Introduction to Web Vulnerabilities and Hardening

Cybersecurity

Web Vulnerabilities and Hardening

Class Objectives

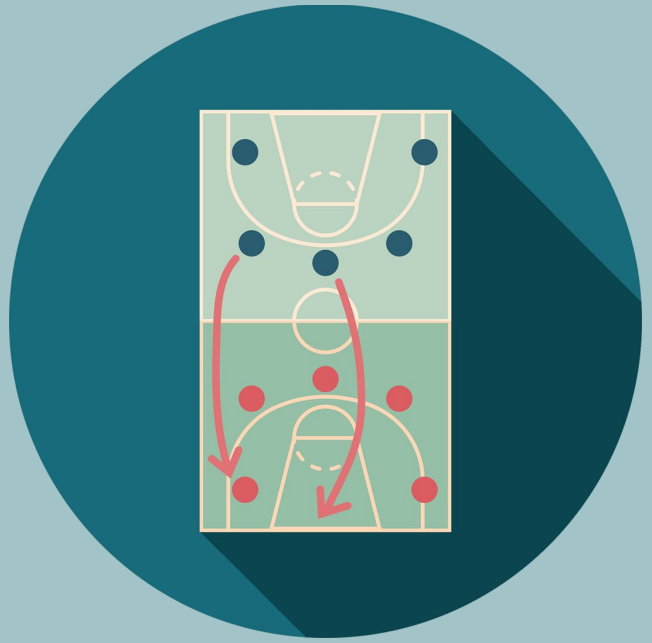
By the end of today's class, we'll be able to:

- Understand the attack-defend methodology for web vulnerabilities.
- Explain how a URL can be manipulated and used to take advantage of web vulnerabilities.
- Identify and differentiate between client- and server-side attacks.
- Use social media, WHOIS, and Wafw00f to gather information that informs attack options.
- Exploit three prevalent web vulnerabilities.

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.

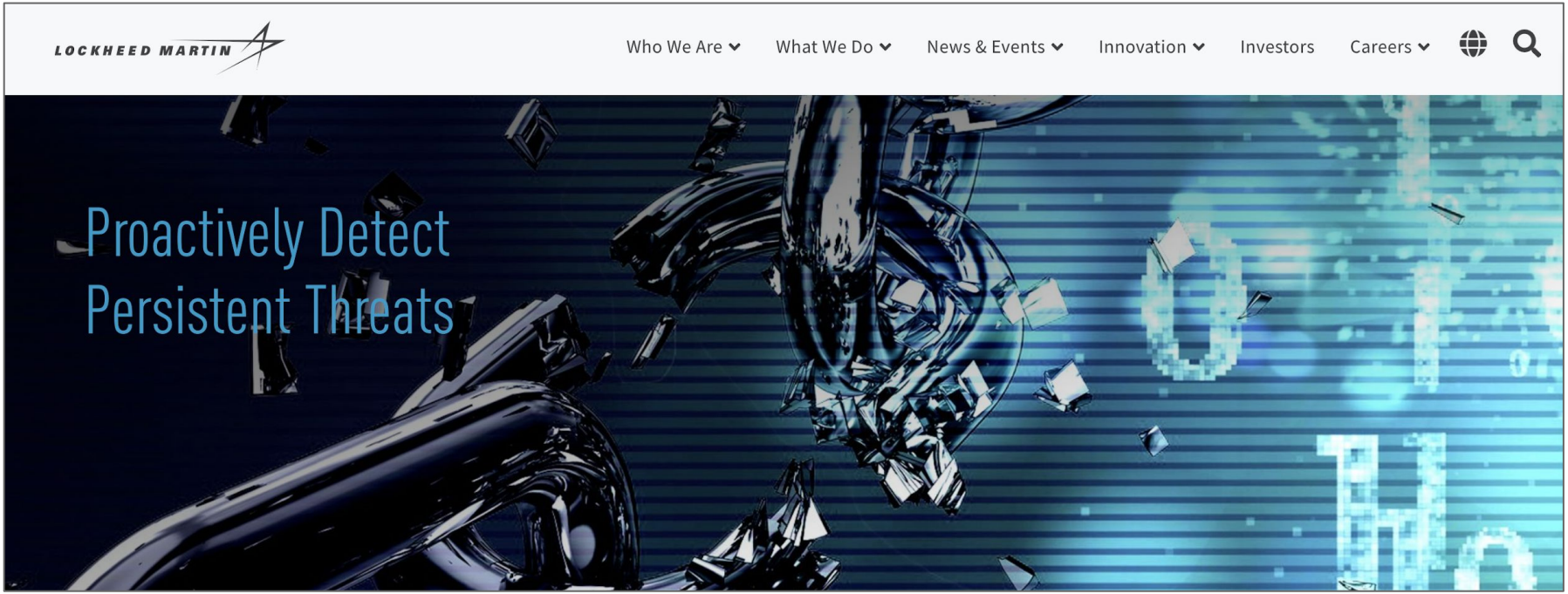


The actions and intents of criminal hackers, hacktivists and other malicious actors that we mimic for demonstrations are in no way condoned or encouraged.

Intro to Web Vulnerabilities and the OWASP Top 10

The Cyber Kill Chain

Lockheed Martin, an aerospace and defense company, developed another form of layered defense: the cyber kill chain.

