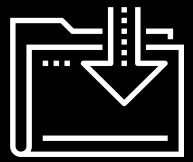
Exploring Exploitation

Cybersecurity Boot Camp

Lesson 16.3



Class Objectives

By the end of today's class, you will be able to:

Understand what command and control (C2) is and how it fits into a penetration tester's toolkit.

Use Metasploit to automate exploitation activities.

Explain what privilege escalation is and how it fits into the attack cycle.

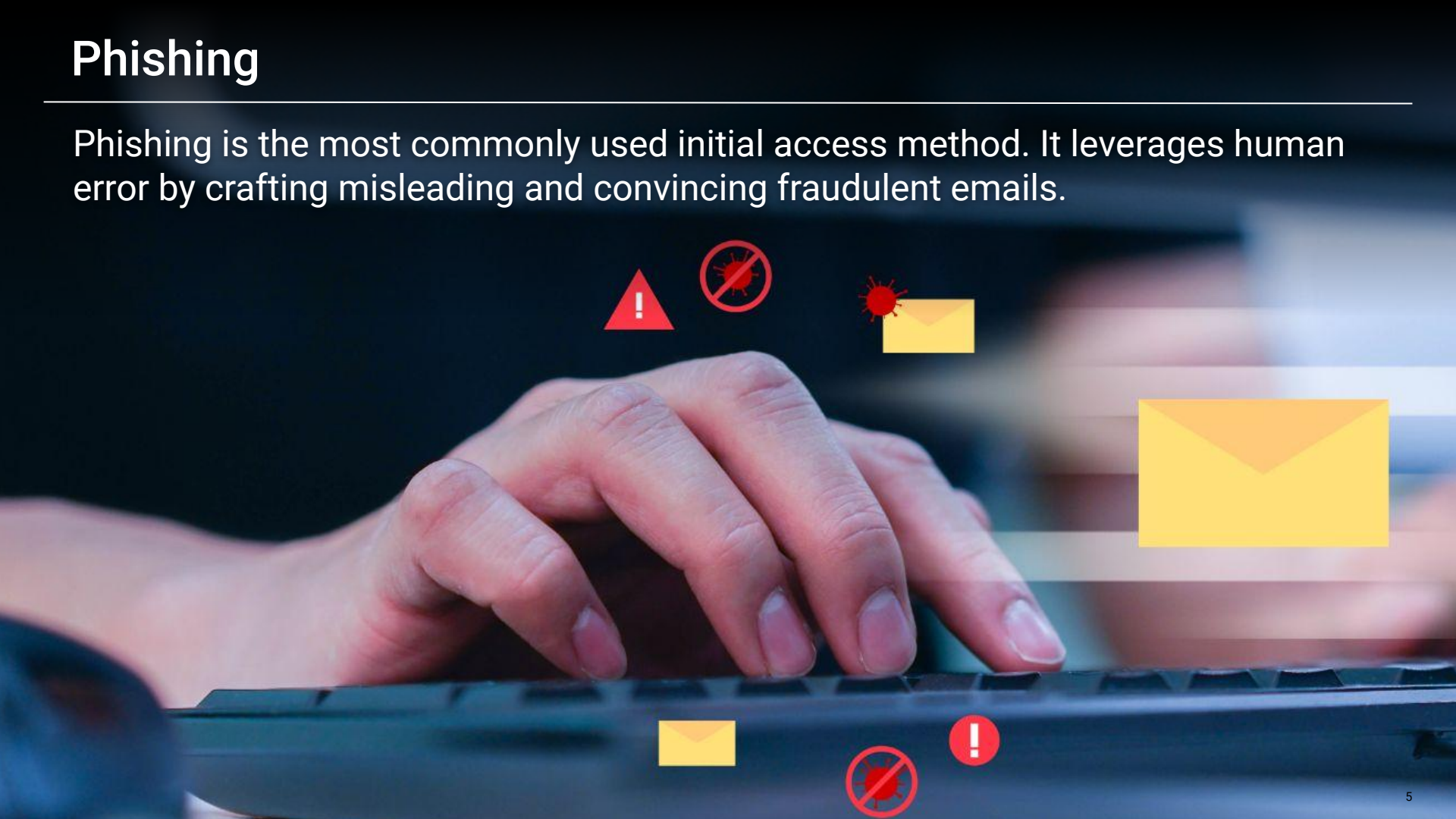
Perform basic privilege-escalation tasks.

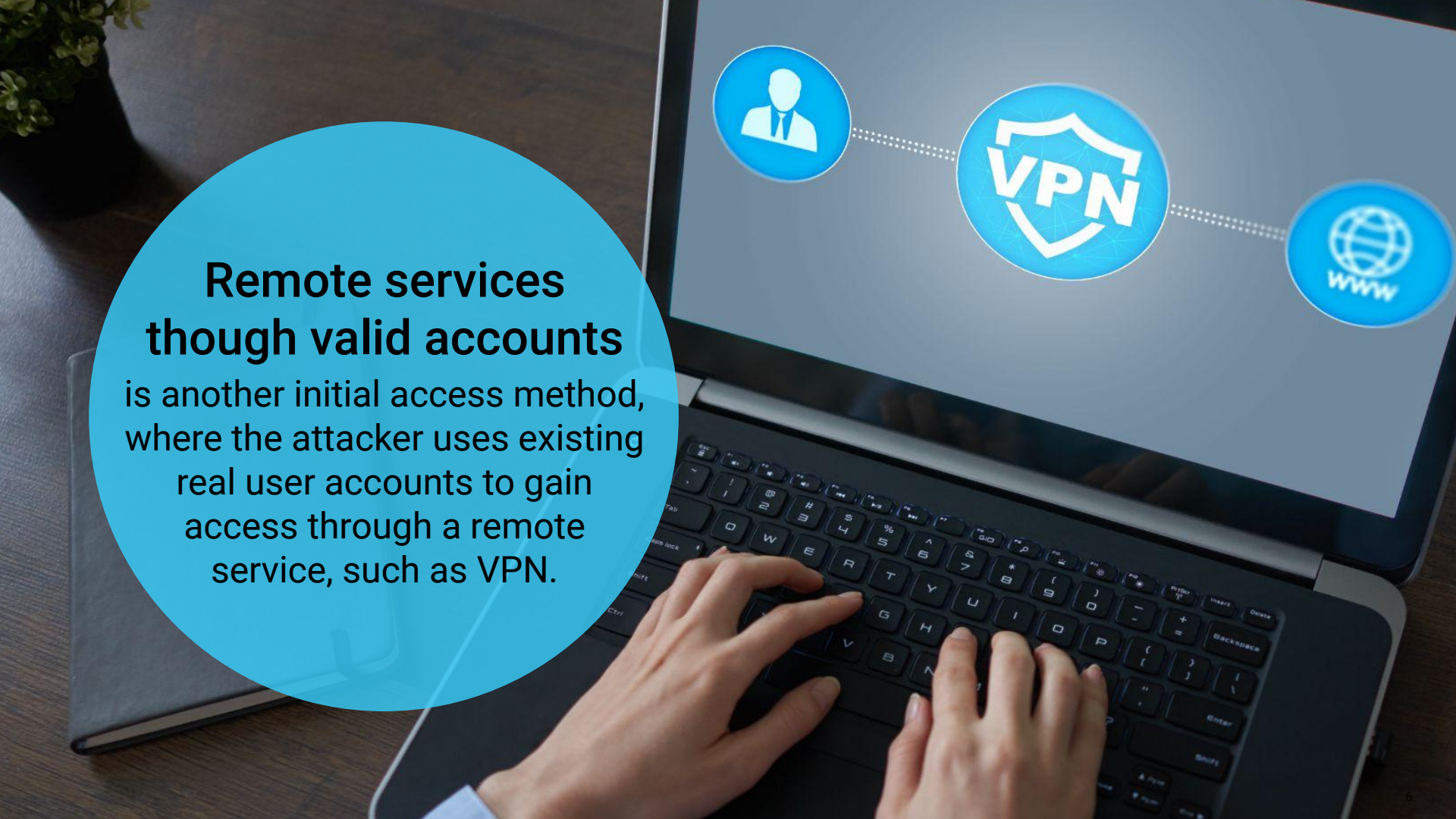

Day 2
Recap

Initial access is a MITRE tactic that covers methods to gain access into a target’s system.

Phishing

Phishing is the most commonly used initial access method. It leverages human error by crafting misleading and convincing fraudulent emails.



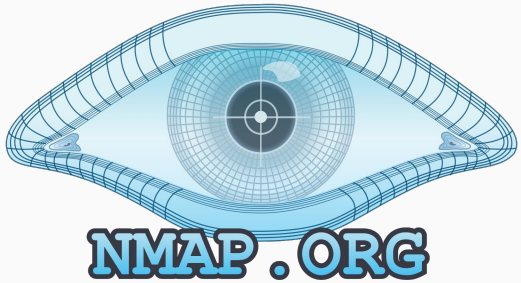


Remote services though valid accounts

is another initial access method, where the attacker uses existing real user accounts to gain access through a remote service, such as VPN.

Day 2 Recap

Scanning is the second phase after Reconnaissance. Scanning utilizes tools to gather information such as network information and potential vulnerabilities.



