



# Pentesting Review and Project




Cybersecurity Boot Camp  
Pentesting 3 Day 3

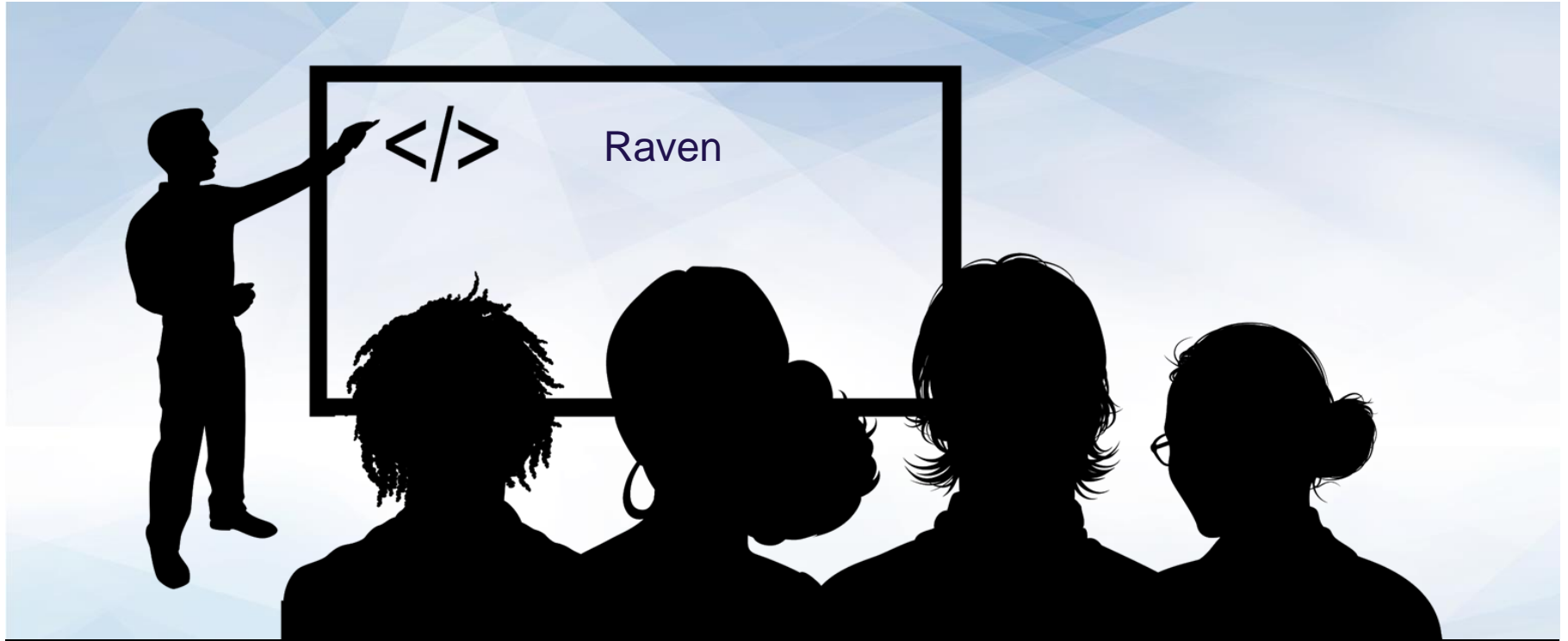


# Class Objectives

---

By the end of class today, students will be able to:

-  Perform URL enumeration and web-application fuzzing.
-  Use [wpscan](#) to enumerate username for a Wordpress installation
-  Brute-force a Wordpress login with [wpscan](#)



# Instructor Demonstration

## Project Set-up

# Attack Strategy

# Attack Strategy

---

Today's VM is built as a Capture the Flag (CTF) exercise.

You will have to complete a series of steps in order to find special files, known as “flags”.

Each flag is a string, like:

29024d823c3f8a90eeb71449204f77be3fbc7afec47873f6c5ddf9ea5e5cfe0f

Or a text file containing such a string, like:

flag.txt, flag2.txt, etc.

You will find as many flags as you can and then write a final report summarizing the vulnerabilities.

# Attack Strategy

---

You will complete the following steps in order to find flags:

- ▶ Use Nmap to discover the target's IP address, and then scan it for open ports. This will reveal an HTTP server, as well as an SSH server.
- ▶ Then, you will exploit vulnerabilities in the website to gain access to its administration panel.
- ▶ Finally, you will attack the SSH server to first get a user shell on the system, and then escalate privileges to “own” the machine.