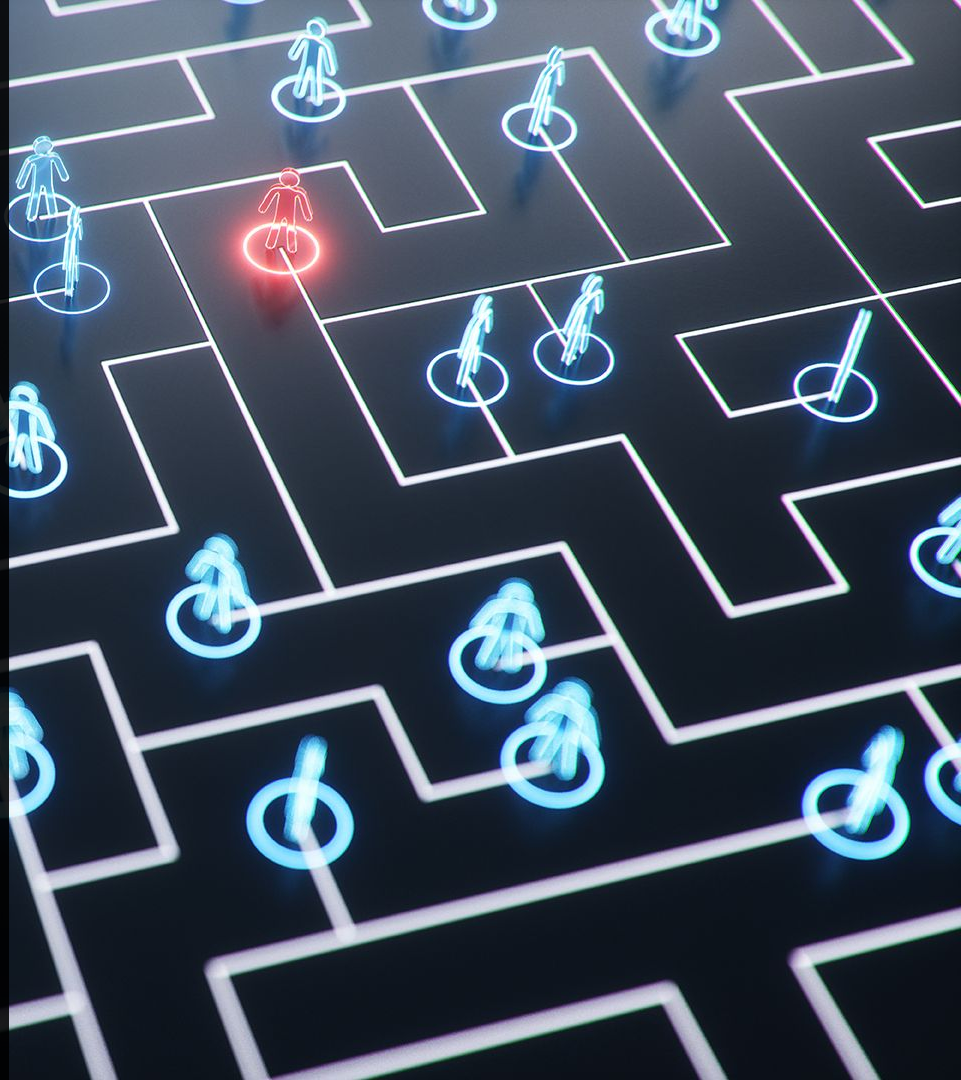


lockheedmartin.com

The cyber kill chain is an "intelligence-driven defense framework" designed to identify and prevent cyber intrusions.

The Cyber Kill Chain

This framework enhances visibility into an attack by improving a security analyst's understanding of an adversary's tactics, techniques, and procedures. It does this by allowing the observation of an attack as it progresses through each stage of the kill chain.

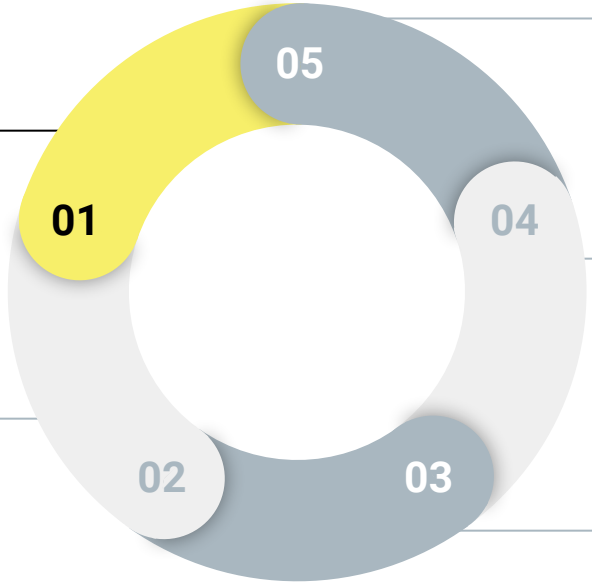


The Hybrid Kill Chain

A derivative of Lockheed Martin's cyber kill chain. The hybrid kill chain has been adapted to apply more directly to web server infrastructure.

The Hybrid Kill Chain

It includes the following stages:



Reconnaissance
Information gathered against a target

Weaponization
Preparation of offensive operations against specific targets using information gathered during reconnaissance.

Exfiltration
Ultimate goal. The exfiltration of private, sensitive data that the target considers critically sensitive.

Exploitation
Active compromise of adversary's apps, servers, or network, and averting physical, logical, or administrative controls.

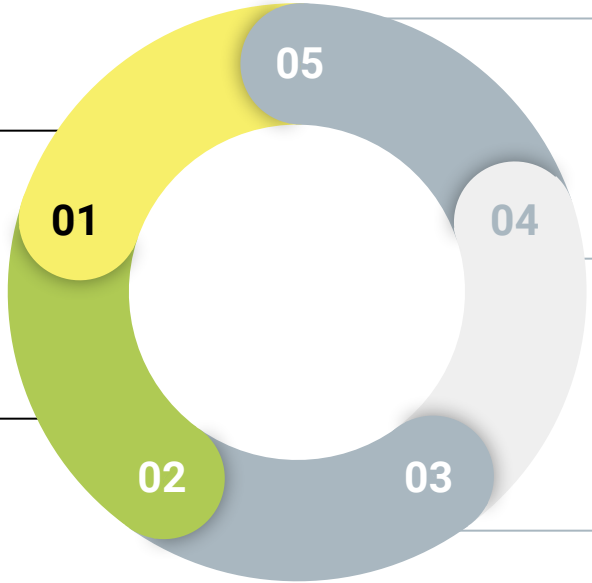
Delivery
Launch of the operation. Attacks carried out based on Red Team offensive strategies.

The Hybrid Kill Chain

It includes the following stages:

Reconnaissance
Information gathered against a target

Weaponization
Preparation of offensive operations against specific targets using information gathered during reconnaissance.



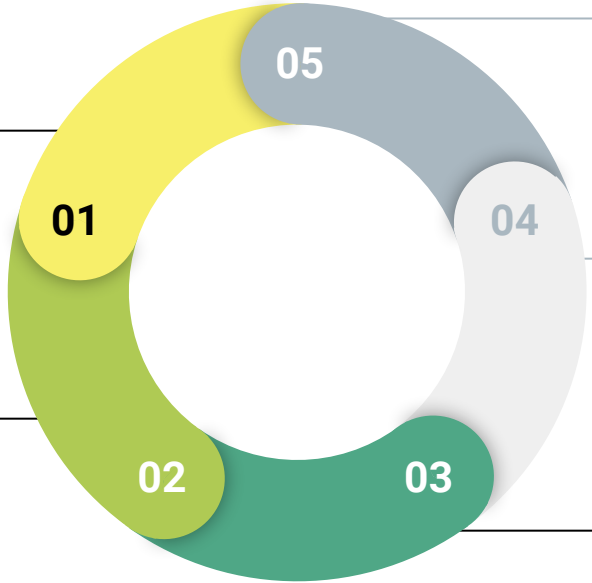
Exfiltration
Ultimate goal. The exfiltration of private, sensitive data that the target considers critically sensitive.

Exploitation
Active compromise of adversary's apps, servers, or network, and averting physical, logical, or administrative controls.

Delivery
Launch of the operation. Attacks carried out based on Red Team offensive strategies.

The Hybrid Kill Chain

It includes the following stages:



Reconnaissance
Information gathered against a target

Weaponization
Preparation of offensive operations against specific targets using information gathered during reconnaissance.

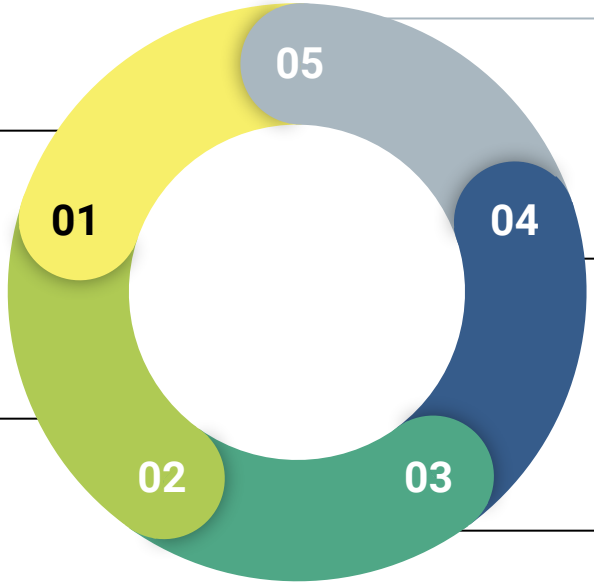
Exfiltration
Ultimate goal. The exfiltration of private, sensitive data that the target considers critically sensitive.