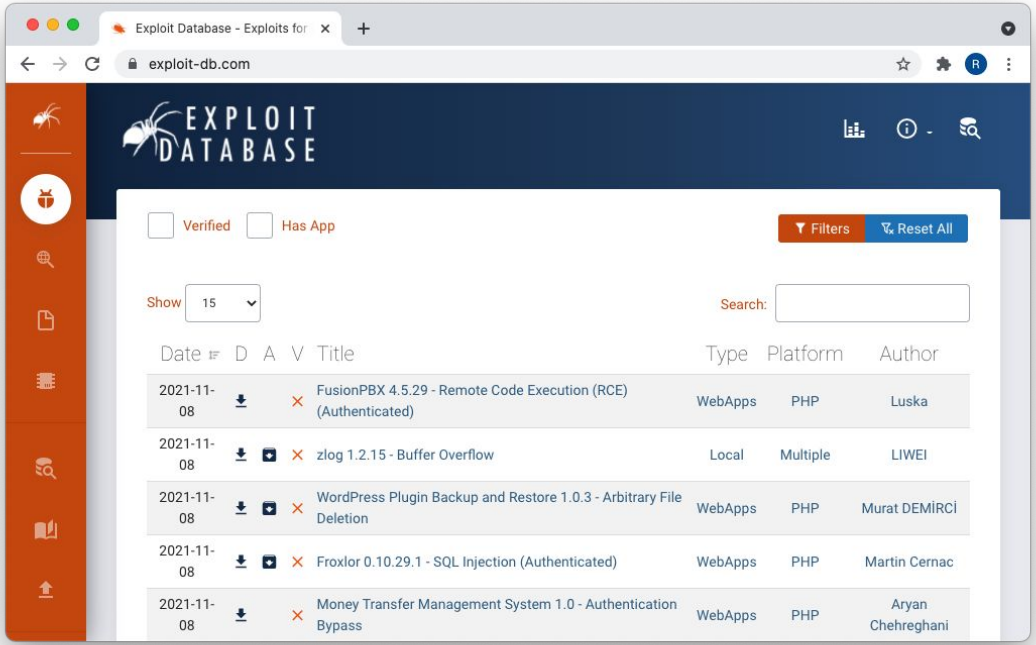
We often use the tools Nessus, Hping, Nmap, and Zenmap to conduct scanning.

NSE (Nmap scripting engine) scripts are scripts that work with Nmap or Zenmap and are commonly used to test whether a service is vulnerable to an exploit.

Day 2 Recap

Exploit-DB.com is a popular online database that contains publicly disclosed exploits, cataloged according to their Common Vulnerability and Exposure (CVE) identifiers.

SearchSploit is a command line utility for Exploit-DB that allows you to take an offline copy of the entire Exploit Database with you wherever you go.



Exploitation

Exploited access to the machine is typically granted in the form of terminal access, known as a **shell**. There are two types of shells:

Bind shell

The remote host opens a port for the current host to connect to.

The current, local host then connects to that remote host's port.

Reverse shell

The remote host connects back to a port on the local host.

The Five Stages of Pentesting Recap

On Day 1, we introduced the five stages of a pen testing engagement:

01A	**Planning**	Define the purpose and scope of the test, and sign all legal contracts.
01B	**Reconnaissance**	Obtain publicly available information about your target.
02	**Scanning**	Use tools to run a scan against your target to gather information, such as open ports, and run services to determine potential vulnerabilities.
03	**Exploitation**	Attack the vulnerabilities discovered in the previous steps in order to gain access to the target.
04	**Post Exploitation**	Gather valuable information from the compromised systems.
05	**Reporting**	Report on the previous five steps to provide a summary of actions taken, findings, and recommended mitigations.

We ended Day 2's lesson by starting the **Exploitation phase (Phase 3)** and manually exploiting a remote host's vulnerable service to obtain a reverse shell.

- Today, we will begin by covering command and control (C2) and Metasploit.
- Then, we'll use Metasploit to exploit another vulnerability and gain another shell on a remote host.

Reporting

Post Exploitation

Exploitation

Scanning

Planning and Reconnaissance

The day will conclude with a **privilege-escalation activity** to escalate from a low-privileged user to root.

Reporting

Post Exploitation

Exploitation

Scanning

Planning and Reconnaissance

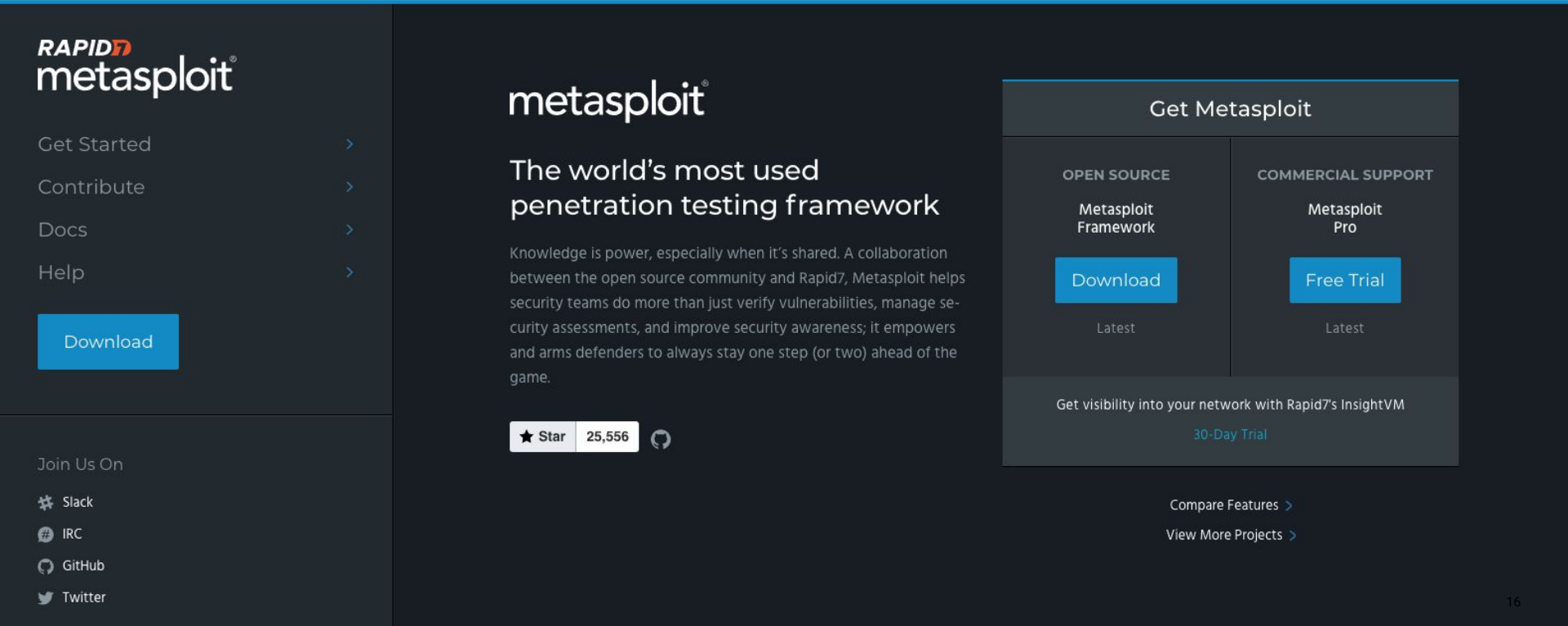
Command and Control

Command and control (C2) is a framework that consists of tools and techniques that attackers use to maintain communication with compromised devices following initial exploitation.

Command and Control

- C2 frameworks come in a variety of languages and are developed by individuals and companies alike.
- The majority of C2 frameworks are open source, meaning that the public can read the source code and they’re free. But a few are closed sourced and must be paid for.
- We will learn about a variety of C2 frameworks, but for the purposes of this class, we'll use the **Metasploit** framework.

Metasploit is a popular and open source C2 framework created by Rapid7.



C2 Architecture

C2 frameworks work within a C2 architecture, which consists of:

01 C2 server

The attacker's server, where the attacker communicates with the compromised machines.

02 C2's agents

The payload that is run on the compromised machine in order to open up a connection back to the C2 server.

C2 Architecture

A C2 architecture is how the C2 server is set up and how the agent communicates back to the C2 server.

In order to compromise a machine, the C2 agent must be running on it.

Fortunately, most C2 frameworks have a built-in capability to generate an agent.

The agent typically runs as a process and needs to run continuously in order for communication to flow consistently between it and the C2 server.

