Web Vulnerabilities and Hardening: Exploitation and Mitigation

Cybersecurity

Web Vulnerabilities and Hardening Day 2

Class Objectives

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

Use SQLMap to execute SQLi attacks.

Execute a BeEF hook to perform various client-side attacks against the victim's web browser.

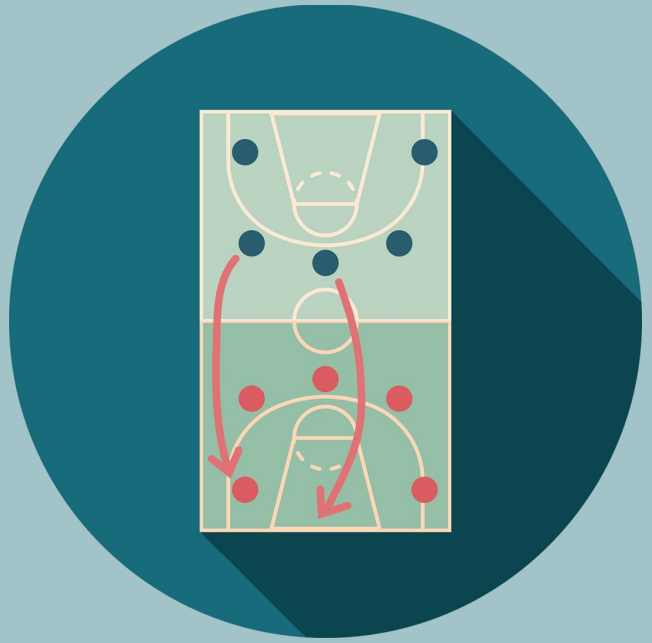
Perform a command injection on a Windows machine to dump and exfiltrate hashed passwords.

Provide mitigation strategies for all executed attacks.

Offense Informs Defense

Throughout this unit, we will act out examples of malicious attacks to show how various hacks and exploits work and how we can better defend against them.

It is important to note that the skills we learn in offensive security units should only be used ethically and with permission.



The actions and intents of criminal hackers, hacktivists and other malicious actors that we mimic for demonstrations are in no way condoned or encouraged.

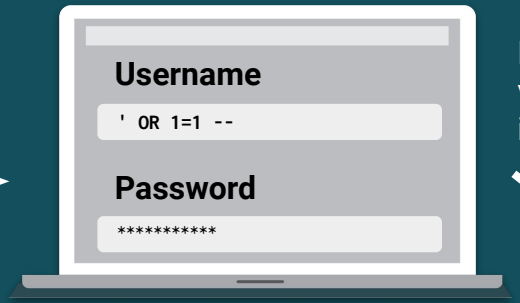
Injections


Injection flaws are at the most exploited web vulnerability in the OWASP Top 10.

SQLi Injections

Injections are a threat to any organization hosting web-based database servers. One specific threat is SQLi attacks.

Hacker identifies vulnerable SQL-driven website and injects malicious SQL query via input data.



Malicious SQL query is validated and the command is executed by the database

Hacker is granted access to view and alter records or act as database administrator

SQLi Injections

S Q L

Structured **Query** **Language**

A language for programming and managing databases.

SQLi attacks inject malicious SQL code into a client-side application such as a browser, revealing private data within the database.

This flaw is easily detectable and exploitable. Any website, no matter how many users it has, may experience these kinds of attacks.

SQLi Injections

Criminal hackers can use SQLi attacks to do the following:

Spoof a user's identity.	SQL commands can be manipulated to scan, modify, and extract usernames and passwords, allowing an attacker to connect as an authorized user.	This affects **Authenticity.**
Expose sensitive data.	SQLi leverage the leak of sensitive data in SQL databases.	This affects **Confidentiality.**
Modify existing data.	The possibility to read sensitive information also makes it possible to modify or delete critical information.	This affects **Integrity**.

SQLi Injections

Criminal hackers can use SQLi attacks to do the following:

Spoof a user's identity.	SQL commands can be manipulated to scan, modify, and extract usernames and passwords, allowing an attacker to connect as an authorized user.	This affects **Authenticity.**
Expose sensitive data.	SQLi leverage the leak of sensitive data in SQL databases.	This affects **Confidentiality.**
Modify existing data.	The possibility to read sensitive information also makes it possible to modify or delete critical information.	This affects **Integrity**.

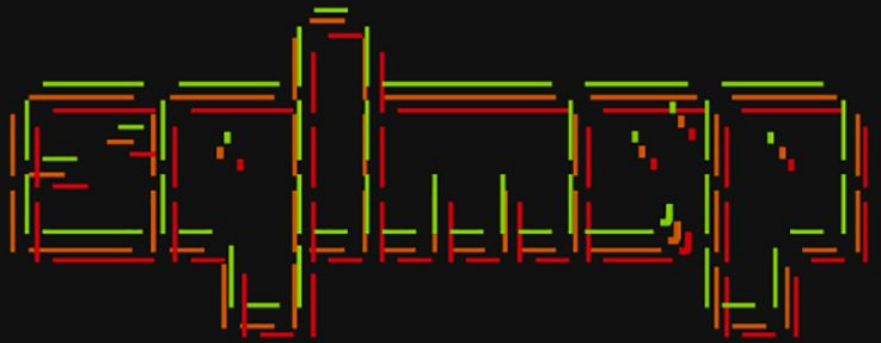
SQLi Injections

Criminal hackers can use SQLi attacks to do the following:

Spoof a user's identity.	SQL commands can be manipulated to scan, modify, and extract usernames and passwords, allowing an attacker to connect as an authorized user.	This affects **Authenticity.**
Expose sensitive data.	SQLi leverage the leak of sensitive data in SQL databases.	This affects **Confidentiality.**
Modify existing data.	The possibility to read sensitive information also makes it possible to modify or delete critical information.	This affects **Integrity**.

