Exploit Vulnerability

Introduction

Target Exploitation

Target exploitation is one area that sets a penetration test apart from a vulnerability assessment. Now that vulnerabilities have been found, you will actually validate and take advantage of these vulnerabilities, by exploiting the system, in the hope of gaining full control or additional information and visibility into the targeted network, and the systems therein.

Writing exploit code from scratch can be a time-consuming and expensive task. Thus, using publicly available exploits and adjusting them to fit your target environment may require expertise, which would assist in transforming the skeleton of one exploit into another, if the similarity and purpose is almost the same. i highly encourage the practice of publicly available exploits in your own labs to further understand and kick-start writing your own exploit code.

Vulnerability Research

Understanding the capabilities of a specific software or hardware product may provide a starting point for investigating vulnerabilities that could exist in that product. Conducting vulnerability research is not easy, neither is it a one-click task. Thus, it requires a strong knowledge base with different factors to carry out security analysis. The following are the factors to carry out security analysis:

Vulnerability Research

Understanding the capabilities of a specific software or hardware product may provide a starting point for investigating vulnerabilities that could exist in that product. Conducting vulnerability research is not easy, neither is it a one-click task. Thus, it requires a strong knowledge base with different factors to carry out security analysis. The following are the factors to carry out security analysis:

1. Programming Skills
2. Reserve Engineering
3. Instrumented Tools
4. Exploitability and Payload Conostruction

Introduction to Metasploit

Metaspoit Framework is an open source penetration tool used for developing and executing exploit code against a remote target machine. In simple words, Metasploit can be used to test the Vulnerability of computer systems in order to protect them and on the other hand it can also be used to break into remote systems.

Metasploit Terms

***VULNERABILITY** -A WEAKNESS WHICH ALLOWS AN ATTACKER TO BREAK INTO OR COMPROMISE A SYSTEM'S SECURITY.*

***EXPLOIT** – CODE WHICH ALLOWS AN ATTACKER TO TAKE ADVANTAGE OF A VULNERABILITY SYSTEM.*

***PAYLOAD-** ACTUAL CODE WHICH RUNS ON THE SYSTEM AFTER EXPLOITATION*

Metasploit Fundamentals

In learning **how to use Metasploit** you will find there are many different interfaces to use with this *hacking tool*, each with their own strengths and weaknesses. As such, there is no one perfect interface to use with the *Metasploit console*, although the ***MSFConsole*** is the only supported way to access most **Metasploit commands**. It is still beneficial, however, to be comfortable with all *Metasploit interfaces*.

What is Msfconsole

The **msfconsole** is probably the most popular interface to the Metasploit Framework (MSF). It provides an “all-in-one” centralized console and allows you efficient access to virtually all of the options available in the MSF. MSFconsole may seem intimidating at first, but once you learn the syntax of the commands you will learn to appreciate the power of utilizing this interface.

Benefits of Using Msfconsole

- It is the only supported way to access most of the features within Metasploit.
- Provides a console-based interface to the framework
- Contains the most features and is the most stable MSF interface
- Full readline support, tabbing, and command completion
- Execution of external commands in msfconsole is possible

Launching Msfconsole

The MSFconsole is launched by simply running **msfconsole** from the command line. MSFconsole is located in the

/usr/share/metasploit-framework/msfconsole directory.

The -q option removes the launch banner by starting **msfconsole** in quiet mode.

Using Msfconsole Commandline

You can pass **-h** to msfconsole to see the other usage options available to you

Entering **help** or a **?** once in the msf command prompt will display a listing of available commands along with a description of what they are used for.

The msfconsole is designed to be fast to use and one of the features that helps this goal is tab completion. With the wide array of modules available, it can be difficult to remember the exact name and path of the particular module you wish to make use of. As with most other shells, entering what you know and pressing 'Tab' will present you with a list of options available to you or auto-complete the string if there is only one option. Tab completion depends on the ruby readline extension and nearly every command in the console supports tab completion.

Using Msfconsole Tab Completion Options

- use exploit/windows/dce
- use .*netapi.*
- set LHOST
- show
- set TARGET
- set PAYLOAD windows/shell/
- exp

The msfconsole is the most commonly used interface for Metasploit. Making yourself familiar with these msfconsole commands will help you throughout this course and give you a strong foundation for working with Metasploit in general.

Using MSFvenom: Exploit Development (Payload)

msfvenom is a combination of *Msfpayload and Msfencode*, putting both of these tools into a single Framework instance. msfvenom replaced both msfpayload and msfencode as of June 8th, 2015.

The advantages of msfvenom are:

- One single tool
- Standardized command line options
- Increased speed

Msfvenom –h

Msfvenom has a wide range of options available:

MSFvenom: Commandline Usage

```
msfvenom -a x86 --platform Windows -p windows/shell/bind_tcp
-e x86/shikata_ga_nai -b '\x00' -i 3 -f python
```

The msfvenom command and resulting shellcode above generates a *Windows bind shell* with three iterations of the *shikata_ga_nai encoder* without any null bytes and in the python format.

MSFvenom: Platforms

Here is a list of available platforms one can enter when using the –platform switch

```
Cisco or cisco OSX or osx Solaris or solaris BSD or bsd
OpenBSD or openbsd hardware Firefox or firefox BSDi or bsdi
NetBSD or netbsd NodeJS or nodejs FreeBSD or freebsd Python
or python AIX or aix JavaScript or javascript HPUX or hpux
PHP or php Irix or irix Unix or unix Linux or linux Ruby or
ruby Java or java Android or android Netware or netware
Windows or windows mainframe multi
```