Metasploit: Section 1 Transcript

Introduction

Welcome to the Metasploit module.

During this module you will learn the basics of Metasploit, so that you can use this framework to perform your exploit and survey operations.

Throughout this module, you'll be presented with opportunities to assess and apply what you've learned.

At the end of this module, you will be able to:

- Use the Metasploit framework to perform exploitation operations.

This module does not contain a Bypass Exam. Click the Next Section button to proceed to the course material.

Metasploit: Section 2 Transcript

Introduction to the Metasploit Framework

1/19

Before we begin to discuss the Metasploit framework, it is important to note that Metasploit is one of many tools that can be used for tunneling and other pen testing capabilities. In this module we will explain concepts using Metasploit, but you can achieve many of the same results using various other similar tools.

Penetration testing tools are written in several different programming languages and most are designed to be standalone. As a result, these tools can be complicated to set up and in many cases the tools do not necessarily work well with each other.

On the other hand, Metasploit is written in a common language, Ruby, which is open source and free so that anyone with a desire can modify or add to the framework. Metasploit is designed to leverage multiple capabilities that are useful for penetration testing into a single common framework. This common framework allows the varied tools or techniques to be combined into a single complimentary package so the penetration tester is not required to know several different programming languages, multiple command syntaxes, or modify various payloads to work with the specific product.

Introduction to the Metasploit Framework

2/19

Metasploit comes built-in with many different types of exploits. From classic remote buffer overflows to the more complicated client-side exploits, where the tester can create custom malicious Javascripts, embed those scripts into a .pdf, and set up a web server within the Metasploit framework that will serve those files up to unwitting users through their browsers.

Metasploit also includes a database capability so that all the activities conducted within the framework such as target enumeration, port scanning, successful exploits, and host credentials are collected and cataloged for future retrieval and use. The database can also accept external input from various devices such as Nessus and Nmap so information gathered by those tools can be read and used by Metasploit.

Metasploit Architecture

3/19

At a basic level, Metasploit is organized by directories. The main directories are Libraries, Interfaces, Tools, Plugins, Data, Documentation, Scripts, and Modules. Click each main directory to learn about them.

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 allow the user to interact with the framework. There are several that can be used. In this course, we will focus on the CLI (command-line interface) and Console.

Tools: The tools 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 third-party 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

Let's learn a little bit more about the Modules directory. Click each module to learn more.

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.

Aux (Auxiliary): 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 to gather intel, escalate privileges, perform internal network reconnaissance, and manage compromised targets.

Directories 5/19

When using Metasploit, it saves everything to `/root/.msf4`. There are several files and folders in .msf4 that you typically see.

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

Next, let's change the directory to metasploit-framework by entering this: `cd /usr/share/metasploit-framework/`. After changing the directory, use the `ls` command to list the files and folders in the Metasploit directory.

Metasploit Console 6/19

Msfconsole (MSF) is the most popular interface for the Metasploit Framework. It does a great job at giving you access to all the Metasploit Framework available options and features. To launch msfconsole, just run `msfconsole` in the command line.

When you log into Metasploit through msfconsole, the first thing you need to do is use the `spool` command. The `spool` command saves all Metasploit output to a file. In this case, we have named the file with the date of the output.

Before we take a look at msfconsole commands, it's worth noting that msfconsole can execute local commands, such as `ping` and `ifconfig`.

Commands: Help, Sessions, and Tab Completion 7/19

Now that you have Metasploit open and ready for business it may be helpful to look at what the core commands are by entering `help` or `?` in the command prompt. This lists available commands and a brief description of each one.

The `help` command can be used with any command by prepending help to the command. For example, by entering `help sessions`, you can see what the options are for the `sessions` command. The `sessions` command allows you to list, interact with, and kill spawned sessions.

Another great feature of msfconsole is tab completion. Because there are so many modules available, sometimes it is difficult to recall the path and name of the module you want to run. Pressing the tab key will display all options that are available. If there is only one option available, it will auto-complete your string.

Commands: Search and Info 8/19

Metasploit lets you search through several directories using specific or general terms. In this example, we used search -h to see what kind of keywords we should look for. And now that we know what search criteria are available, we can search for server-side Windows-based Microsoft exploits.

Once you find a module or script that you are interested in, you can use the `info` command to get detailed information on that module or script. In this example we are interested in learning more about the spoolss exploit.

Use, Show, and Set

9/19

When you decide which module to use, set it up by invoking the `use` command and the path to the module. You should now notice that the command prompt has changed to note the module is now active.

Most modules have options that need to be set before you run them. The show options command shows these options.

If you are using an exploit module, as we are in our example, one of the options you need to set is the type of payload you want to use. Enter the command set payload and the type of payload, in our case it is reverse tcp. The payload also has additional options that need to be set. Let's do that next.

Set Options and Variables

10/19

As you can see, the exploit module requires the RHOST option to be set. The payload requires the LHOST to be set. The other module and payload options are already set, but you can change them if you desire.