# Attack Strategy

---

This VM contains four flags:

- ▶ One flag is located on the website served by the target's HTTP server.
- ▶ One flag is located in the administration panel for the target's WordPress installation.
- ▶ Two flags are located on the server's filesystem, outside of the web server directories.

# Possible Project Methods

---

These tasks can be completed in a myriad of ways, with a myriad of tools:

**Phase 1:** Identify all open ports and running services on the target with Nmap.

**Phase 2:** The Web Angle

- ☐ Generate a sitemap of the HTTP server with Burp Suite and Spider
- ☐ Find flags in the website's HTML
- ☐ Perform URL enumeration against the HTTP server with `wfuzz`.
- ☐ Perform user enumeration against the Wordpress blog with `wpscan`.
- ☐ Use `wpscan` to brute-force the passwords of discovered users.
- ☐ Find flags in the WordPress administrator panel.

**Phase 3:** The Network Angle

- ☐ Brute-force the target's SSH server
- ☐ Spawn a root shell using `python`.
- ☐ Find flags on the server's file-system.



# Network Enumeration

---

Network enumeration and host discovery reveal which machines are discoverable on the target network.



Nmap is the most important tool for network enumeration.



You should assess the IP address of the vulnerable VM via host discovery.



Then, port scan that IP address (revealing open TCP and UDP ports, OSm and versions of services running on open ports).



Try running the quietest port scan possible (-Pn, -n, --disable-arp-ping)

# Website Enumeration

---

Web Server exploitations begin with host enumeration. Attacker will often do the following:

01

Navigate to the target URL and browse the site manually.

02

Use Burp Suite to “spider” the site.

03

Use a tool called [wfuzz](#) to brute-force directory enumeration.

04

Use a tool called [wpscan](#) to attack WordPress installations.



Browsing a target site manually allows attackers to generate a sitemap without sending a large number of consecutive requests.

Thus, attackers can determine which resources the site exposes without triggering an intrusion detection system or rate-limiting firewall rule.

# Burp Spider

---

Spider automatically discovers web pages on a given domain. It works as follows:

01

Reads the HTML of the target site's home page.

02

Finds links to other pages on the site ("internal links").