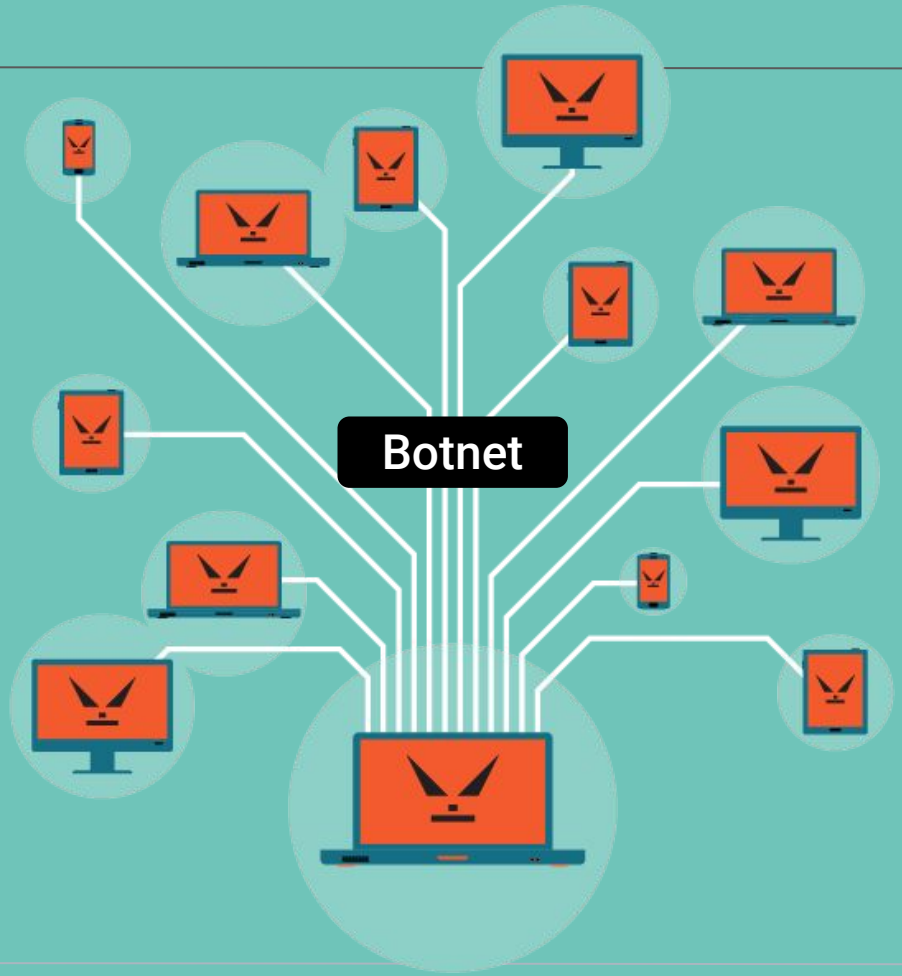
Metasploit’s agent is built into the code of its exploits.

Zombies and Botnets

Once the machine has the C2's agent running on it, it is compromised—sometimes referred to as a **zombie.**

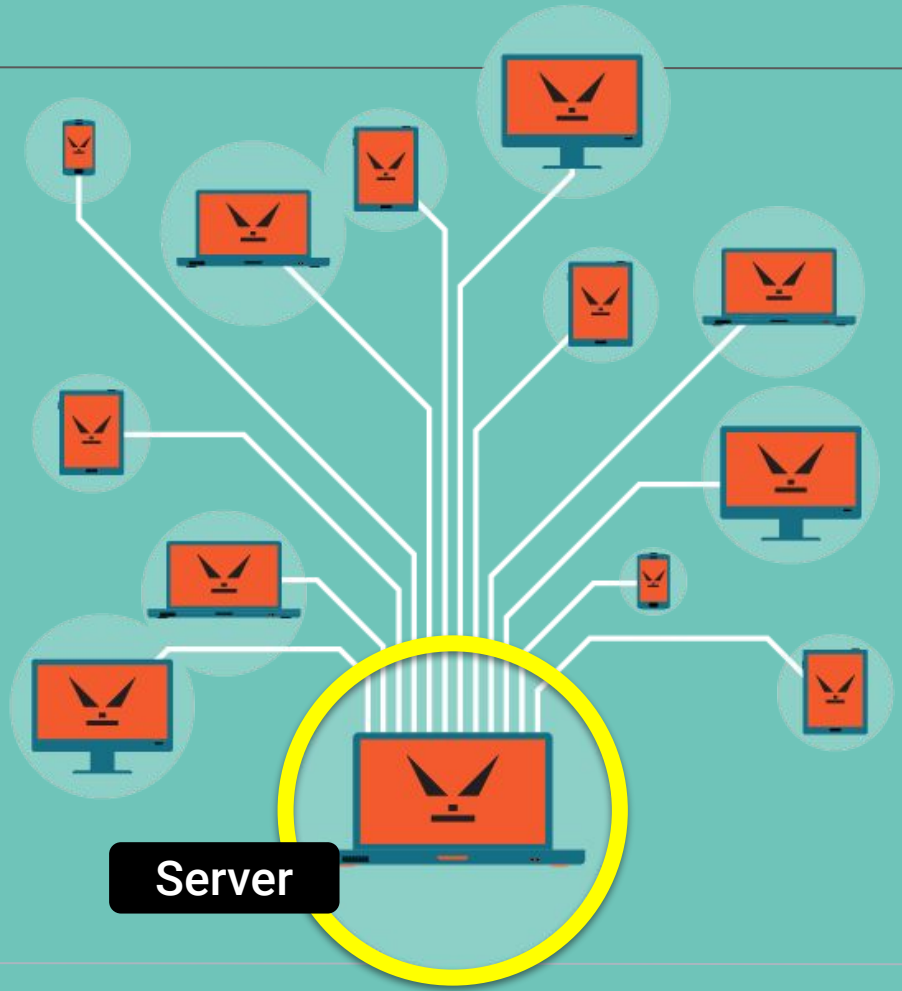
Zombies and Botnets

Multiple zombies form a **botnet**.



Zombies and Botnets

The entire botnet of compromised machines is controlled through the single C2 server.



Concealing the C2 Server

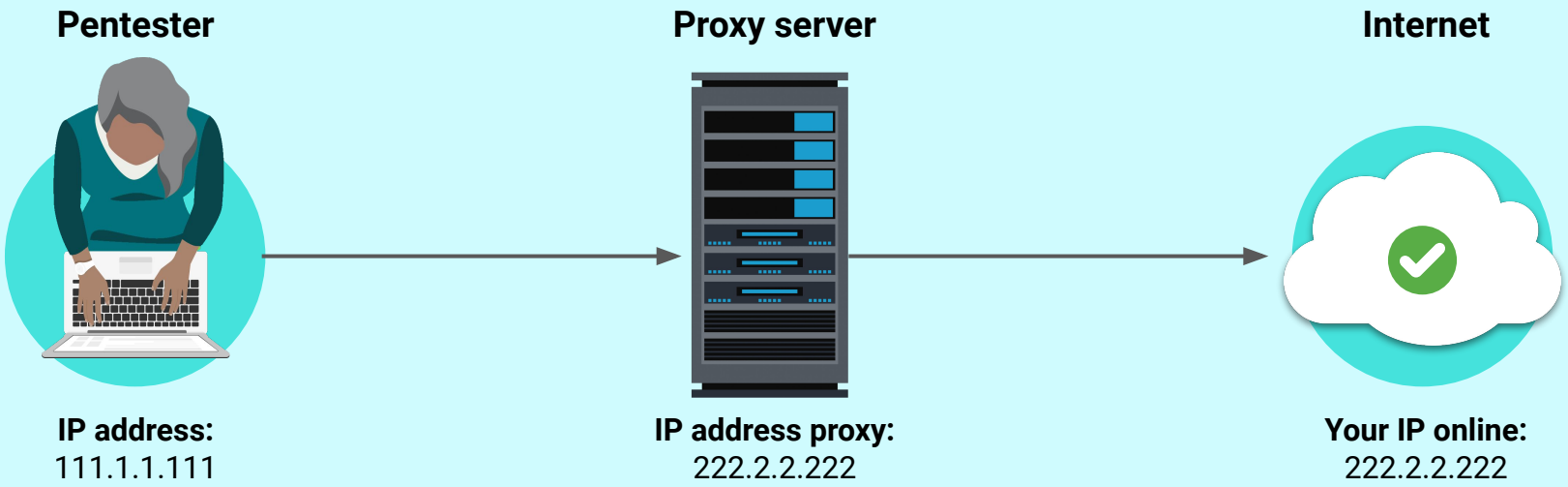
It is very important that a penetration tester conceal the C2 server in order to avoid detection.

Concealing the C2 Server

Method 1: Redirection Through Proxies

We will use a host that directly connects to the C2 server for simplicity. However, in a real penetration test, that would be a poor decision, because a simple packet collection would detect the C2 server IP, which could be blocked.

To avoid this detection, a pentester could have the C2 redirected through proxies.