SQLMap

SQLMap is an open-source command-line tool that automates the process of detecting and exploiting SQL injection flaws in order to take control of database servers.



SQLMap

SQLMap is an open-source command-line tool that automates the process of detecting and exploiting SQL injection flaws in order to take control of database servers.

SQLMap contains a powerful detection engine that allows attackers to access to an underlying database file system.

SQLMap

SQLMap is an open-source command-line tool that automates the process of detecting and exploiting SQL injection flaws in order to take control of database servers.

SQLMap contains a powerful detection engine that allows attackers to access to an underlying database file system.

Attackers can use SQLMap to execute commands on a database server with out-of-band connections, meaning they can remotely control a back-end database using a back door connection, such as an RAT (Remote Access Trojan).

SQLMap Demo

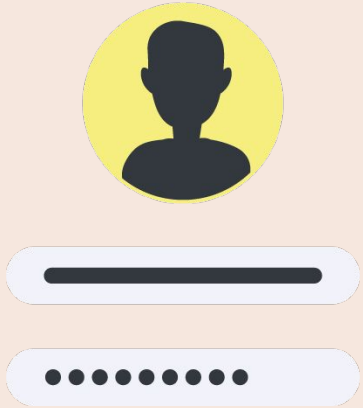
We'll demonstrate a SQLi attack with the following scenario:

A recently fired employee knows of a vulnerability in the company's back-end database.

SQLMap Demo

We'll demonstrate a SQLi attack with the following scenario:

A recently fired employee knows of a vulnerability in the company's back-end database.

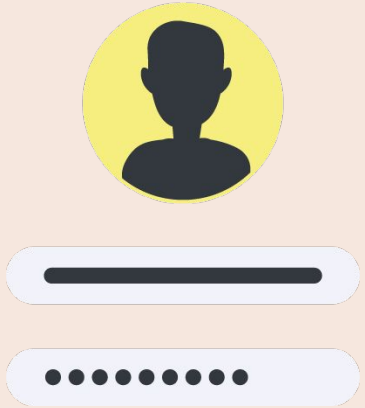


They will exploit it and steal usernames and passwords with the intent to sell them on the dark web.

SQLMap Demo

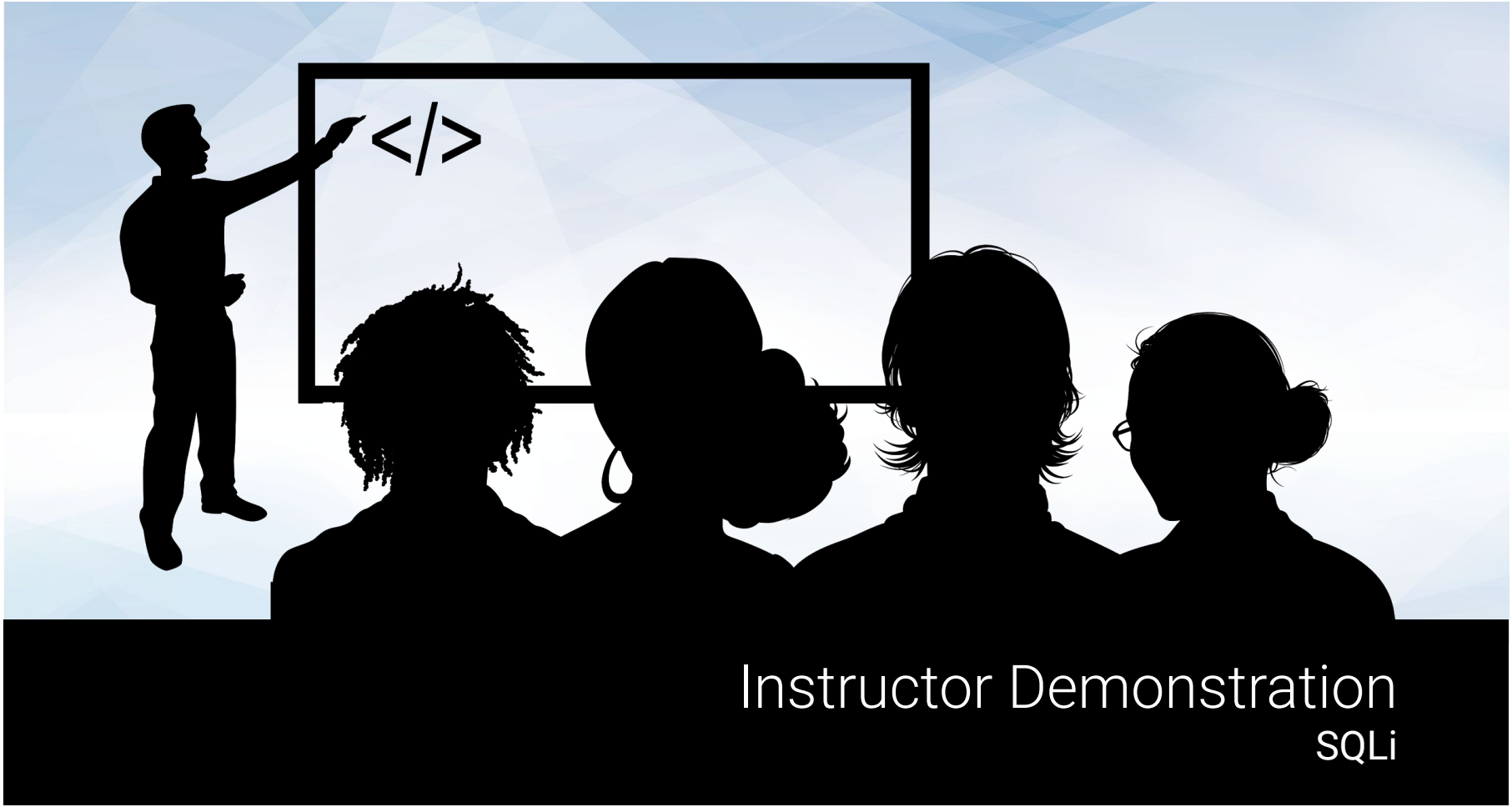
We'll demonstrate a SQLi attack with the following scenario:

A recently fired employee knows of a vulnerability in the company's back-end database.



