain the backdoors for Metasploit. The three payload subdirectories are singles, stagers, and stages. Singles are stand-alone in that they consist of a single file. Singles are usually limited in functionality such as bind_tcp shell or adduser. The stagers and stages consist of multiple files, and the payloads provide more functionality such as Meterpreter. The payloads are sent or delivered in parts, where the stager is the smaller initial piece that gets sent to the target to open a communications channel to upload and install the stage.
EXPLOITS	This module contains the actual exploits for the framework.
ENCODERS	These modules prepare the payload for upload and execution on the target. They may remove specific bad characters that would be detected, modify the payload to avoid detection by an intrusion detection system (IDS) or antivirus, or convert the payload to a format or language that the target's architecture can understand.
NOPS	This module contains pads to be placed in the various exploits or payloads so the payload sizes are consistent, allowing the correct memory jump locations to be utilized.
AUXILARY	This directory contains non-payload exploits and other useful modules such as scanners, fuzzers, and enumeration scripts.
POST	A new module directory is the Post directory. This directory contains scripts that can be used post-exploitation. These include scripts to gather intel, escalate privileges, perform internal network reconnaissance, and manage compromised targets.

Directories

Metasploit saves everything to /root/.msf4. There are several files and folders in .msf4 that you typically see.



Several files and folders that you typically see

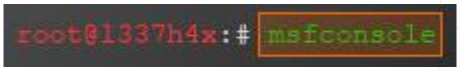
```
/root/.msf4
root@1337h4x:~/.msf4#
root@1337h4x:~/.msf4# ls
history  local  logs  loot  metasploit-ex.txt  modules  plugins
root@1337h4x:~/.msf4# █
```

- `History` is the command history that you type
- `Logs/framework.log` is the error log
- `Loot/` is everything that you acquire throughout the operation
- `Modules/` are the modules that you personally create and will load when msfconsole is run
- `Plugins/` are all plugins that you personally create

Metasploit Console

Msfconsole (MSF) is the most popular interface for the Metasploit Framework.

To launch msfconsole, just run msfconsole in the command line.



```
root@1337h4x:# msfconsole
```

Note: msfconsole has the capability of executing local commands, such as `ping` and `ifconfig`.

Commands

Here are some of the core commands we can use on the meterpreter:

help	The help command can be used with any command by prepending help to the command.
sessions	The sessions command allows you to list, interact with, and kill spawned sessions.
tab completion	Pressing the tab key will display all options that are available. If there is only one option available, it will auto-complete your string.
search	Metasploit allows you to search through several directories using specific or general terms.
info	The info command can be used to find detailed information on that module or script.
use	When you decide which module to use, set it up by invoking the use command and the path to the module.
show	Most modules have options that need to be set before you run them. The show options command shows these options.
set	Enter the command set payload and the type of payload. The payload also has additional options that need to be set.
exploit/run	Issue the exploit command when you are ready to run the module.

Meterpreter

Meterpreter, short for Meta-Interpreter, is an advanced payload that is included in the Metasploit Framework. It is a shell-style environment, containing core commands, as well as plugins. Extensions are loaded after exploitation, as needed. It lives only in computer memory and uses Transport Layer Security (TLS) encryption.

Meterpreter has quite a few built-in commands, listed below.

Core Commands	
?	help menu
background	moves the current session to the background
bgkill	kills a background meterpreter script
bglist	provides a list of all running background scripts
bgrun	runs a script as a background thread
channel	displays active channels
close	closes a channel
help	help menu
interact	interacts with a channel
irb	go into Ruby scripting mode
migrate	moves the active process to a designated PID
quit	terminates the meterpreter session
read	reads the data from a channel
run	executes the meterpreter script designated after it
use	loads a meterpreter extension
write	writes data to a channel

File System Commands	
cat	reads and output to stdout the contents of a file
cd	changes directory on the victim
del	deletes a file on the victim
download	downloads a file from the victim system to the attacker system
edit	edits a file with vim
getlwd	prints the local directory
getwd	prints working directory
lcd	changes local directory
lpwd	prints local directory
ls	lists files in current directory
mkdir	makes a directory on the victim system
pwd	prints working directory
rm	deletes a file
rmdir	removes directory on the victim system
upload	uploads a file from the attacker system to the victim