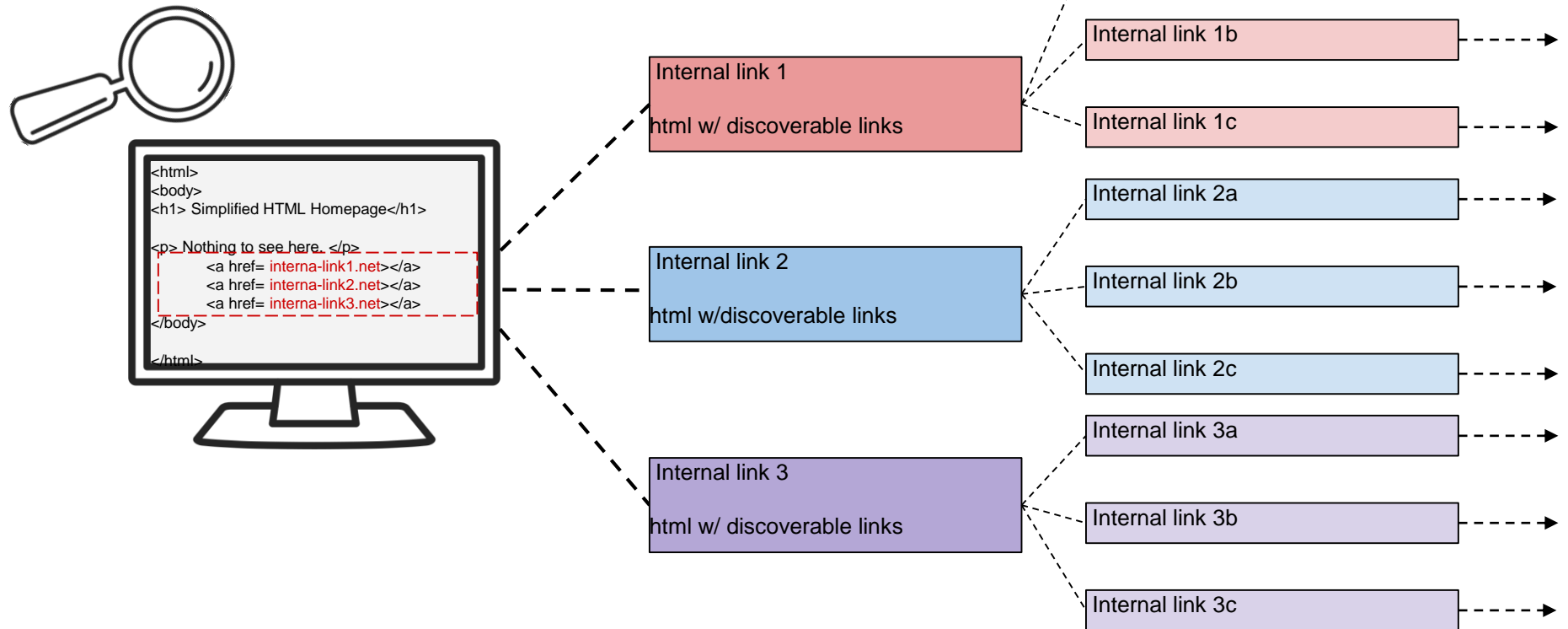
03

Follows these links and records the URLs.

04

Repeats the above steps on each of the discovered page.

# Burp Spider





Hidden files that are not linked from pages on the site can be discovered via brute-force URL enumeration.

# Brute-force URL enumeration

---

A fuzzing tool called [wffuzz](#) is used for brute-force URL enumeration.



Fuzzing is a vulnerability test in which attackers send a barrage of random information to an application, with the goal of finding errors in how applications handle unexpected input.

Fuzzing tools typically do one of three things:

1. Taking legal input and mutating it. E.g.: [john](#) → [john\[](#)
2. Generating random input from scratch. For example: [amy,amy1,as81::l](#)
3. A combination of the above.

# wfuzz for URL Enumeration

---

**wfuzz** simplifies the process of sending multiple url requests to a website and seeing which urls give errors.

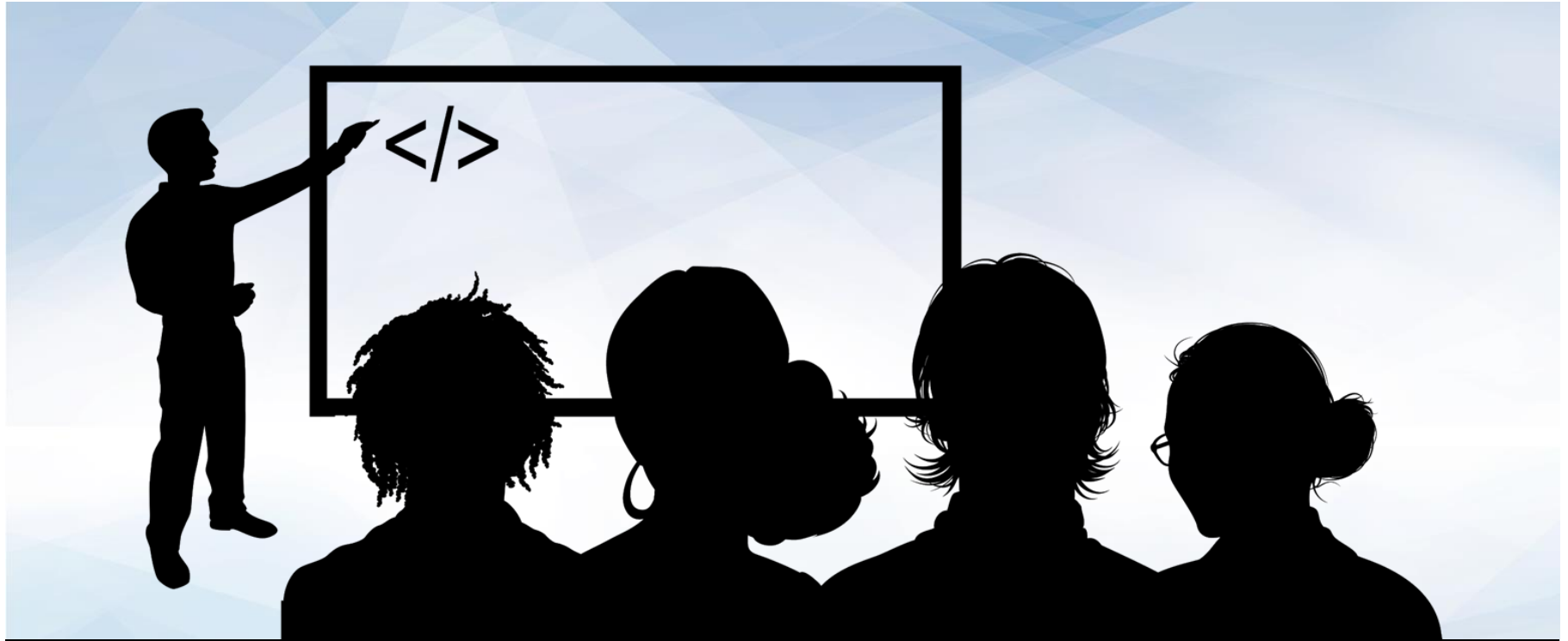
How it works:

- Provide **wfuzz** with a url and a list of words to inject into the url.
- **wfuzz** tries all the variations that result and reports what variations give errors or responses.



Fuzzing can cause denial of service because it requires sending a large number of requests very quickly. **Do not** use wfuzz against a site you do not have permission to attack.





# Instructor Demonstration

wfuzz



# WordPress



# WordPress

---

WordPress is an open-source PHP framework for building websites.



A content management system for writing, sharing, and reviewing content through blog posts, without knowing HTML, CSS, or PHP.



WordPress powers **33%** of websites on the Internet, making it a popular target for attacker.



WordPress is usually run on LAMP servers.



There is a large community of WordPress developers who create plugins for WP sites, which are often a source of major vulnerabilities.



Keeping WordPress and various plugins updated is a logistical challenge that often leaves websites vulnerable to attacks.

# wpscan

---

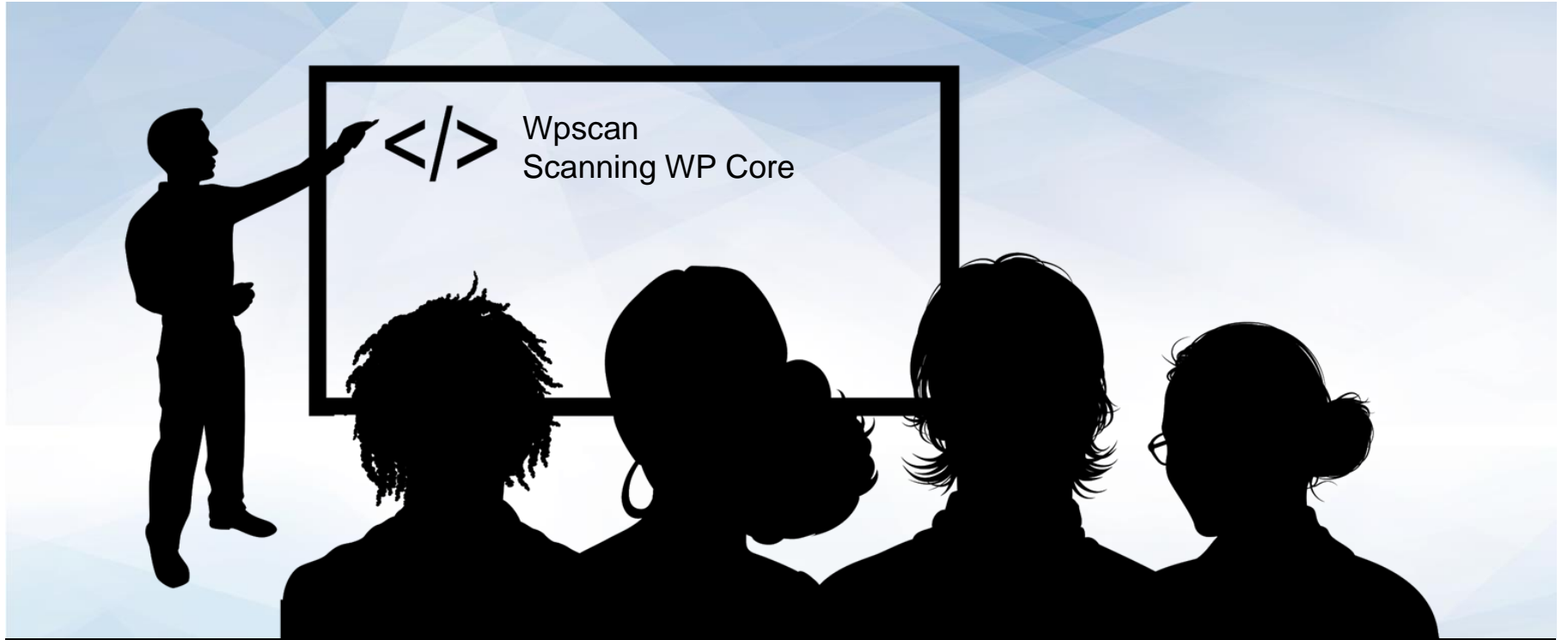
**wpscan** is a command line tool that performs WordPress-specific tasks:

1. Identifies the target WordPress version and find vulnerabilities in the core code for that version
1. Scans all the plugins that a WordPress site is using and reports potential vulnerabilities.

It also has options for enumerating users and brute forcing WordPress login credentials.

- Pentesters can attempt brute-force attacks against the login form using usernames wpscan discovers
- They can also use **wpscan** to identify exploitable vulnerabilities in WordPress core code or plugins.





# Instructor Demonstration

## WordPress Demo

# WordPress Demonstration

---

We will cover the following:

01

Hack.me Setup

02

wpscan

03

Scanning for Vulnerabilities in WordPress Core

04

Scanning for Vulnerabilities in Plugins

05

User Enumerations

06

Brute-Forcing the Login Form

# Summarizing Web Enumerations

---

## Some Takeaways:



When attacking a web server, begin by exploring manually and generating a Site Map with Burp Site



After exploring the site manually, use Burp Spider to find all internal links.



After finding internal links, use wfuzz to perform URL enumeration.



If you find a WordPress site, use wpscan to:



Determine the installed WordPress version and enumerate vulnerabilities in WordPress Core



Perform user enumeration



Brute-force user logins



# Summarizing Web Enumerations

---

## Some Takeaways:

- When attacking a web server, begin by exploring manually and generating a Site Map with Burp Site
- After exploring the site manually, use Burp Spider to find all internal links.
- After finding internal links, use wfuzz to perform URL enumeration.

If you find a WordPress site, use [wpscan](#) to:

- Determine the installed WordPress version and enumerate vulnerabilities in WordPress Core
- Perform user enumeration
- Brute-force user logins



# The Network Angle: The Second Attack Pathway

# The Network Angle

---



## Brute-forcing Login Protocols

In addition to URL enumeration of a web server, you'll use Hydra to attack a network protocol. You'll be responsible for using Nmap results to determine protocol to attack.



## Sudo Manipulation and Post-Exploitation

Brute-forcing a login server typically allows attackers to get a user shell. Then they'll want to escalate to superuser privileges



## Investigating MySQL

We'll be able to find credentials for logging into the MySQL database, allowing us to dump important information.

1. Find Database
2. Gain a shell on target system.
3. Use credentials to manually login.

# Pillaging MySQL

---

You'll use MySQL CLI to:

01

List all database

02

Select a specific database to explore

03

List all table in the selected database

04

List all columns in the selected table

05

Dump all rows from the selected table

06

Export the database as a text file with mysqldump

# Project Objective Summary

---

**Phase 1:** Identify all open ports and running services on the target with Nmap.

## Phase 2: The Web Angle

- ☐ Generate a sitemap of the HTTP server with Burp Suite and Spider
- ☐ Find flags in the website's HTML
- ☐ Perform URL enumeration against the HTTP server with `wfuzz`.
- ☐ Perform user enumeration against the Wordpress blog with `wpscan`.
- ☐ Use `wpscan` to brute-force the passwords of discovered users.
- ☐ Find flags in the WordPress administrator panel.

## Phase 3: The Network Angle

- ☐ Brute-force the target's SSH server
- ☐ Spawn a root shell using `python`.
- ☐ Find flags on the server's file-system.

## Phase 4: Bonus

- ☐ Find MySQL credentials on the target file-system.
- ☐ Dump the database.



## Activity: Begin Attack Strategy Project.

With the remaining class time you will begin the project. Take your time, work with your classmates, and consult your instructional team if you have any questions.

**Suggested Time:**  
Until End of Class