They will exploit it and steal usernames and passwords with the intent to sell them on the dark web.

They will use SQLMap to enumerate usernames and passwords, then dump them for extraction.



Instructor Demonstration

SQLi

SQLMap Demo Takeaways

01

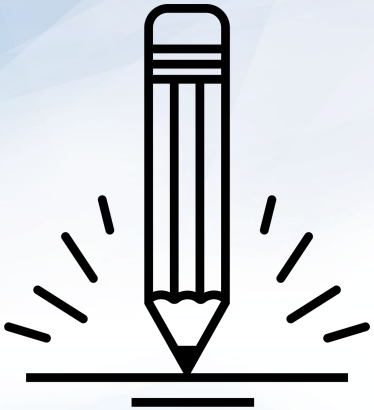
Back-end database systems are valuable sources of information for hackers.

02

Complacency can cause significant harm. Remember that just because back-end databases are deep in web server architecture and protected by firewalls doesn't mean they’re safe.

03

As proven in this demonstration, the URL can be manipulated in various ways to circumvent layered defense mechanisms contained within web infrastructure. This is accomplished by exploiting existing trust-based systems that face the public, such as HTTP port 80 and the URL.



Activity: Mapping the Database

In this activity, you will use SQLMap to expose a vulnerable back-end server.

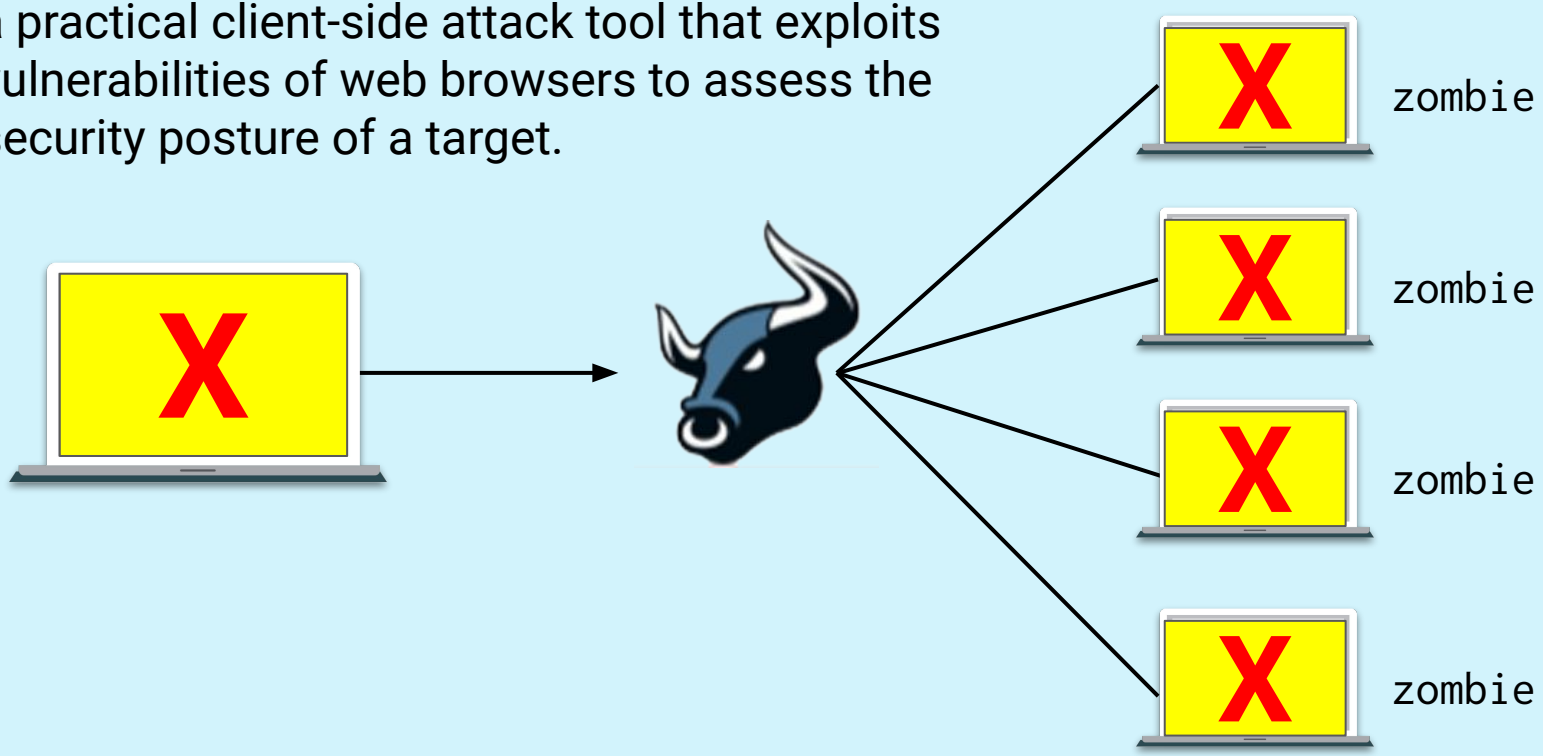
Suggested Time:
25 Minutes

Time’s Up! Let’s Review.