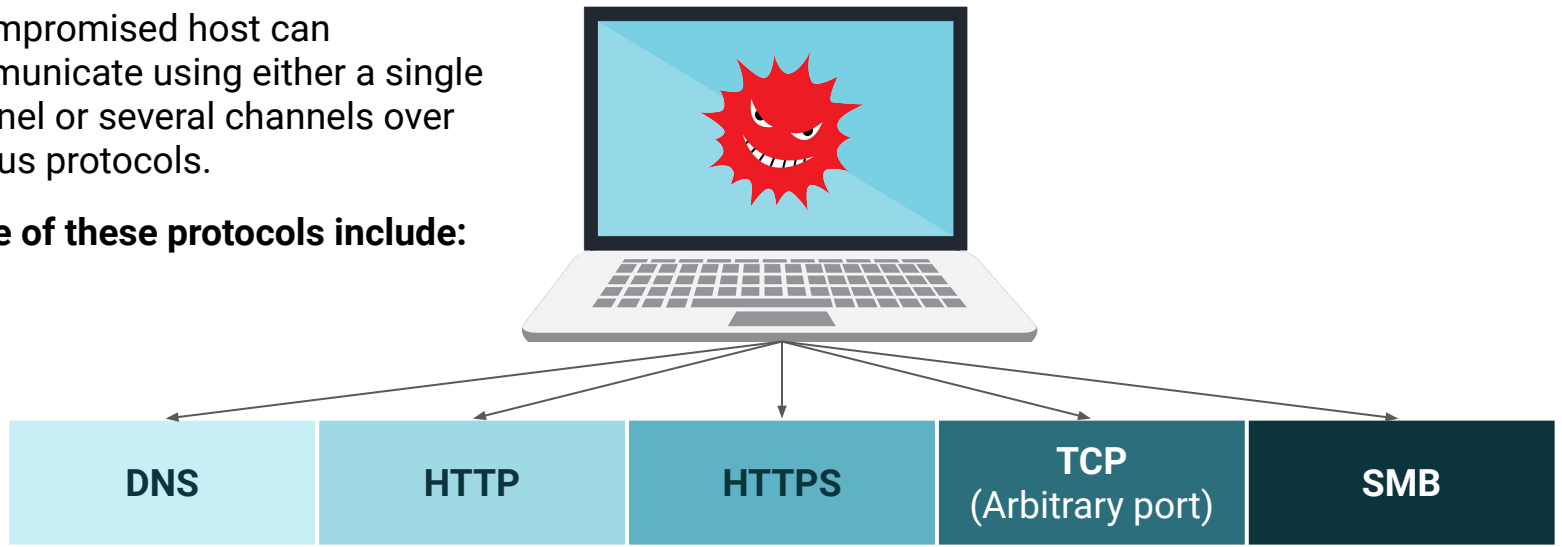


Method 2: Utilizing Multiple Channels

Another method that most C2 frameworks support is having the agent use multiple ways, or **channels**, to communicate back to the server.

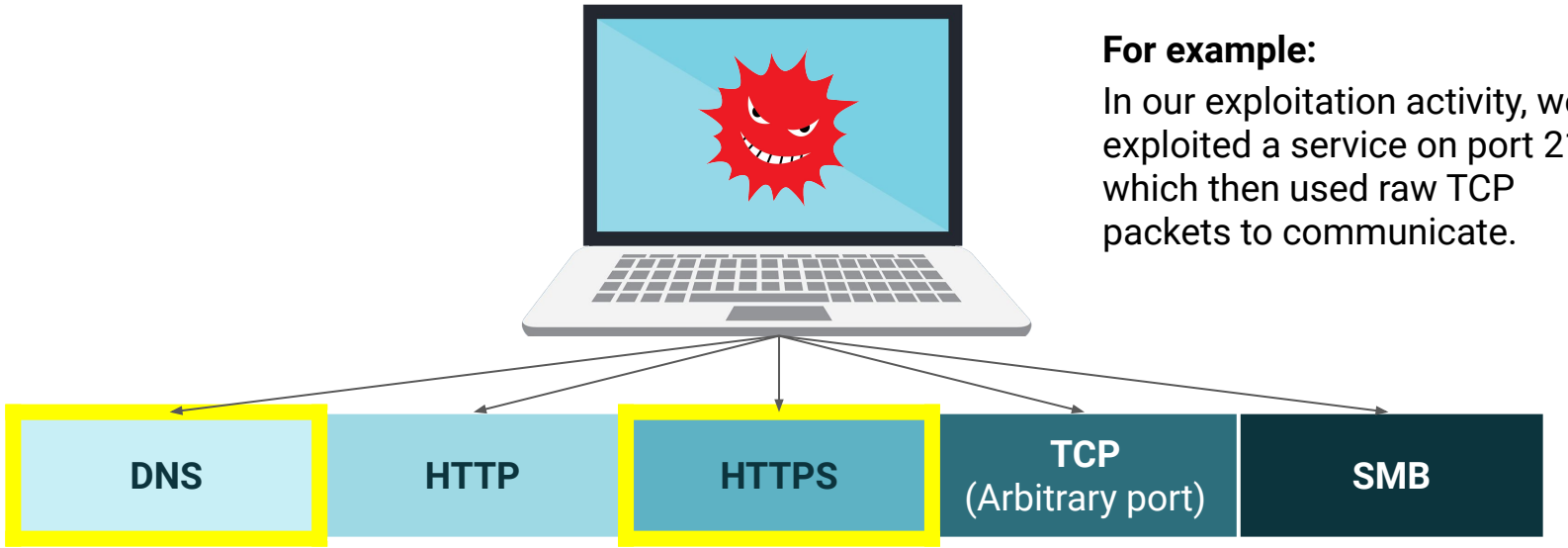
A compromised host can communicate using either a single channel or several channels over various protocols.

Some of these protocols include:



Method 2: Utilizing Multiple Channels

The protocol used is dependent on the target's network firewall egress rules. Most egress rules allow HTTP/S and DNS outbound, which is why those channels are commonly used for C2 communication.



For example:

In our exploitation activity, we exploited a service on port 21, which then used raw TCP packets to communicate.

While we will only use the **Metasploit** C2 framework in this class, most C2 frameworks are open source, which allows you to use them freely.

It’s good practice to have multiple C2 frameworks available for a penetration test or red team engagement.

Method 3: Utilizing Multiple C2 Frameworks

C2 frameworks each have their own “fingerprints” or “indicators of compromise.”

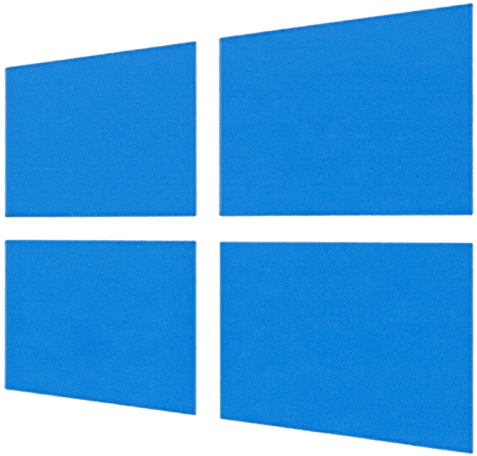
For example, Metasploit’s payload and agent generator, `msfvenom`, will generate a payload in a specified format (e.g. `.exe`).

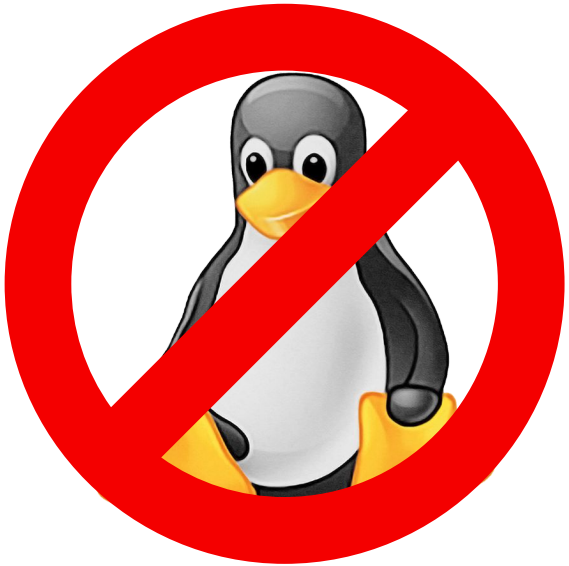
These payloads have well-known signatures, and many antivirus products will alert on them and kill the payload.

In an event where a certain AV product is killing the payload during or after execution, it may be a good idea to switch C2 frameworks.