Exploitation
Active compromise of adversary's apps, servers, or network, and averting physical, logical, or administrative controls.

Delivery
Launch of the operation. Attacks carried out based on Red Team offensive strategies.

The Hybrid Kill Chain

It includes the following stages:



Reconnaissance
Information gathered against a target

Weaponization
Preparation of offensive operations against specific targets using information gathered during reconnaissance.

Exfiltration
Ultimate goal. The exfiltration of private, sensitive data that the target considers critically sensitive.

Exploitation
Active compromise of adversary's apps, servers, or network, and averting physical, logical, or administrative controls.

Delivery
Launch of the operation. Attacks carried out based on Red Team offensive strategies.

The Hybrid Kill Chain

It includes the following stages:

Reconnaissance
Information gathered against a target

Weaponization
Preparation of offensive operations against specific targets using information gathered during reconnaissance.

Exfiltration
Ultimate goal. The exfiltration of private, sensitive data that the target considers critically sensitive.

Exploitation
Active compromise of adversary's apps, servers, or network, and averting physical, logical, or administrative controls.

Delivery
Launch of the operation. Attacks carried out based on Red Team offensive strategies.

Web Vulnerabilities and the Business

Previously we learned that cybersecurity is often considered an obstacle for business operations. This is because enforcing good cybersecurity practices can cause production delays and increase budgets.

Web Vulnerabilities and the Business

Results of complacency and relaxed security to maximize profit can include:

01

A defaced web page containing malicious content or links to inappropriate sites, ultimately damaging a company's reputation.

02

A compromised web server used to download malicious software to anyone visiting the webpage.

03

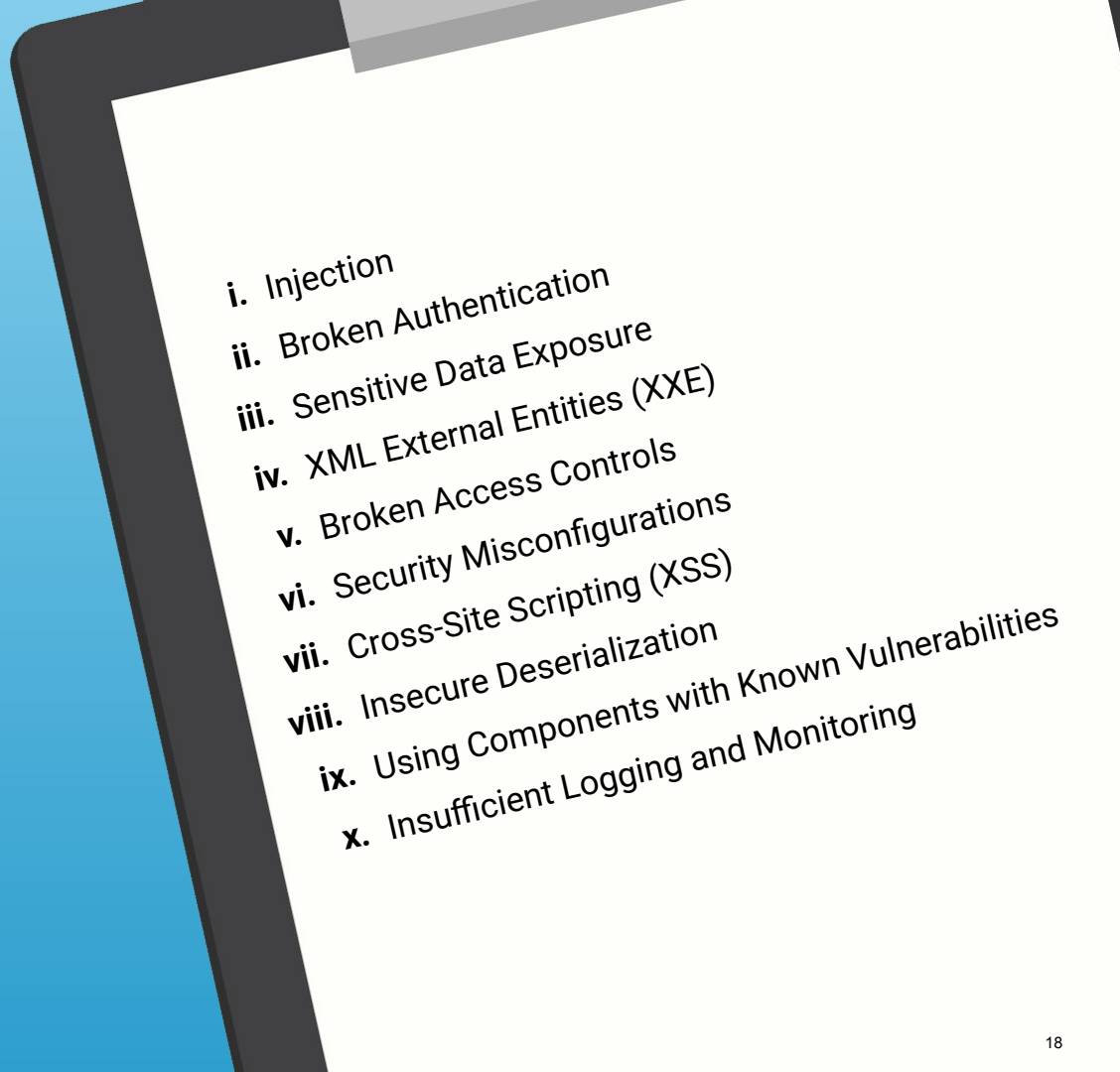
Compromised data used to commit fraudulent activities, leading to loss of business or lawsuits.

OWASP Top 10

The OWASP Top 10 is widely considered to represent the most prevalent security risks facing web applications today.

OWASP Top 10

The current list includes:

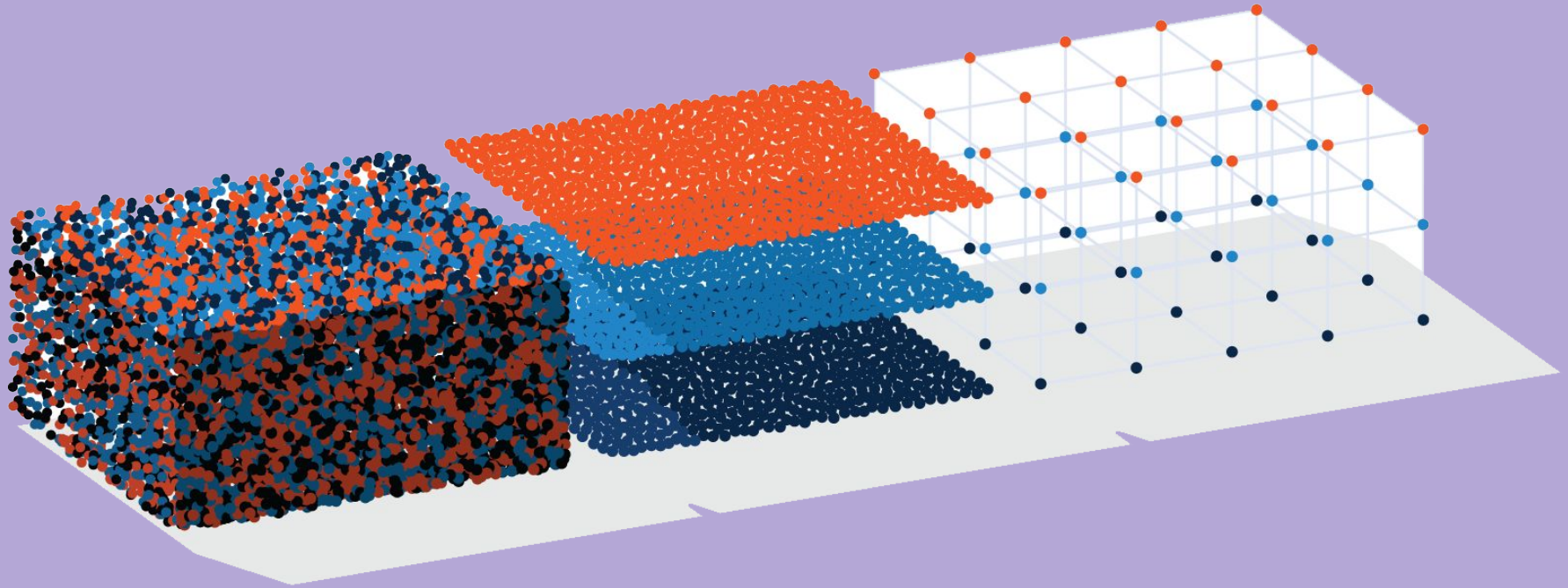


OWASP Top 10

Created to educate software developers, designers, architects, managers, and organizations about the consequences of web application security weaknesses.

OWASP Top 10

Developed through industry surveys completed by over 500 individuals from hundreds of organizations and over 100,000 real-world applications and APIs.

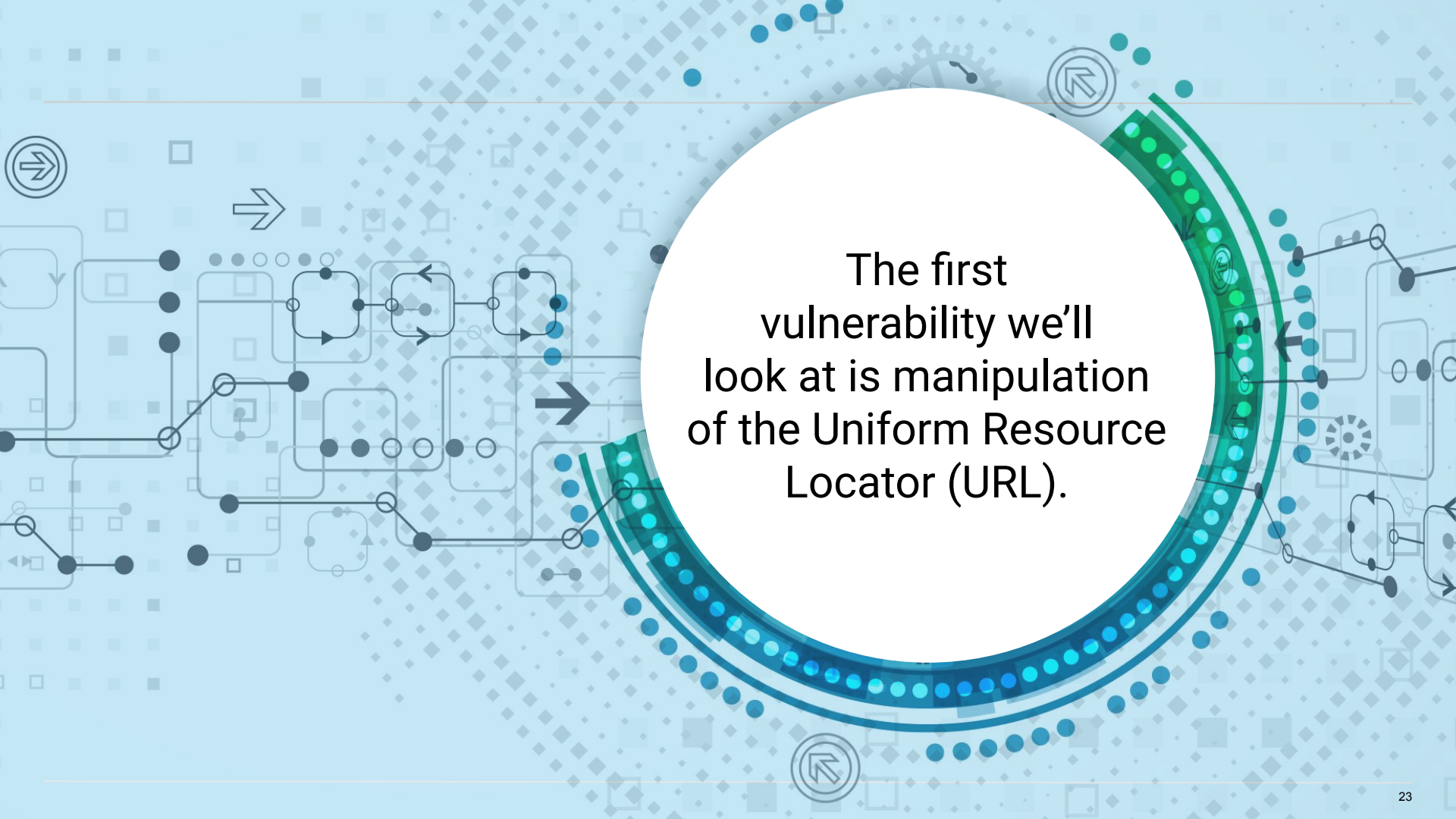


OWASP Top 10

Prioritized according to:

- Threat level prevalence data
- Consensus estimates of exploitability
- Detectability
- Impact

The URL Cruise Missile


The first vulnerability we'll look at is manipulation of the Uniform Resource Locator (URL).

The URL

URLs are the standardized naming convention for addressing documents that are accessible over the internet.

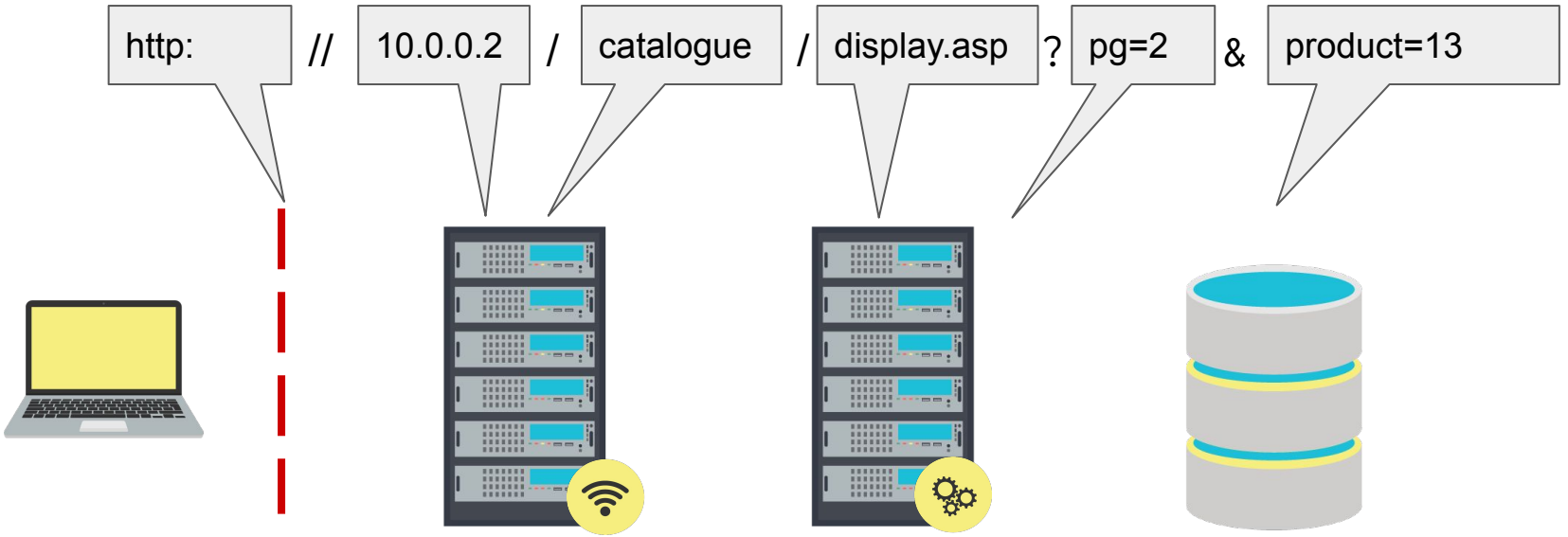
A web address is essentially a unique set of directions to an online resource.