Browser Exploitation Framework (BeEF)

Browser Exploitation Framework

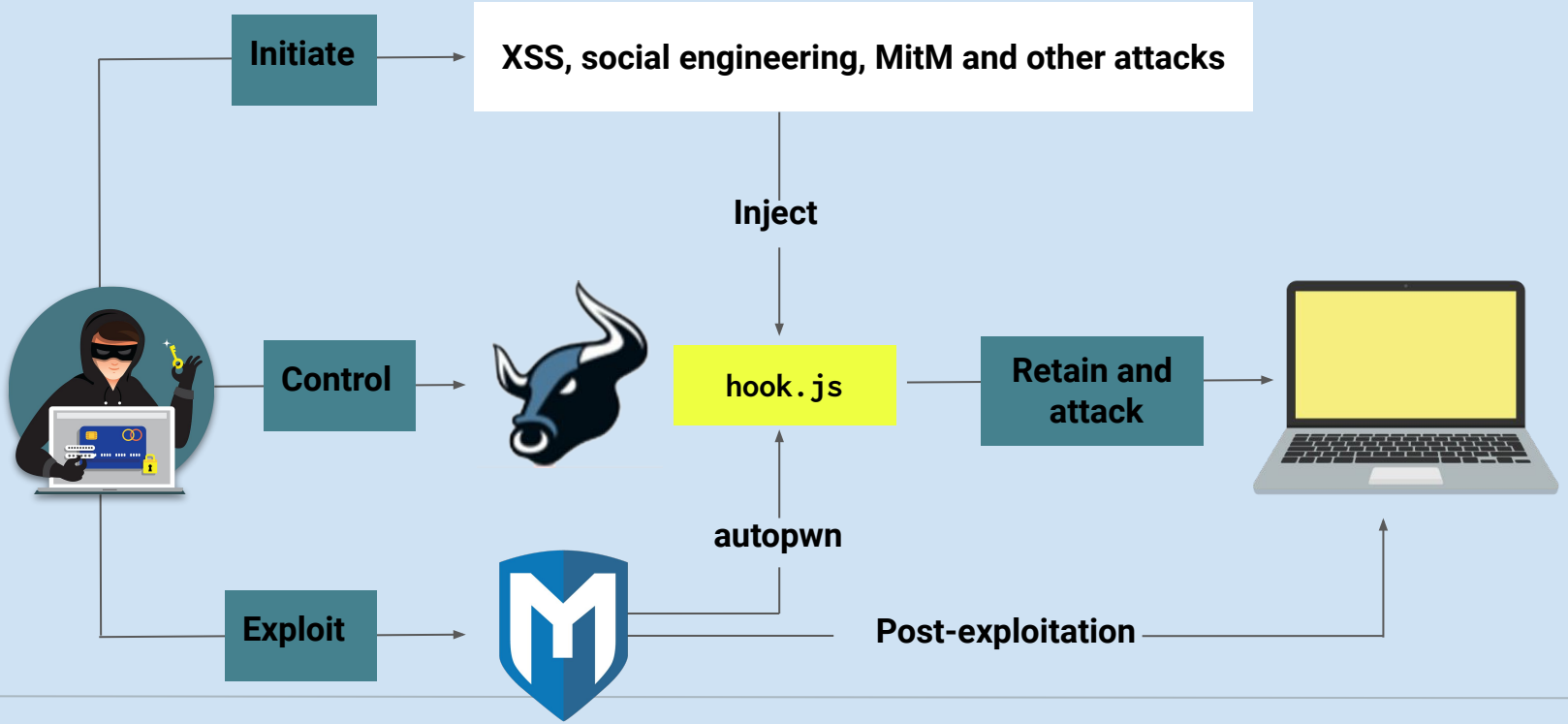
The Browser Exploitation Framework (BeEF) is a practical client-side attack tool that exploits vulnerabilities of web browsers to assess the security posture of a target.



BeEF was developed for lawful research and penetration testing, but criminal hackers have started using it as an attack tool.