Networking Commands	
ipconfig	displays network interfaces with key information including IP address, etc.
portfwd	forwards a port on the victim system to a remote service
route	deletes a file on the victim view or modify the victim routing table

System Commands	
clearav	clears the event logs on the victim's computer
drop_token	drops a stolen token
execute	executes a command
getpid	gets the current process ID (PID)
getprivs	gets as many privileges as possible
getuid	gets the user that the server is running as
kill	terminates the process designated by the PID
ps	lists running processes
reboot	reboots the victim computer
reg	interacts with the victim's registry
rev2self	calls RevertToSelf() on the victim machine
shell	opens a command shell on the victim machine
shutdown	shuts down the victim's computer
steal_token	attempts to steal the token of a specified (PID) process
sysinfo	gets the details about the victim computer such as OS and name

User Interface Commands	
enumdesktops	lists all accessible desktops
getdesktop	gets the current meterpreter desktop
idletime	checks to see how long since the victim system has been idle
keyscan_dump	dumps the contents of the software keylogger

keyscan_start	starts the software keylogger when associated with a process such as Word or browser
keyscan_stop	stops the software keylogger
screenshot	grabs a screenshot of the meterpreter desktop
set_desktop	changes the meterpreter desktop
uictl	enables control of some of the user interface components

Privilege Commands	
getsystem	uses 15 built-in methods to gain sysadmin privileges

Password Dump Commands	
hashdump	grabs the hashes in the password (SAM) file

Timestomp Commands	
timestomp	manipulates the modify, access, and create attributes of a file

Helpful Hints for Using Meterpreter

- Changing directories in Meterpreter is slightly different; you must use two slashes to change the directory. You also have the option to use a Linux-style forward slash to specify the file and directory paths.
- The two main locations where scripts are stored are /usr/share/metasploit-framework/modules/post and /usr/share/metasploit-framework/scripts/meterpreter.
- If you need to drop out of Meterpreter and use the system shell, enter the shell command. You will be spawned in the current path, under the current user.
- Commands are run in the context of the target, and executables must be on the target to run.
- uploadexec is a Metasploit script that allows you to upload a program, run it, and then have it automatically removed from the box.
- A few options we want to use when we upload our file are `-e`, `-r`, and `-v`.
    - `-e` - the file will upload and execute on the local path
    - `-r` - remove the file after execution
    - `-v` - brings back the output from stdout
- You should have a second shell for your connection, because you will have your shell lockup and crash from time to time.

Recommended Internet Sites

- The Metasploit Directory: http://web.archive.org/web/20160802171011/http://www.hak5.org/episodes/metasploit-minute/the-metasploit-directory-structure-metasploit-minute

- Offensive Security Blog V2.0: http://web.archive.org/web/20160802171145/http://www.r00tsec.com/2014/02/metasploit-directory-structure.html
- Msfconsole (Metasploit Unleashed): http://web.archive.org/web/20160802171230/https://www.offensive-security.com/metasploit-unleashed/Msfconsole/
- Metasploit Framework Console Output Spooling (RAPID7Community): http://web.archive.org/web/20160802171531/https://en.wikipedia.org/wiki/Unix_signal
- Capture All Metasploit Input/Output (c0decstuff): http://web.archive.org/web/20160802171631/http://c0decstuff.blogspot.com/2011/07/capture-all-metasploit-inputoutput.html
- Metasploit 4.5.0-dev.15713 Cheat Sheet: http://web.archive.org/web/20160802171734/https://www.cheatography.com/huntereight/cheat-sheets/metasploit-4-5-0-dev-15713/
- Msfconsole Commands (Metasploit Unleashed): http://web.archive.org/web/20160802171849/https://www.offensive-security.com/metasploit-unleashed/msfconsole-commands/
- Metasploit/MeterpreterClient (Wikibooks): http://web.archive.org/web/20160802171933/https://en.wikibooks.org/wiki/Metasploit/MeterpreterClient

Please contact the Course Coordinators if you are unable to access any of the Recommended Internet Sites.