Method 3: Utilizing Multiple C2 Frameworks

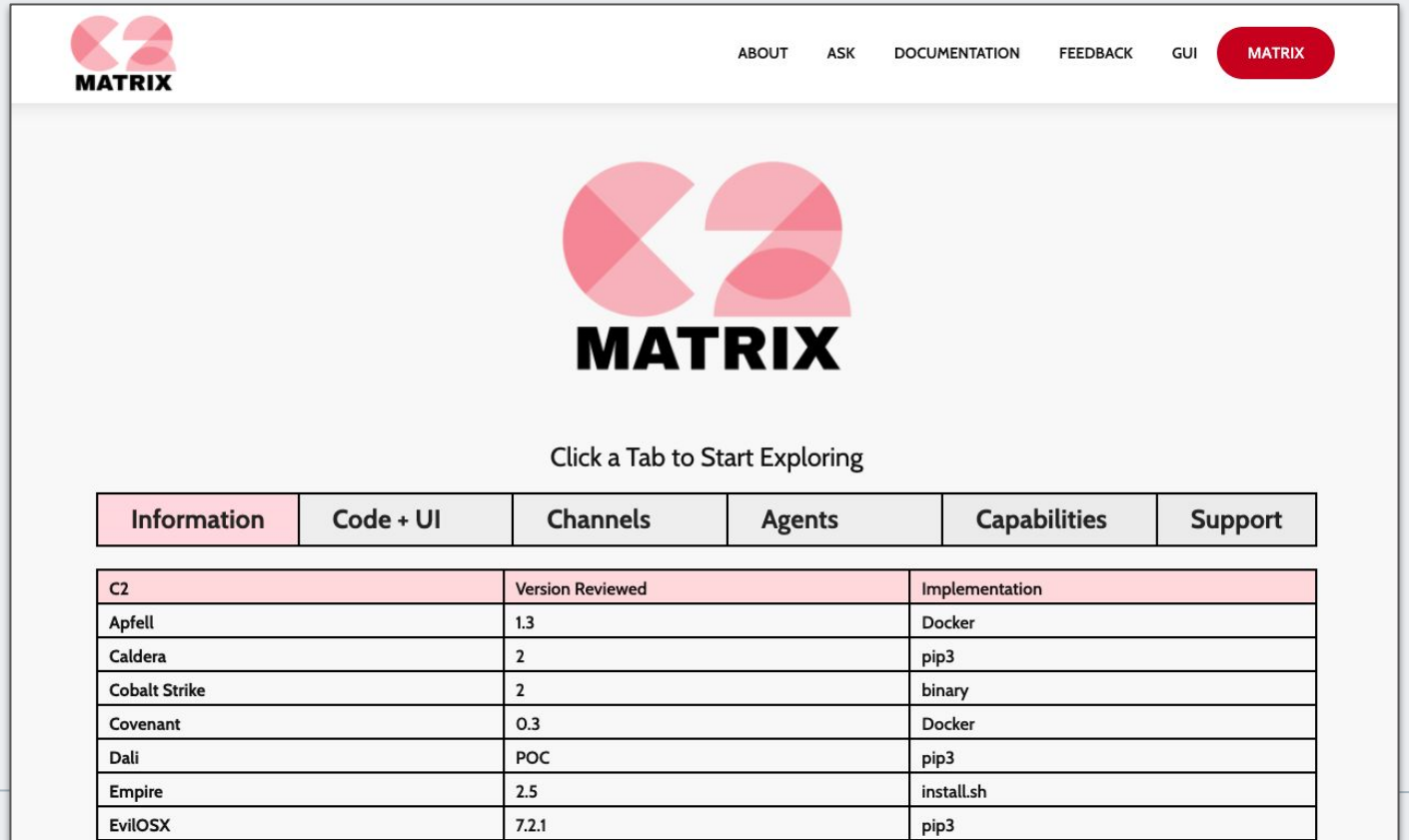
Not all C2 frameworks support every OS. Some only support Windows, not Linux, for example.





It’s good to have a secondary C2 framework that supports an OS if the primary C2 framework doesn’t support it.

To select the best C2 framework for a given project's requirements, a penetration tester could use the website C2 Matrix.



C2	Version Reviewed	Implementation
Apfell	1.3	Docker
Caldera	2	pip3
Cobalt Strike	2	binary
Covenant	0.3	Docker
Dali	POC	pip3
Empire	2.5	install.sh
EvilOSX	7.2.1	pip3

Instructor Demonstration

C2 Matrix

C2 Matrix Demo

For this demonstration, we've been tasked with selecting a C2 framework that meets the following requirements:

- It must support the channels of HTTP and SMB.
- It must support Windows.
- It must provide logging.

C2 Matrices

Information	Contains the version number reviewed.
Code + UI	Contains the server and agent language used.
Channels	Contains the supported channels.
Agents	Contains the supported OS.
Capabilities	Contains other available capabilities.
Support	Contains the additional support provided.

Summary

Can you define the following terms?

- C2
- C2 architecture
- C2 agent
- Zombie
- Botnet

Activity: C2 Research

In this activity, you will research various C2 frameworks and choose the one(s) most appropriate for your MegaCorpOne engagement.

Suggested Time:

15 Minutes

Time's Up! Let's Review.

Questions?

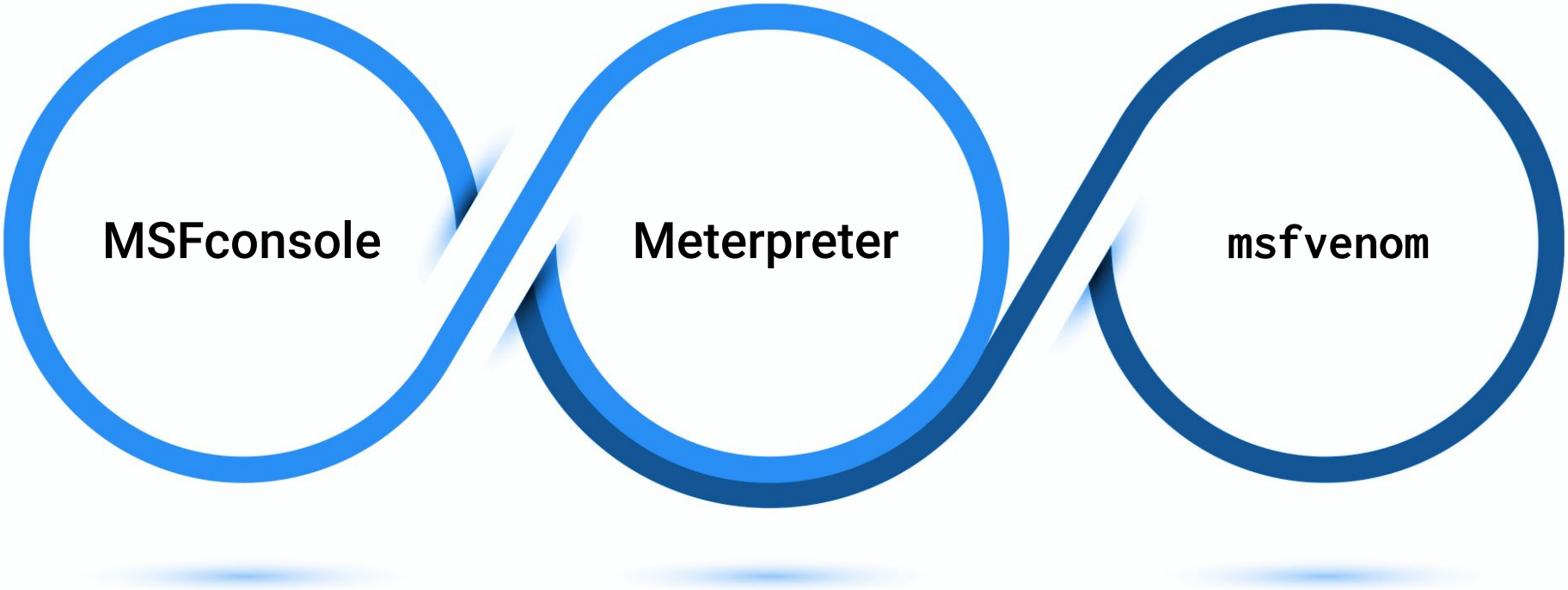
Metasploit

Metasploit

- Metasploit is a C2 framework that was created in 2003 and is owned by the company Rapid7.
- There's a free version, which comes installed on Kali by default, and a professional version, which expands on its exploitation and C2 capabilities.
- Metasploit contains a suite of tools for hacking servers and other networked devices.
- Metasploit also has tools for port and service scanning as well as enumeration.
- In addition to exploitation and scanning, Metasploit lets you save the results of scans to a database for easy review.

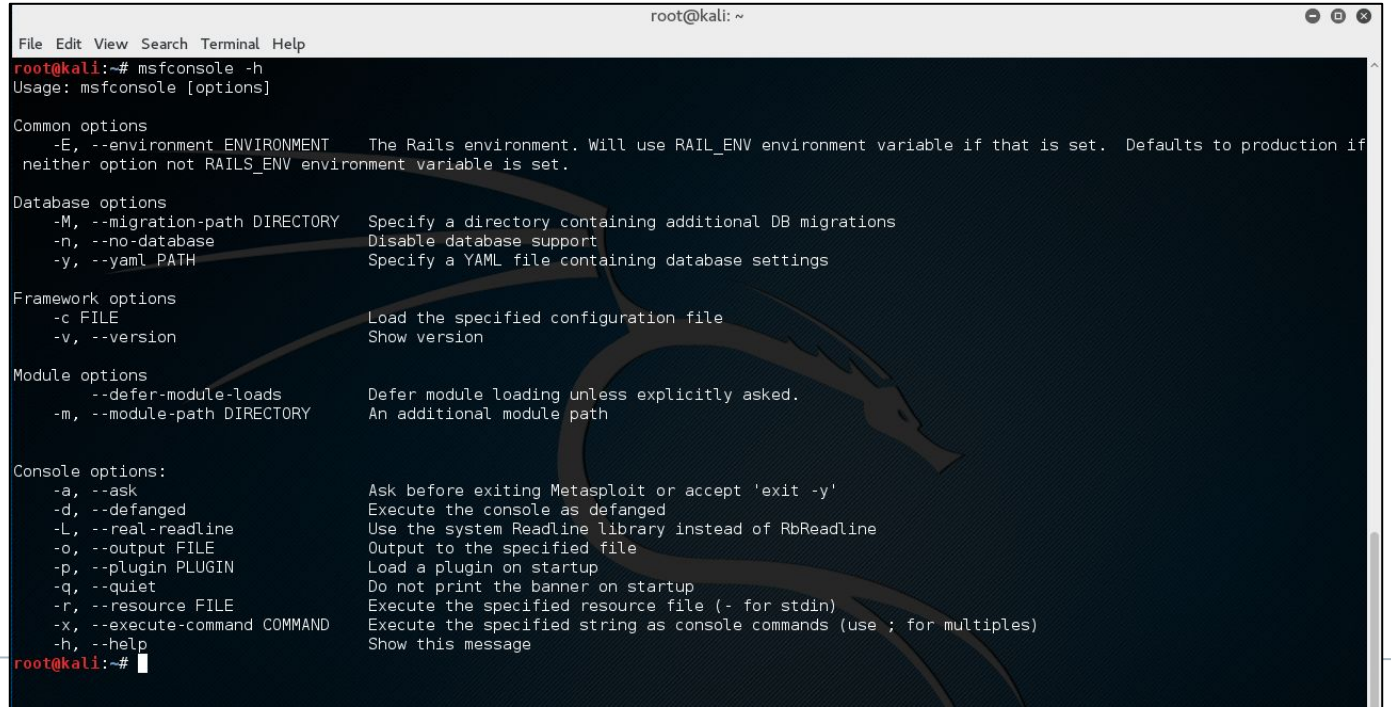
Metasploit

The main Metasploit tools that we'll focus on are:



MSFconsole

The main interface for Metasploit. Offers a centralized console to access all the options and modules. MSFconsole runs on your local machine, not on the machines you compromise.



Meterpreter

A Linux-style shell that Metasploit launches when you successfully break into a target machine. Unlike MSFconsole, Meterpreter runs on the machines you compromise, not on your local machine.

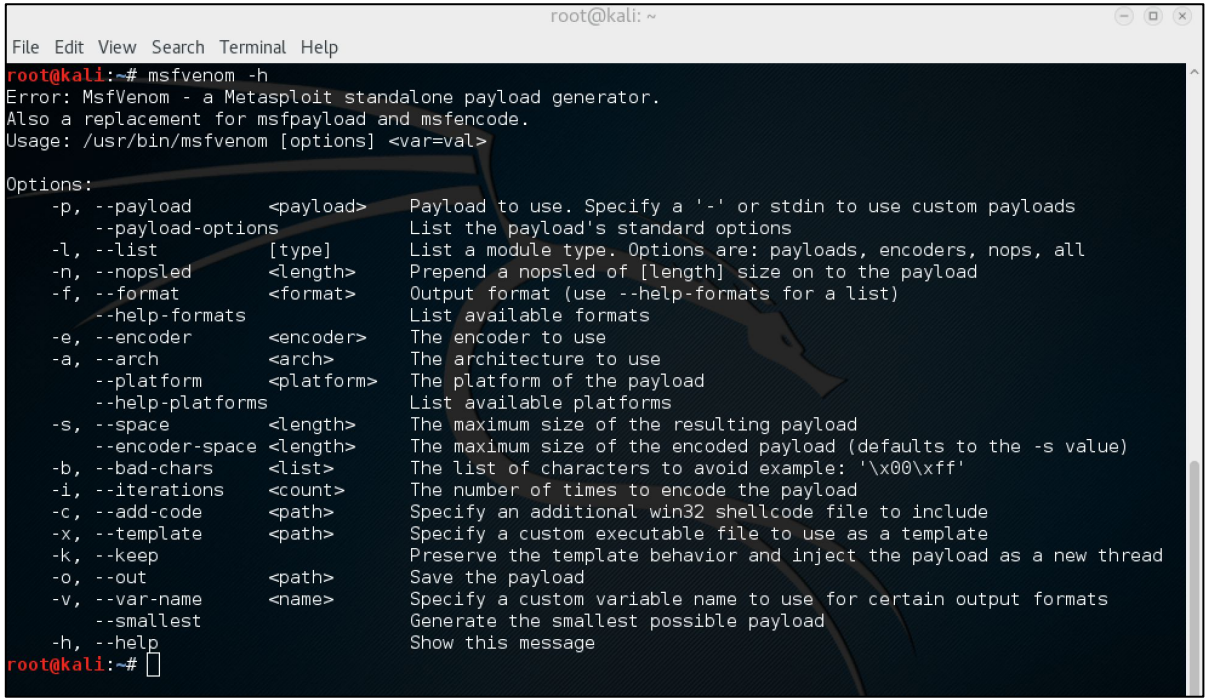
```
meterpreter > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    background    Backgrounds the current session
    channel       Displays information about active channels
...snip...
```

msfvenom

Allows the operator to craft the malicious agents and payloads that will be used to communicate back with Metasploit.

These agents and payloads can be created in several formats, including:

- .exe
- .py
- bash
- and much more



Today, we will focus on MSFconsole. We'll explore Meterpreter and `msfvenom` in depth in future classes.

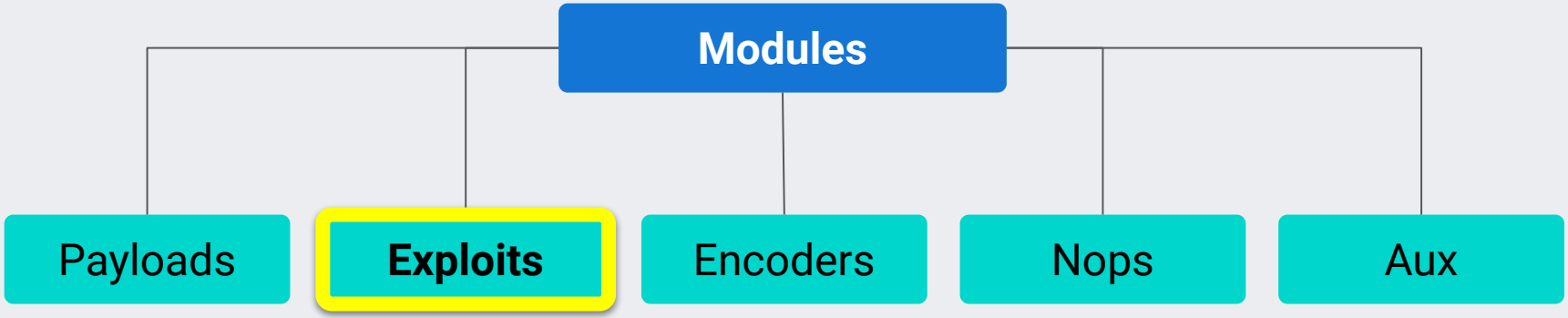
Metasploit Modules

MSFconsole is a unified interface for a variety of functions.
Each of these functions is called a **module**.

Auxiliary modules	Used for information gathering, enumeration, and port scanning. Can also be used for things like connecting to SQL databases and performing man-in-the-middle attacks.
Exploit modules	Generally used to deliver exploit code to a target system.
Post modules	Offers post-exploitation tools such as the ability to extract password hashes and access tokens. Provides modules for taking a screenshot, key-logging, and downloading files.
Payload modules	Used to create malicious payloads to use with an exploit. If possible, the aim is to upload a copy of Meterpreter, which is Metasploit's default payload.
Encoder modules	Encoders ensure that payloads make it to their destination intact.

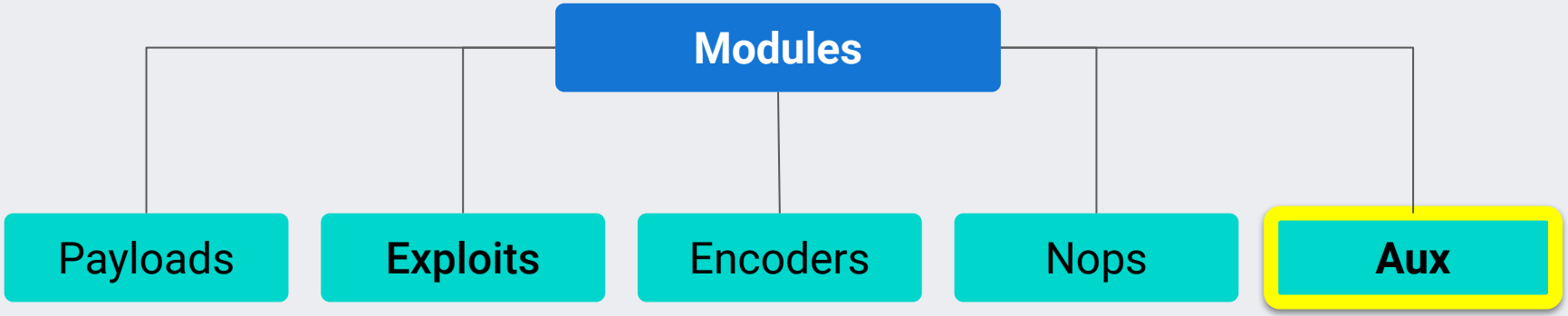
The choice of module depends on what you're trying to accomplish.

If the goal is to **exploit a vulnerability**…



…then an exploit module is appropriate,
as it actively tries to exploit the vulnerability.

If **recon** is the goal…



…then an auxiliary module is typically the right solution, as they're primarily for recon, scanning, and fuzzing.