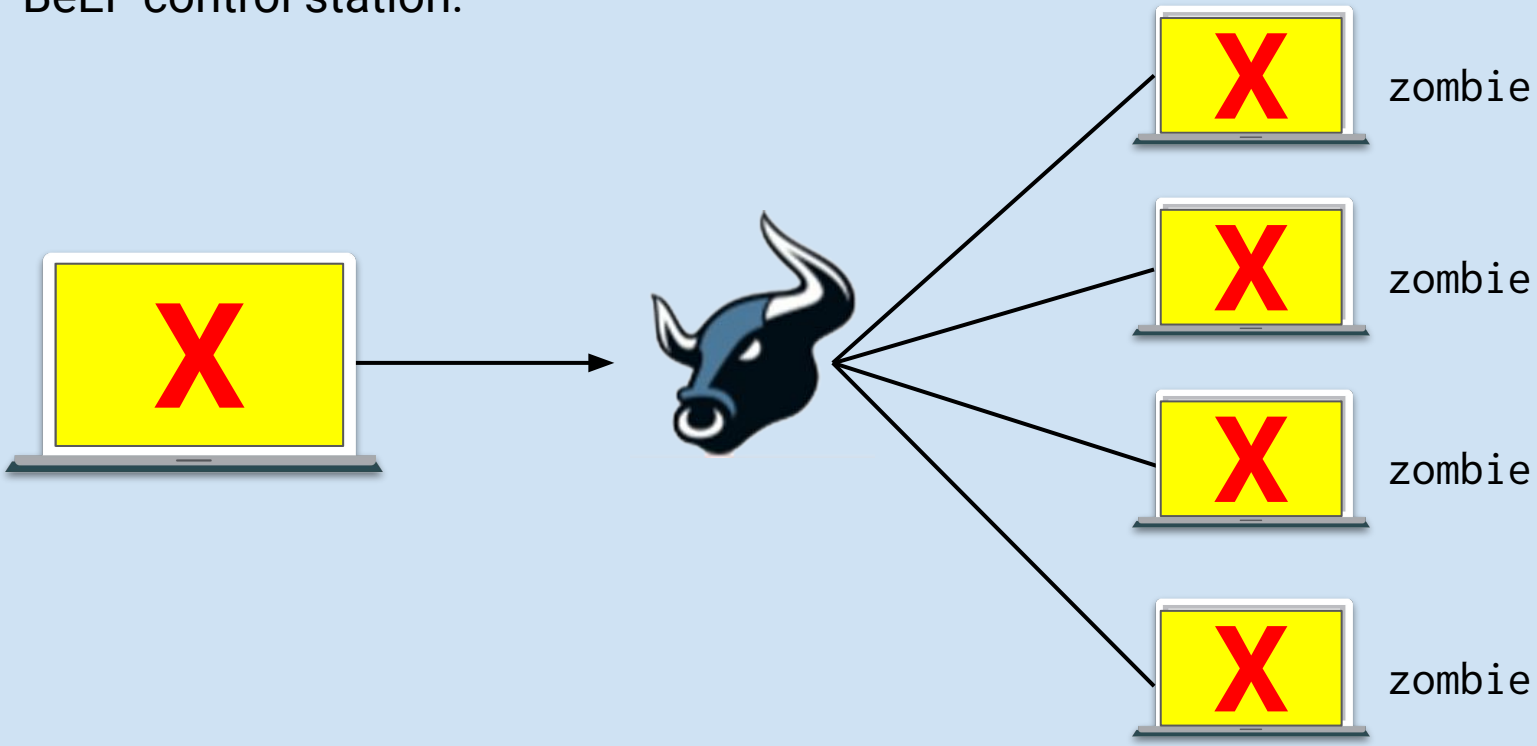
Browser Exploitation Framework

BeEF uses "hooks" to activate a simple but powerful API, which takes remote control of client-based web browsers.



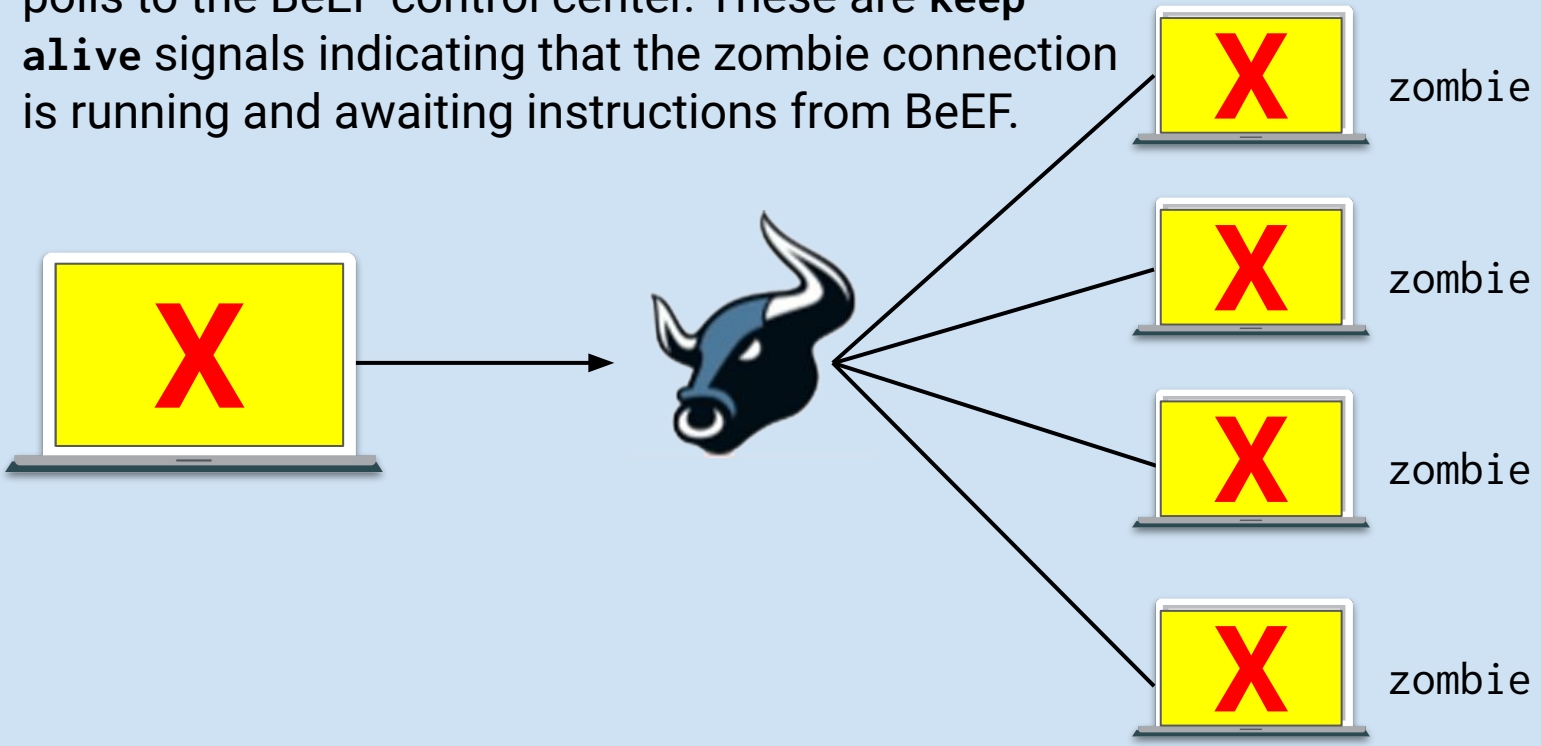
Browser Exploitation Framework

Once a browser is hooked, it becomes a zombie and awaits instructions from the BeEF control station.