The RHOST option determines where the packet goes next. Because our payload is a reverse_tcp backdoor, the LHOST will be the IP of the box that the target box will send the packet to. The LPORT is the port that needs to be listening on the box that will receive the packet from the target. The Metasploit default port is 4444. This port stands out fairly easily, so it's a good idea to change the number. We choose LPORT 80.

Setting the options is simple. You just use the set command followed by the option you want to set, and the variable for that option.

Now that you have your options configured, use the show options command to verify the options set. Once it looks good, you are ready to run the module.

Exploit/Run Commands

11/19

Issue the exploit command when you are ready to run the module. The listener started on our LPORT, and the exploit is testing remote target to determine the target version to use. In this example, it is determined the target is an XP Service Pack 3, English Language box. Next, it sends the correct version of the exploit that should work against that target system.

When you see the sending stage line, it means the exploit has been triggered and the payload is being uploaded for execution. The "Meterpreter session opened" line means your backdoor has successfully executed and connected back to your listener.

This is now your interactive shell on the remote box. Remember to run ipconfig to make sure you are on the correct target.

Managing Sessions

12/19

You are not limited to only one session in Metasploit. You can have multiple sessions that are on the same target or are on different targets. The sessions are managed via the background and sessions commands.

In this example, we sent our session to the background. This returns us to the exploit prompt, allowing us to set up another exploit. Use the sessions -l command to list your available active sessions. Use the sessions -i command followed by the session number to interact with one of the active sessions.

Now that you are familiar with msfconsole, let's look at another important piece of software included in the Metasploit Framework, Meterpreter.

Meterpreter

13/19

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 uses stages to upload and inject into a victim process. This means no new processes are generated on the target and you have the permissions of the victim process; however, if that process is killed, you lose those permissions.

Meterpreter has quite a few built-in commands. For a list of these command, refer to the Resources for this module.

Meterpreter File and Directory Paths

14/19

Changing directories in Meterpreter is slightly different. For example, when entering cd c:\windows\temp, Meterpreter states that the operation failed. This is because you must use two slashes to change the directory. So, in order to change the directory to the c:\windows\temp directory, you must enter cd c:\\windows\\temp. The first slash is an escape character that tells the Ruby interpreter not to interpret the second slash as an escape character.

You also have the option to use a Linux-style forward slash to specify the file and directory paths.

Meterpreter Scripts and Plugins

15/19

Unfortunately, Meterpreter does not have every possible tool or command built in it. Luckily, there are quite a lot of scripts and plugins that can do a wide range of tasks. These scripts, when installed, can be difficult to find, because they are in several different locations. These are the two main locations where scripts are stored (/usr/share/metasploit-framework/modules/post and /usr/share/metasploit-framework/scripts/meterpreter).

Many of these scripts and plugins are user/community built. They are frequently updated, so you should expect bugs and issues to pop up occasionally.

Issue the run command and then the script name to run a script. Almost all scripts understand the -h command, which provides basic usage information.

System Shell

16/19

Meterpreter shells are not extremely stable; they break easily. Just remember it is not your fault if your shell locks up. 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. Now let's look at uploading and executing an executable on the target.

Upload and Execute

17/19

Metasploit does not have a plugin or script for everything you want to do, so you may need to still use third-party binaries to get the information you want; however, uploadexec is a Metasploit script that allows you to upload a program, run it, and then have it automatically removed from the box. By default, the uploadexec script uploads the binary or file to the system's %TEMP% directory. You can find where the temp directory is located by running the get_env script.

A few options we want to use when we upload our files are -e, -r, and -v.

- With -e, the file will upload and execute on the local path,
- -r will remove the file after execution, and
- -v brings back the output from stdout

For our example, let's upload and execute promiscdetect.exe. When you enter this command: (Run uploadexec -e /root/promiscdetect.exe -r -v), it uploads promiscdetect.exe, renames it to svhost78.exe, runs it, and then deletes it after completing its task.

Remember, it's extremely important to always confirm that the file was deleted.

Make a Backup!

18/19

The final thing you should take note of, is that you should have a second shell for your connection. This is important because you will have your shell lockup and crash from time to time. If you have a backup, you can continue with your operation. Using the 'duplicate' command in Meterpreter gives you the capability to stay on target by having an additional shell that you will need when something happens in your primary shell.

Exercise Introduction

19/19

It is time for an Exercise. In order to successfully complete the Exercise, you are expected to use your notes and any available resources presented throughout the course, in addition to conducting your own Internet research.

Metasploit: Section 3 Transcript

Summary

You have completed the Metasploit module.

In this module you learned the basics of Metasploit so that in the future you can use this framework to perform your exploit and survey operations.

You should now be able to:

- Use the Metasploit framework to perform exploitation operations.

To receive credit and advance to the next module, you must achieve a passing score on the Module Exam.

Click the Next Section button to begin the Module Exam.