The URL is the gateway to the web. But the URL can also be used as a weapon to launch attacks.

The URL Cruise Missile

The strength of a weaponized URL depends on the web vulnerability that it exploits. The more vulnerable the web client or server, the more dangerous the attack.

`http://10.0.0.2/catalogue/display.asp?pg=2&product=13`



The URL Cruise Missile

Various stages of a URL/URI

Protocol	Host Name	Path	Parameters
`http://`	`www.example.com`	`/add.asp`	`?item=3478`

URL Example: Protocol

Protocol indicates the protocol or application to use with the request, such as:

Protocol	Host Name	Path	Parameters
http://	www.example.com	/add.asp	?item=3478

HTTP (Hypertext Transfer Protocol)	For transferring web pages
FTP (File Transfer Protocol)	For file transfer requests (upload and download).
SPDY (Speedy)	Google's version of HTTP, designed to speed up web content loading.

URL Example: Host Name

Host Name identifies a specific web server for the request of web resources:

Protocol	Host Name	Path	Parameters
http://	www.example.com	/add.asp	?item=3478

Domain Name	Example.com, google.com, facebook.com, etc.
Sub-Domain	Typically used for a specific subsite within a larger domain. The most common subdomain is www, which stands for world wide web. Some domains use this as an indication of publicly accessible resources and content.

URL Example: Path

Path indicates which web application will provide resources to the client.

Protocol	Host Name	Path	Parameters
`http://`	`www.example.com`	`/add.asp`	`?item=3478`

`/add.asp`	is a directory, similar to a file or folder on your computer. Many websites, such as retailers, have a long hierarchy of categories such as: `/books/non-fiction/computers/tutorials/internet/`
`.asp` Active Server Page	It is a technology that enables designers to produce dynamic and interactive web pages.

URL Example: Parameters

Parameters are specifically formatted data that interact with back-end servers such as email and web databases.

Protocol	Host Name	Path	**Parameters**
`http://`	`www.example.com`	`/add.asp`	`?item=3478`

`?item=3478`	Example of a parameter encoding specific data into the URL for use by the server. In this case, the client-side user is requesting an item with ID 3478.
Each parameter is made up of a few different parts	• The question mark (`?`) indicates the beginning of a list of parameters. • Each individual parameter has a name and a value, separated by a hash (`#`). The name of this parameter is item and the value is `3478`. • A URL can have multiple parameters. These will be separated by an ampersand symbol (`&`). For example: `?item=3478&price=299`

The URL is composed of several different layers designed to access various parts of a website: web pages, emails, or bank account information, for example.

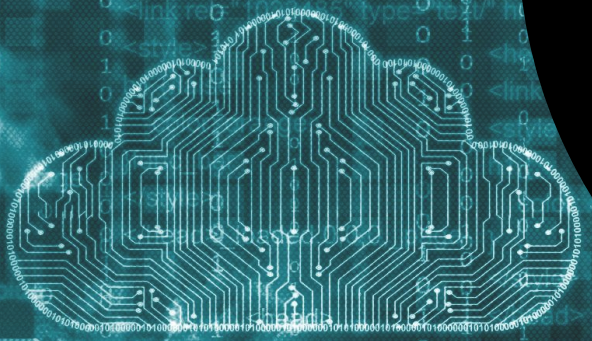
Attackers can exploit weak web server infrastructure to manipulate URLs and infiltrate these various parts of the web server architecture.

Web Server Infrastructure

Web Infrastructure

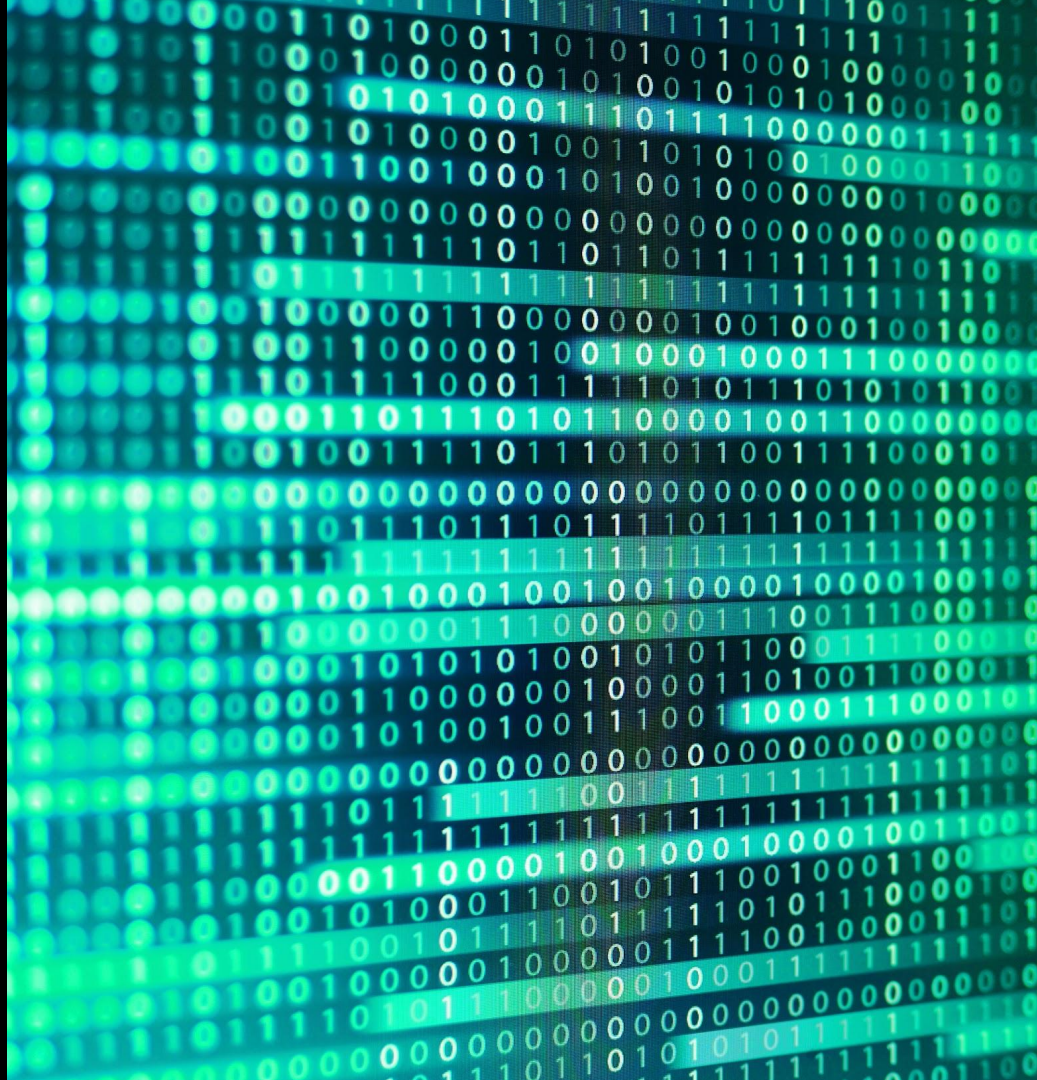
In the early days of the internet, web pages were mostly static and had long load times. Since then, the internet has made a drastic shift towards user experience, which includes instant response times.