Browser Exploitation Framework

Zombies that have been hooked by BeEF send out polls to the BeEF control center. These are **`keep alive`** signals indicating that the zombie connection is running and awaiting instructions from BeEF.



Browser Exploitation Framework

01

The majority of BeEF exploits occur as the result of an XSS attack, along with social engineering and man-in-the-middle attacks.

02

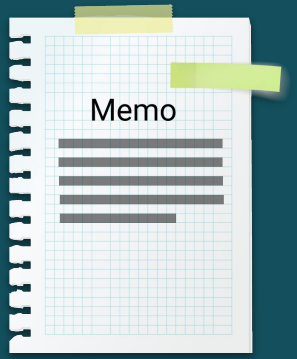
Other attack programs can be used as part of a post-exploitation campaign. While outside the scope of this lesson, the Metasploitable Framework provides a wide variety of post-exploitation attack modules.

03

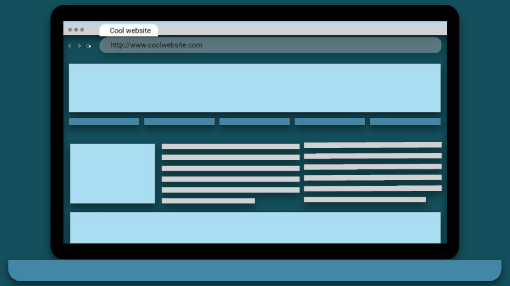
The BeEF framework also allows more advanced criminal hackers to integrate custom scripts.

BeEF Demo Set Up

We'll demonstrate BeEf with the following scenario:



Your CISO released a memo about potential web vulnerabilities in the company's web browsers.



Your security manager asked you to perform a penetration test to identify any underlying client-side browser vulnerabilities and recommend mitigation strategies.

You will use the BeEF framework to perform your research during your penetration tests.

BeEF Demo

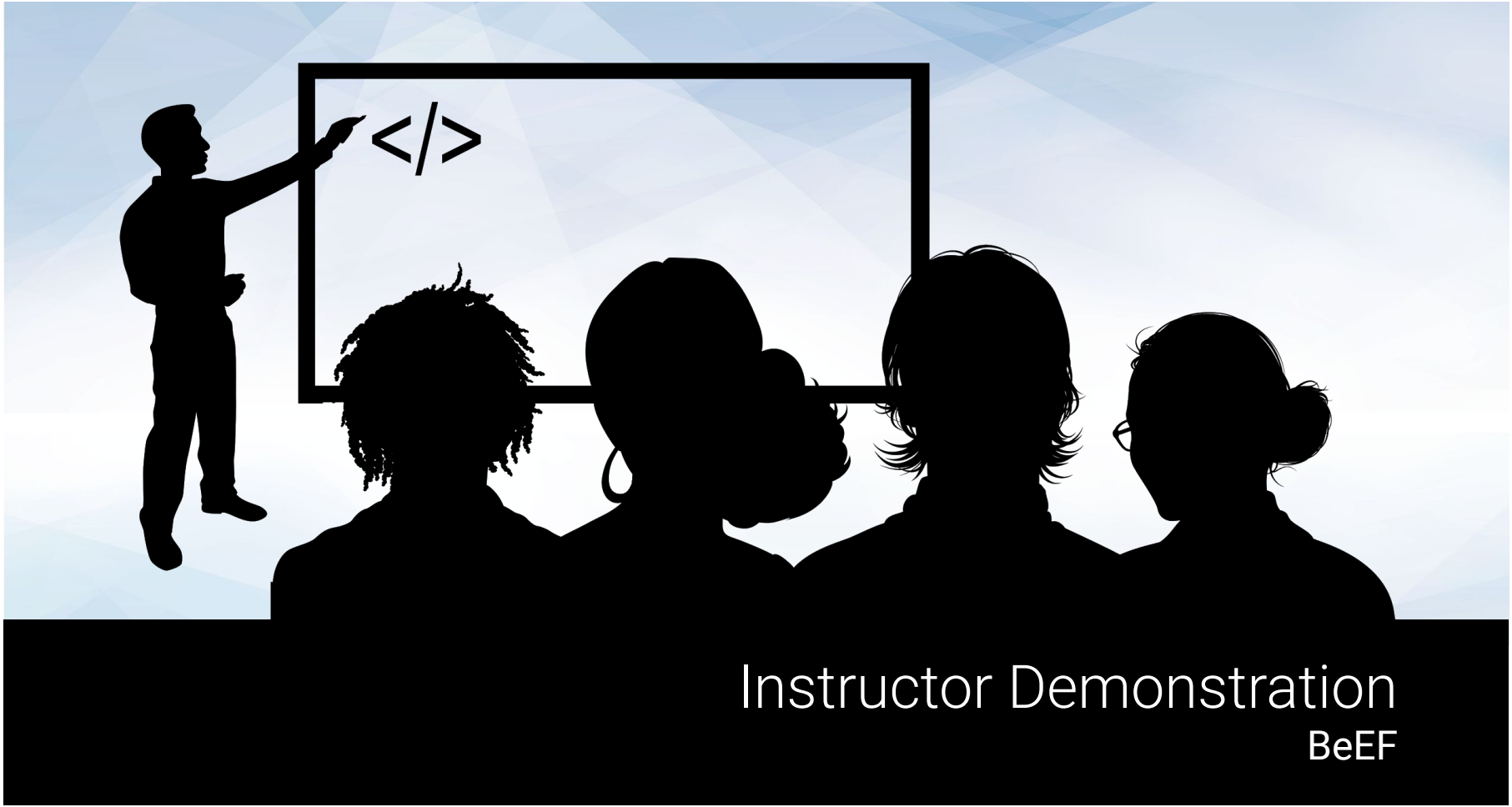
We'll complete the following steps:

Edit the `.html` file and add malicious JavaScript to the webpage that we'll use in our attack.