Instructor Demonstration

Metasploit

Metasploit Demo

01 Launch MSFconsole.

02 Search for the commonly used auxiliary module `ftp/anonymous`, as this module can indicate if the target allows anonymous logins via FTP.

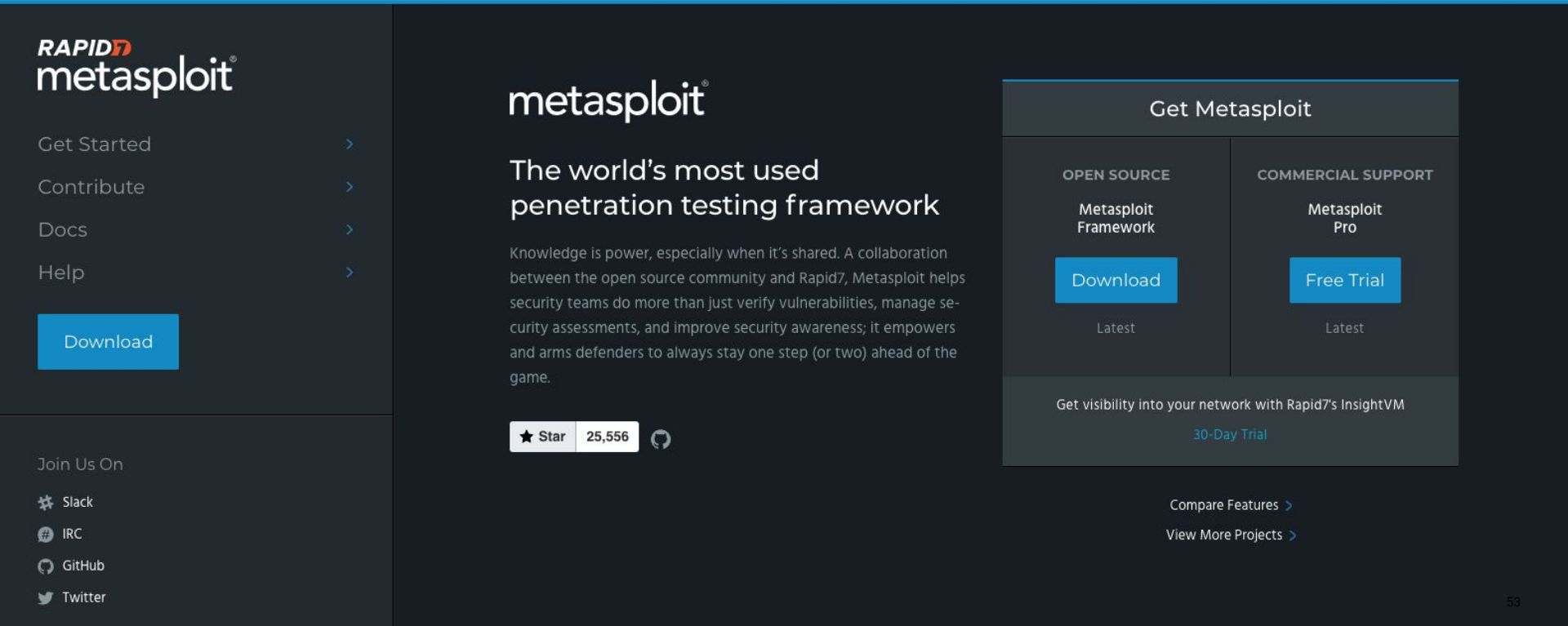
03 Select the `ftp/anonymous` module.

04 Set the required values for the `ftp/anonymous` module.

05 Execute the module.

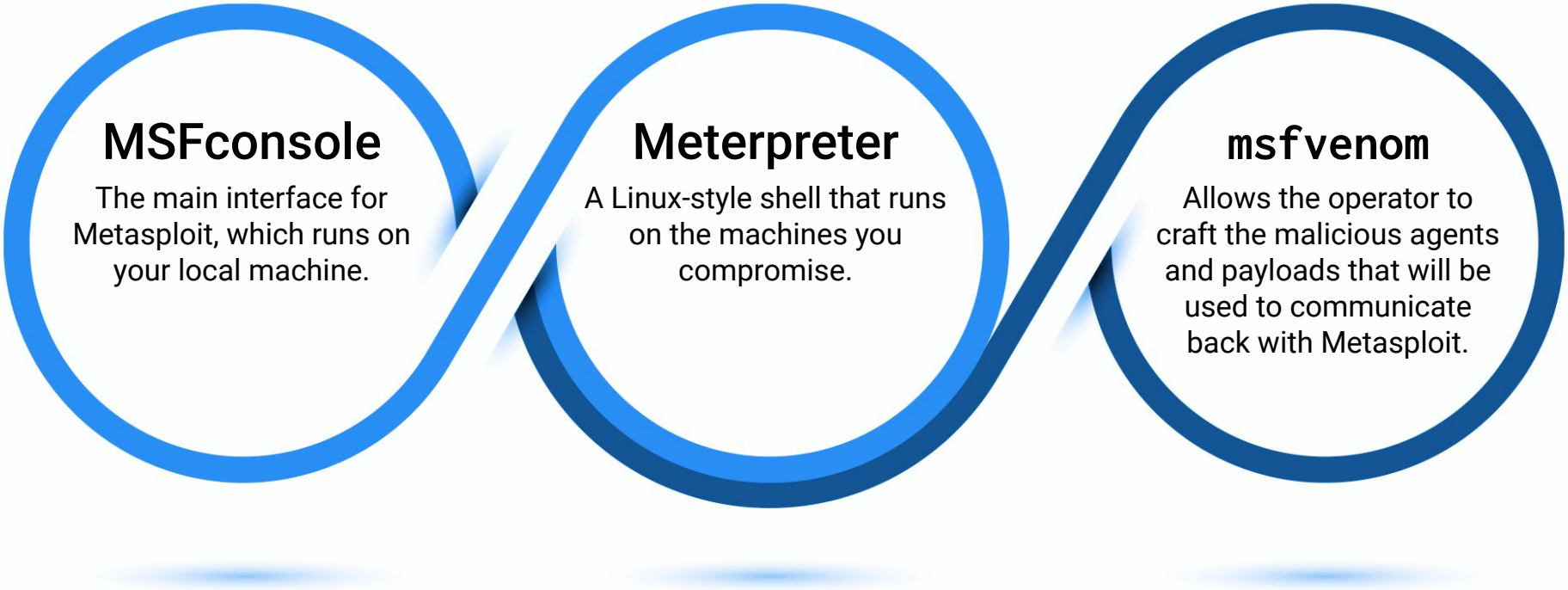

Recap

Metasploit is a popular C2 framework that contains a suite of tools for enumerating and exploiting servers and other networked devices.



Metasploit

The main tools of Metasploit are:



Metasploit Modules

MSFconsole is a unified interface for a variety of functions.
Each of these functions is called a module.

The five main types of Metasploit modules are:

Auxiliary modules	Used for information gathering, enumeration, and port scanning.
Exploit modules	Generally used to deliver exploit code to a target system.
Post modules	Offer post-exploitation tools.
Payload modules	Used to create malicious payloads to use with an exploit.
Encoder modules	Used to ensure that payloads make it to their destination intact.

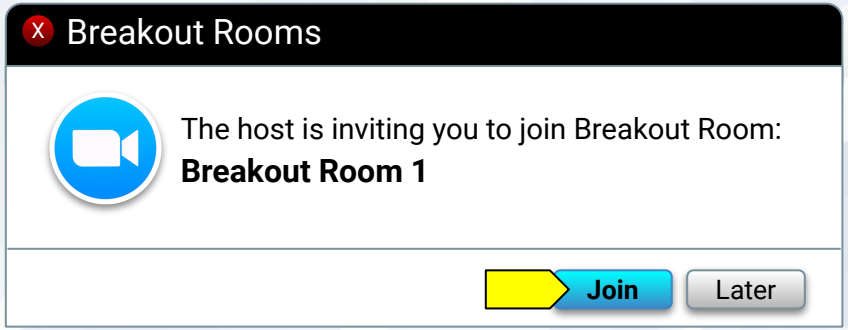
Metasploit Commands

`msfconsole`	Launches MSFconsole from the command line.
`Show` **+ module type**	Displays all the available modules. (For example: `show auxiliary`)
`Search` **+ keyword**	Searches for modules based on a keyword.
`Use` **+ module name**	Selects a specific module. (For example: `use scanner/ftp/anonymous`)
`info`	Displays the module details as well as the required and optional options.
`options`	Just displays the options available for the module.
`exploit OR run`	Runs the module.

Questions?

Activity: Metasploit

In this activity, you will exploit the vulnerable services that you discovered in your previous Nmap scan and obtain a reverse shell on the remote host by using an exploit module.



Suggested Time:

35 Minutes

Time's Up! Let's Review.

Questions?


Break

Phase 4:
Post Exploitation


Recap

Day 3 Recap

Let's recap what we've covered so far:

We introduced C2 frameworks and learned how penetration testers utilize their tools and features to conduct various aspects of penetration tests.

We then learned about a popular C2 framework called **Metasploit**, and how it can be used to:

- Search for potential exploits.
- Run the potential exploits.

Getting a shell on a machine is only the first part of having full control over the system.

The last activity you completed was part of the Exploitation phase.