This shift meant visually appealing web applications also had to be supported.



- The advancements that have improved user experience for web application users are the same ones that criminal hackers exploit.

- In order to harden such systems, Security administrators need to have a basic understanding of standard website infrastructure and its sub-components.



Components of Web Infrastructure

Typical web server infrastructure primarily consists of five components.

01 **Client**

02 **Firewall**

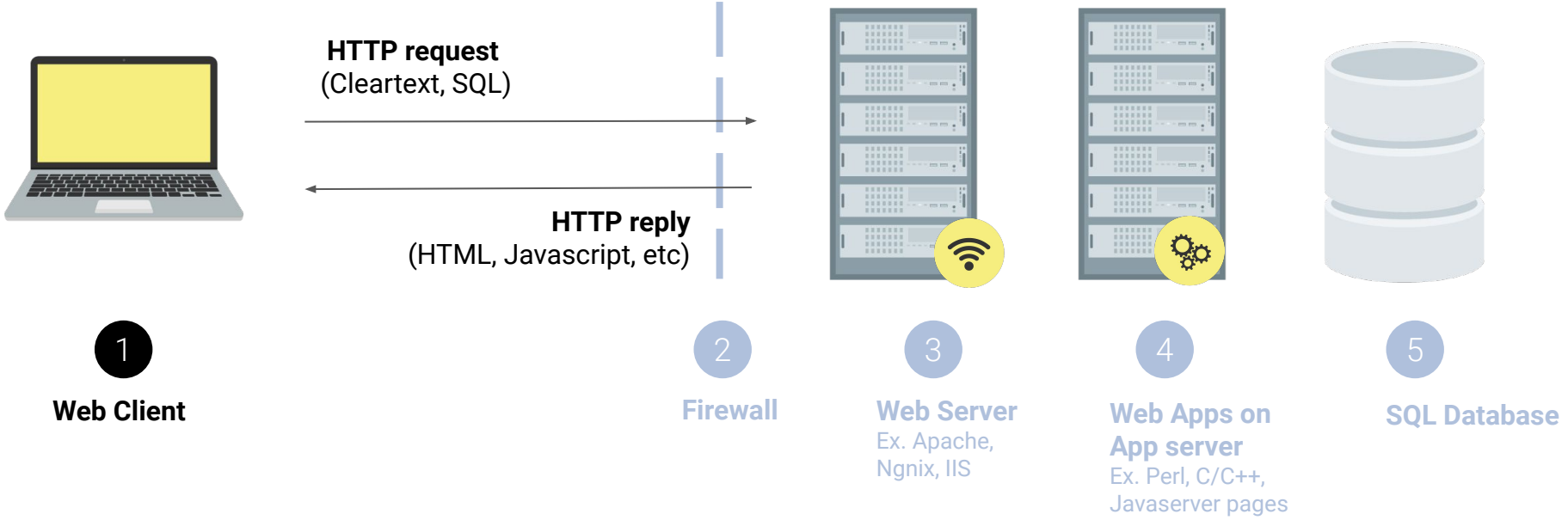
03 **Web Server**

04 **Web Application**

05 **Database**

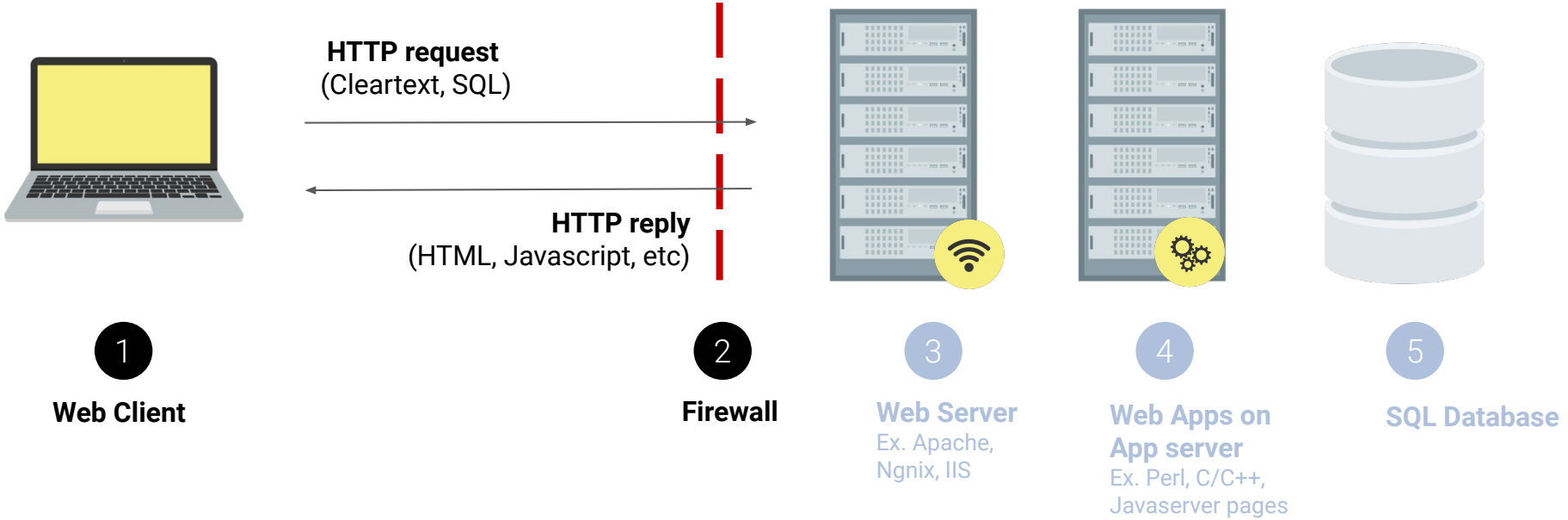
Components of Web Infrastructure: Client

A user who interacts with a web server using HTTP or FTP through either a web browser or file transfer software.