Visit the infected webpage from a host on the network (the victim).

Execute a BeEF **hook**, then perform various client-side attacks against the victim's web browser.

Recommend three mitigation strategies that defend against BeEF hooks and malicious JavaScript.



Instructor Demonstration

BeEF

BeEF Mitigation

Mitigation strategies against BeEF hooks include:

01

Vegan Chrome Browser Extension

This extension detects BeEF hooks and blocks the offending domain, preventing the attack.

02

Snort Rule

Add an emerging threats Snort rule to the company's IDS.

03

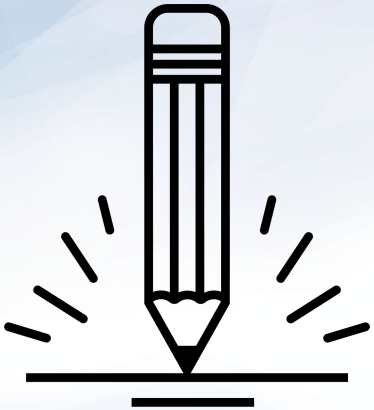
Content Security Policy

An added layer of network security that detects and mitigates specific types of attacks, such as XSS and injection attacks.

Countdown timer

15:00

(with alarm)



Activity: BeEF

In this activity, you will use BeEf to test a web client for vulnerabilities.

Suggested Time:
25 Minutes

Time’s Up! Let’s Review.

Windows-Based Injections

Windows Injections

Now we'll look at injections specific to Windows machines.

To search for a file located on the `c:\` drive:

```
dir c:\filename /s
```

Would return the location of a file called **`filename`**.

Windows Injections

```
dir c:\filename /s
```

Returns the location of a file called **`filename`**.

Command to list the directory.

Windows Injections

```
dir c:\filename /s
```

Returns the location of a file called **`filename`.**

Argument the command is run against.

Windows Injections

`dir c:\filename` `/s` Returns the location of a file called **`filename`**.

Lists the file if included in a subdirectory.

Windows Injections

We can inject code into files by outputting strings of characters into an argument with commands like **echo**.

```
&& echo This_is_a_test > "c:\filename"
```

Links this command to the previous **`dir c:\filename /s`** command.

Windows Injections

We can inject code into files by outputting strings of characters into an argument with commands like **echo**.

```
&& echo This_is_a_test > "c:\filename"
```

Directs content into an argument.

Windows Injections

We can inject code into files by outputting strings of characters into an argument with commands like **echo**.

```
&& echo This_is_a_test > "c:\filename"
```

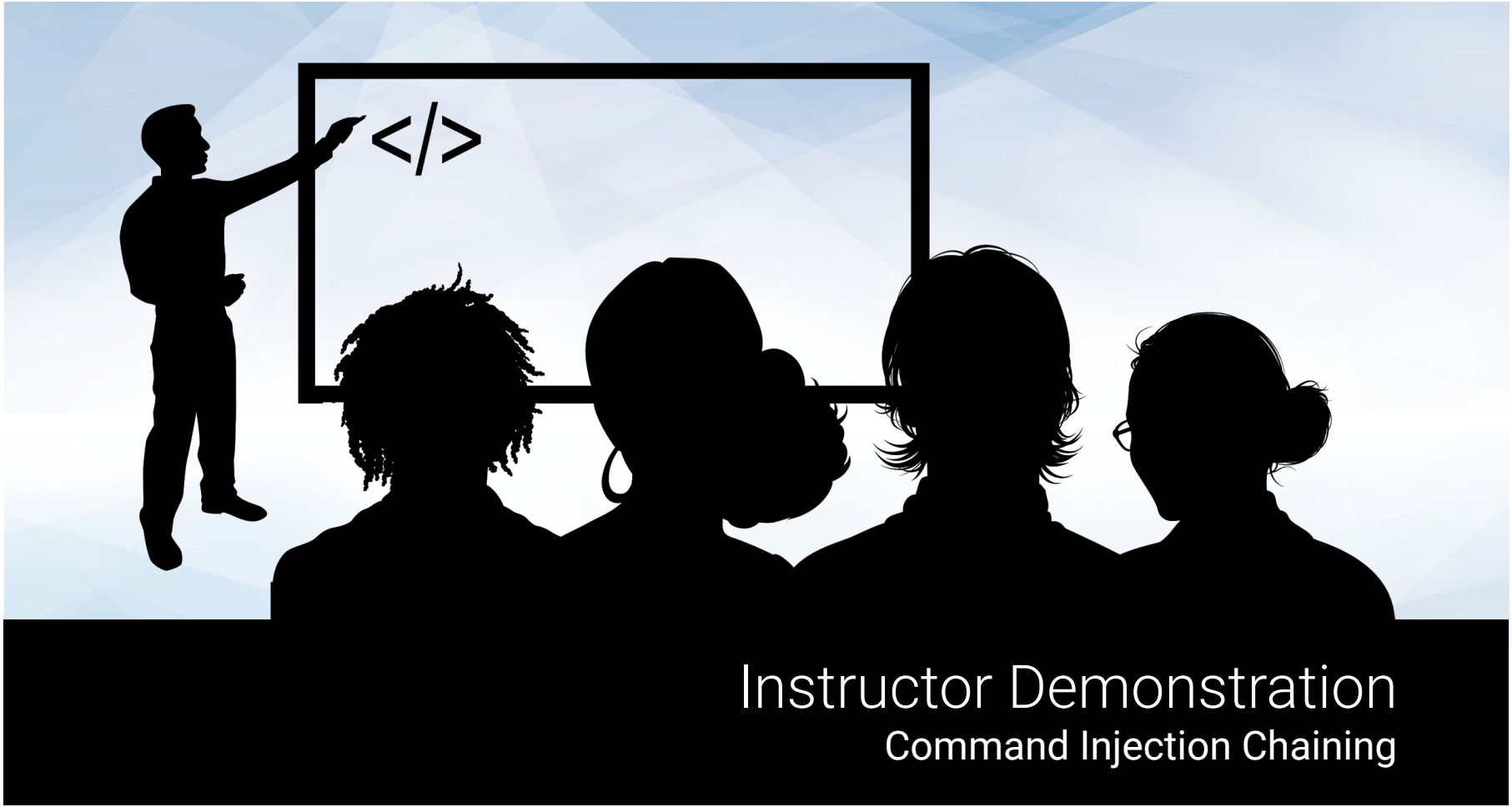
The string that is injected.