Next, we'll focus on the **Post Exploitation** phase.

Reporting

Post Exploitation

Exploitation

Scanning

Planning and Reconnaissance

Phase 4: Post Exploitation

Post Exploitation involves multiple steps, typically used to accomplish the following three tasks:

01 Enumeration and searching for useful data

02 Persistence

03 Privilege escalation

Phase 4: Post Exploitation

01 Enumeration and searching for useful data

Similar to the enumeration process during the Reconnaissance phase, except we are enumerating and searching for data from *inside* the target after it has been exploited.

This means we search for any useful information that's available from inside the target, including:

Current user context	(Name, groups, ID, etc.)
List of users on the machine	(Logged in and not)
Current user access	(Can they read the majority of files on the machine?)
Identification of any defensive countermeasures	(Antivirus, logging, endpoint detection & response solutions, etc.)

Phase 4: Post Exploitation

02 Persistence

The process of attempting to maintain long-term access to your target.

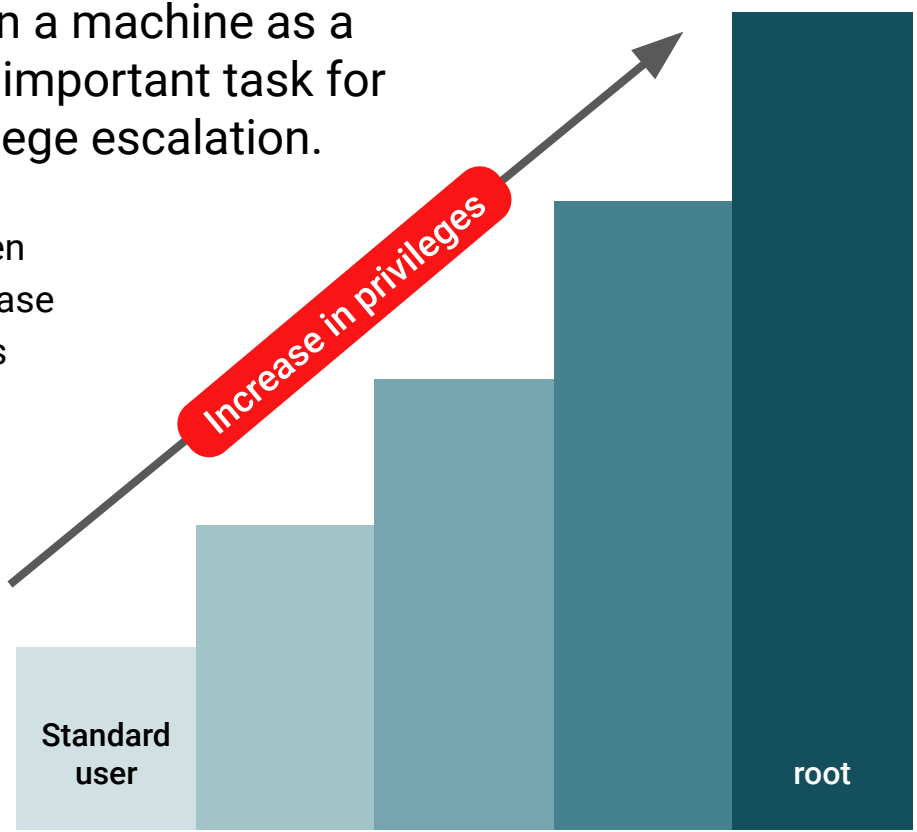
03 Privilege escalation

The process of escalating privileges from a less privileged user to a more privileged user (or a low-privilege to high-privilege user).

Privilege Escalation

When pen testing, it's common to land on a machine as a low- or no-privilege user. In this case, an important task for the penetration tester is to perform privilege escalation.

Much like the Reconnaissance phase is split between external and internal recon, the Post Exploitation phase is split between low-privilege post-exploitation tasks and high-privilege post-exploitation tasks.



High- and Low-Privilege Post-Exploitation Tasks

Low-Privilege Tasks

Require minimal permissions and include:

- Gathering system/OS details.
- Gathering services and versions.
- Getting a list of users and groups.

High-Privilege Tasks

Require sudo or root privileges and include:

- Reading sensitive files, such as `/etc/shadow` and SSH keys.
- Creating users and assigning them to privileged groups.
- Installing new services.
- The majority of persistence tasks.

The high-privilege user has more powerful capabilities, so it is important for a penetration tester to try and obtain these additional privileges.

Tips for Successful Privilege Escalation

01

Utilize information gathered during enumeration.

02

Use the path of least resistance.

03

Exploit vulnerable software.

Tips for Successful Privilege Escalation

01

Utilize information gathered during enumeration.

Gathering information allows us to identify a potential path to privilege escalation.

Common Linux paths of privilege escalation:

- **Abusing current access**
  (E.g., sudo group, writable files, read access to sensitive files)
- **Exploitation of software**
  (Some services may run as a higher-privileged user.)
- **Utilizing weak passwords of privileged users**

Tips for Successful Privilege Escalation

02

Use the path of least resistance.

Search for the easiest possible way to obtain your objective instead of jumping to a difficult task.

For example:

- Some users keep passwords in a text file. Instead of hijacking a user's web browsing session and keylogging their strokes in an attempt to gather passwords, you can just find the text file.
- In addition to being easier, finding passwords in files or abusing functionality (e.g., scripts run as root but world-writable) is also stealthier, which is a key component of red teaming.

Tips for Successful Privilege Escalation

03

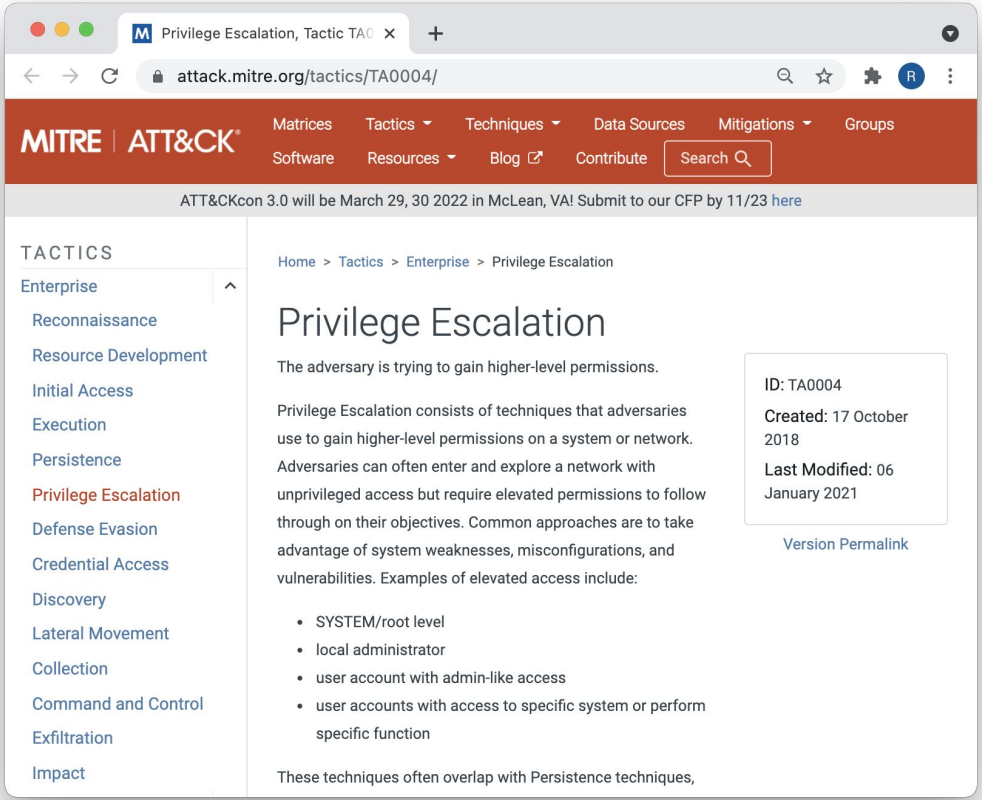
Exploit vulnerable software.

Exploitation of vulnerable software is often suggested as a last step due to the nature of privilege-escalation techniques.

- Often, it involves abusing a service which has the potential to crash and bring down the machine.
- For this reason, finding as much information as possible on the machine during enumeration is key to help find any other possible privilege-escalation paths.

MITRE <> Privilege Escalation

Privilege escalation has its own MITRE tactic.



In the next activity, you will attempt to find a way to escalate your privileges from the daemon user to a higher-privileged user.

But first, everyone needs to be at the same starting point on the remote machine.

Please follow along with the demonstration to get set up for the next activity.

Instructor Demonstration

Post Exploitation Activity Setup

After the demo, leave the current window and Metasploit session open for the next activity.

Activity: Privilege Escalation

In this activity, you will conduct post-exploitation tasks in order to attempt to gain privilege escalation.

Suggested Time:

20 Minutes

Privilege Escalation Activity

While this task may have seemed trivial, it is *extremely* common. Other trivial Linux privilege-escalation methods include:

- Being in the sudoers group.
- Finding files that are run as root but writable by anyone (also known as **setuid**).
- Finding passwords in configuration files (Apache and NGINX are common frameworks that have passwords in files).

Questions?


The End