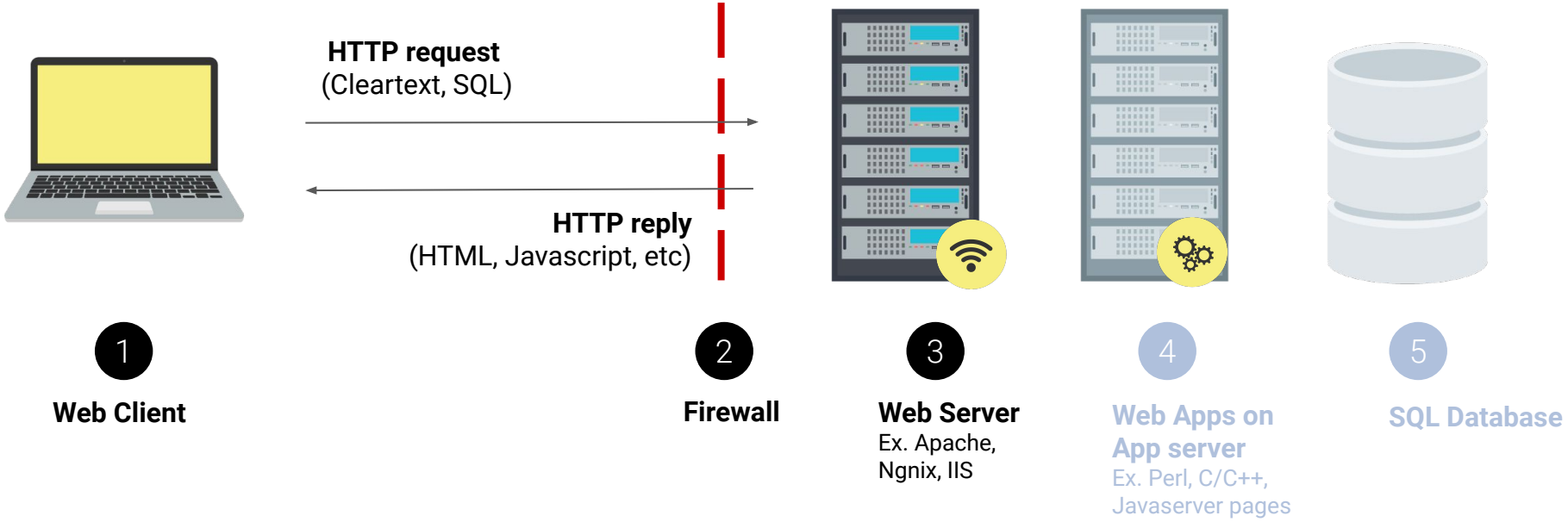
Components of Web Infrastructure: Firewall

A perimeter defense used to protect the web server sitting behind it.



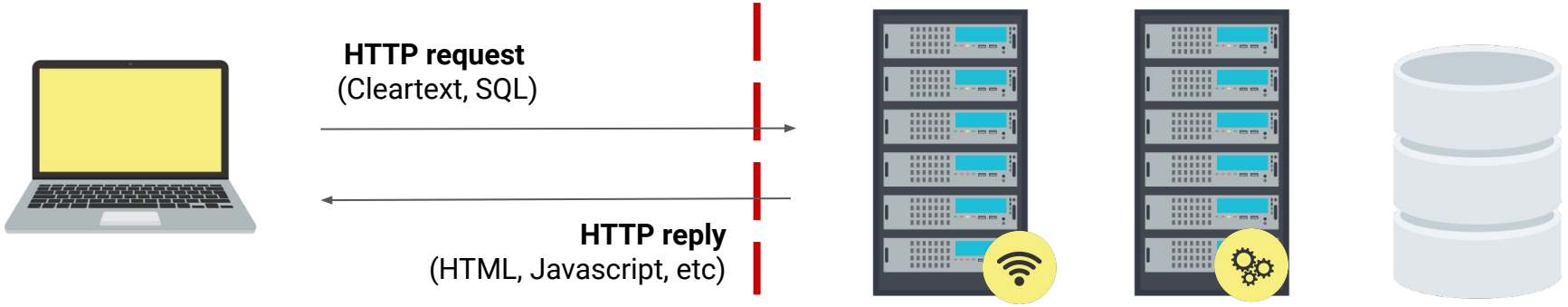
Components of Web Infrastructure: Web Server

A program, such as Apache, Nginx, or IIS, that responds to a client's requests for resources.



Components of Web Infrastructure: Web Application

The software that runs on a remote server, such as Facebook, Twitter, Amazon.



1 **Web Client**

2 **Firewall**

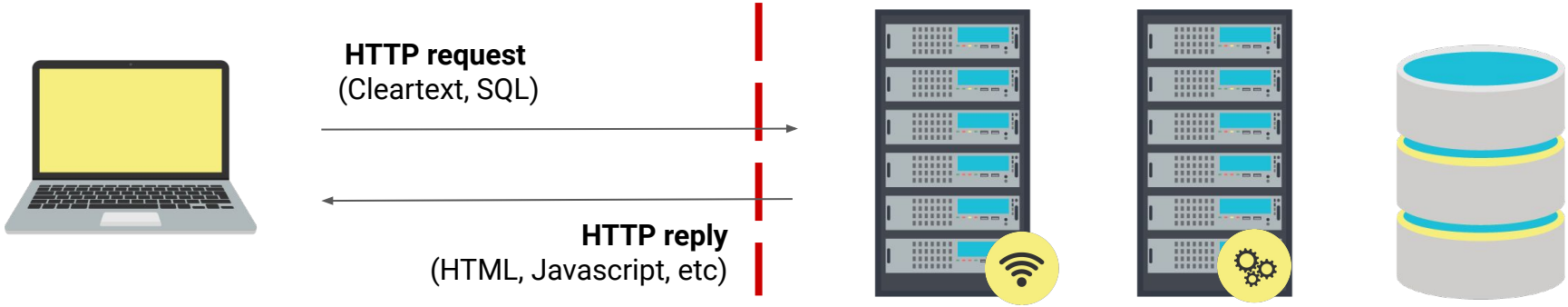
3 **Web Server**
Ex. Apache, Ngnix, IIS

4 **Web Apps on App server**
Ex. Perl, C/C++, Javaserver pages

5 SQL Database

Components of Web Infrastructure: Database

Typically the innermost part of the web architecture, storing data like customer names, addresses, account numbers, and credit card info.



Web Infrastructure Security

Securing web infrastructure includes some of the following processes:

01 Input Sanitation

The process of cleansing and scrubbing user input in order to prevent it from exploiting security holes. This is ensured by, when necessary, changing the value input by the user.

02 Input Validation

The testing of input supplied by a user or application, designed to prevent malformed data from entering a data information system. This is done by verifying user input meets specific criteria.

03 Secure Software Deployment Cycles (SDLC)

A software development methodology that ensures secure programming occurs at every stage of the software development process.

Web Vulnerabilities

SDLC faces the following challenges, giving way to vulnerabilities:

- High Implementation costs
- Insufficient support from management
- Insufficient standardization
- No quality management
- Reactive security postures (*If it ain't broke, don't fix it)*
- Total reliance on web application firewalls

Popular Web Servers and Associated Vulnerabilities

Internet Information Server (IIS) is a general-purpose web server from Microsoft that runs on Windows systems and serves HTML pages and files to web clients.

Associated Web Vulnerability	Description	National Vulnerability Database Reference
Cross-Site Scripting (XSS)	Allows remote attackers to alter a URL with a malicious script that will redirect a message or request.	CVE-2003-0223
Directory Traversal	Allows remote attackers to view source code and determine the existence of arbitrary files via a hex-encoded `%c0%ae%c0%ae` string, which is the Unicode representation for ".." (dot dot).	CVE-2002-1744
Denial of Service	Buffer overflow of the data transfer mechanism in IIS allows remote attackers to cause a denial of service or execute code.	CVE-2002-0147

Popular Web Servers and Associated Vulnerabilities

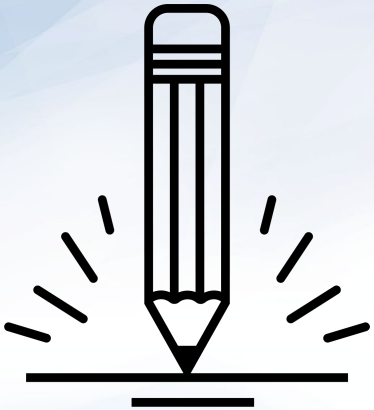
Apache is an open source web server alternative that runs on UNIX, Linux, and Windows.