Windows Injections

We can inject code into files by outputting strings of characters into an argument with commands like **echo**.

```
&& echo This_is_a_test > "c:\filename"
```

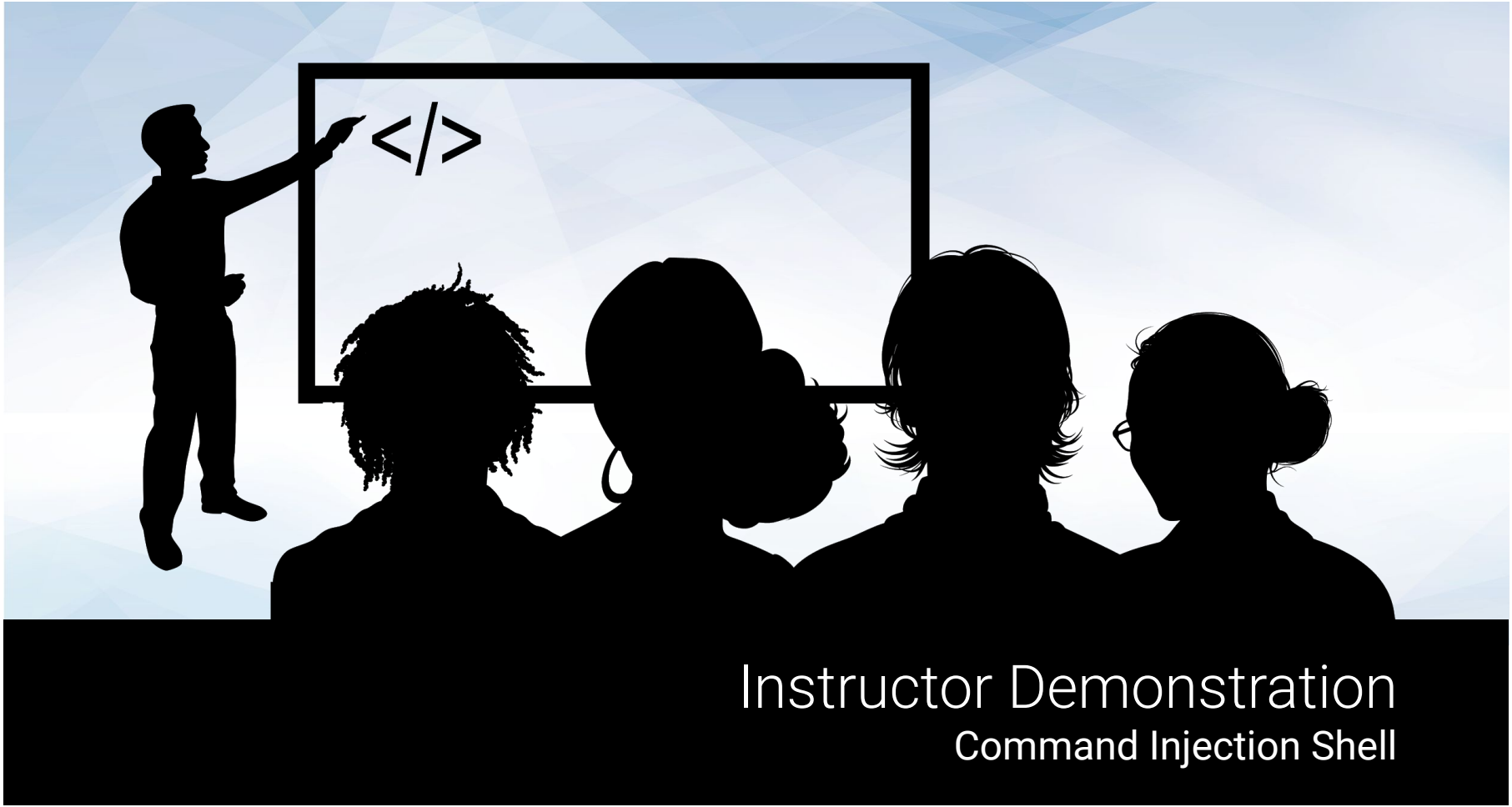
The file that is injected with code (the argument).



Instructor Demonstration

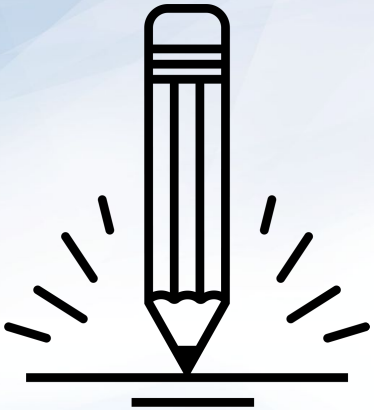
Command Injection Chaining

As an alternative to chaining, attackers can use a command injection shell to establish a reverse shell from the infected web server back to the attacker’s machine, where a listener waits to complete the connection.



Instructor Demonstration

Command Injection Shell



Activity: Command Injection

In this activity, you will perform a command injection attack that dumps hashed passwords into a text document, archives it, and compresses it in preparation for exfiltration.

Suggested Time:
15 Minutes

Time’s Up! Let’s Review.


The End