Associated Web Vulnerability	Description	National Vulnerability Database Reference
OpenMeetings SQL Injection	Allows for modification of the structure of existing queries, resulting in the exfiltration of queries being made by the back-end application.	CVE-2017-7681
Apache Ranger Security Bypass	Any characters after a wildcard symbol (*) is ignored, resulting in unintended behavior.	CVE-2017-7676
Apache HTTP Server Authentication Bypass	The use of the string `ap_get_basic_auth_pw()` by third-party modules can bypass authentication requirements.	CVE-2017-3167

Popular Web Servers and Associated Vulnerabilities

Nginx is an open-source, less popular alternative to Apache that runs on Linux.

Associated Web Vulnerability	Description	National Vulnerability Database Reference
Nginx SPDY Heap Buffer Overflow	Allows remote attackers to craft requests that execute arbitrary code.	CVE-2014-0133
Nginx Root Privilege Escalation Vulnerability	Allows local users to gain root privileges.	CVE-2016-1247
Remote Integer Overflow Vulnerability	Sensitive data can be leaked with crafted requests that exploit a range filter module in certain Nginx versions.	CVE-2017-7529



Activity: URL Cruise Missile

In this activity, you will strengthen your knowledge of concepts related to the URL to understand how the URL can be manipulated to act as a weapon against web server infrastructure.

Suggested Time:
15 Minutes

Time’s Up! Let’s Review.

Client-Side and Server-Side Attacks

In today’s globally connected cyber community, network- and OS-level attacks are well defended through the proper deployment of technical security controls like firewalls, IDS, data loss prevention, and endpoint security.

However, web servers are accessible from anywhere on the web, making them vulnerable to attack.

Client and Server Response Refresher

How responses and requests travel through web infrastructure:

01

Client: A user generates an HTTP request from their web browser using a URL or web address, such as: http://example.com/add.asp?ItemID=123&Price=999

02

Web Server: The user's HTTP request arrives at the front-end web server as directed by the domain and processes the following portion the URL. (http://example.com)

03

Web Application: The user's HTTP request invokes the application service from the application server as directed by the path. (/add.asp)

04

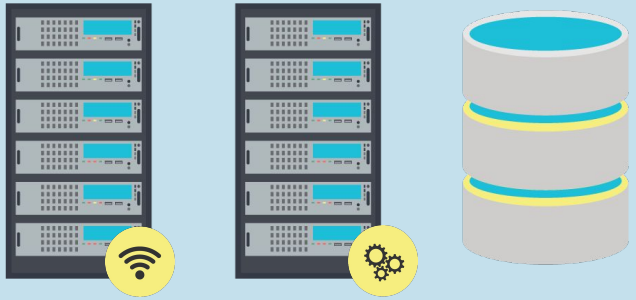
Database Server: The user's HTTP request invokes the database server and performs a query in search of a specific item number and its price. (?ItemID=123&Price=999)

Client-Side and Server-Side Attacks

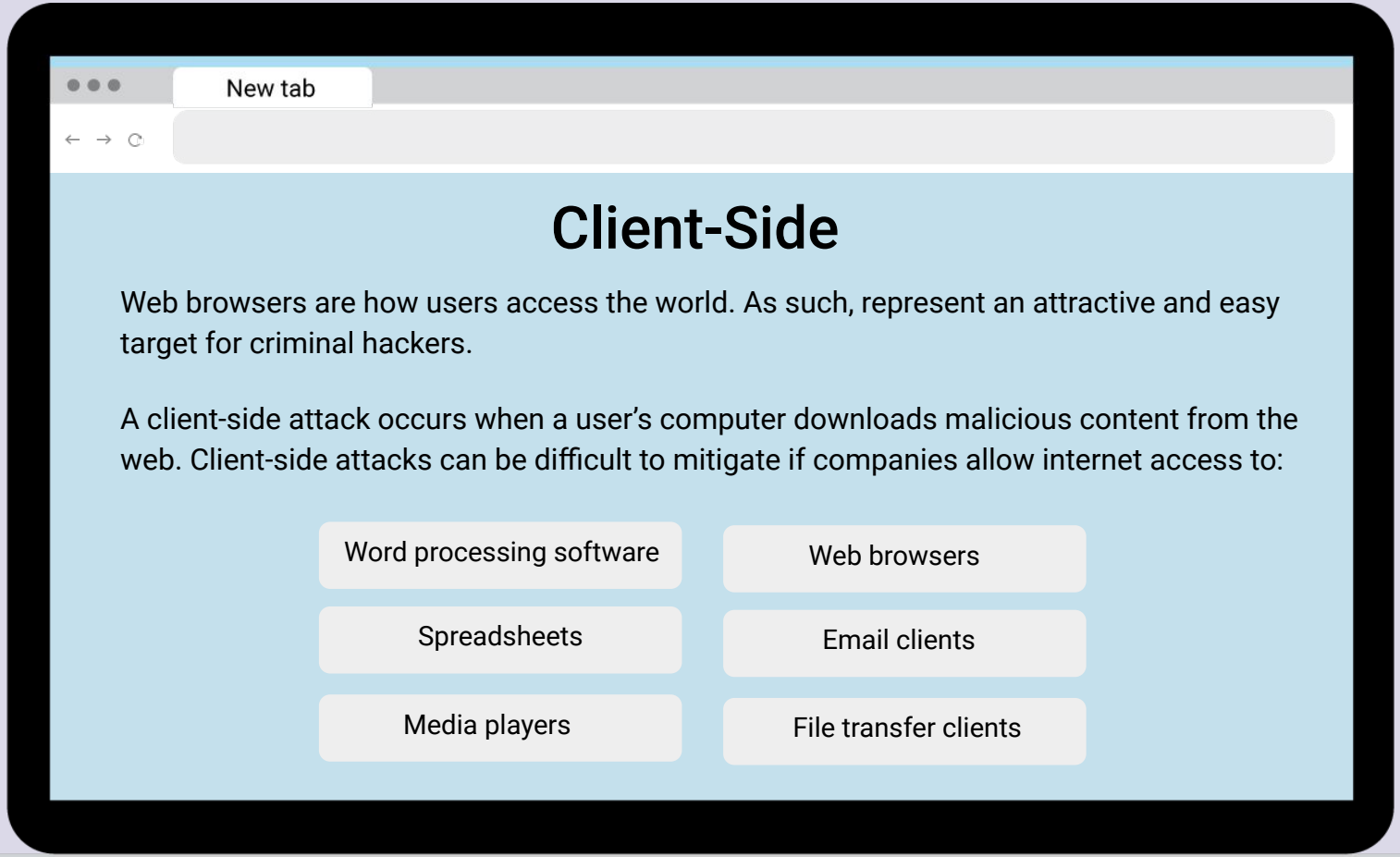
Client-side attacks

Target software installed on the desktop, such as web browsers, email clients, media players, and office suites.

Server-side attacks



Seek to compromise and breach applications and data on a server.



Client-Side

Web browsers are how users access the world. As such, represent an attractive and easy target for criminal hackers.

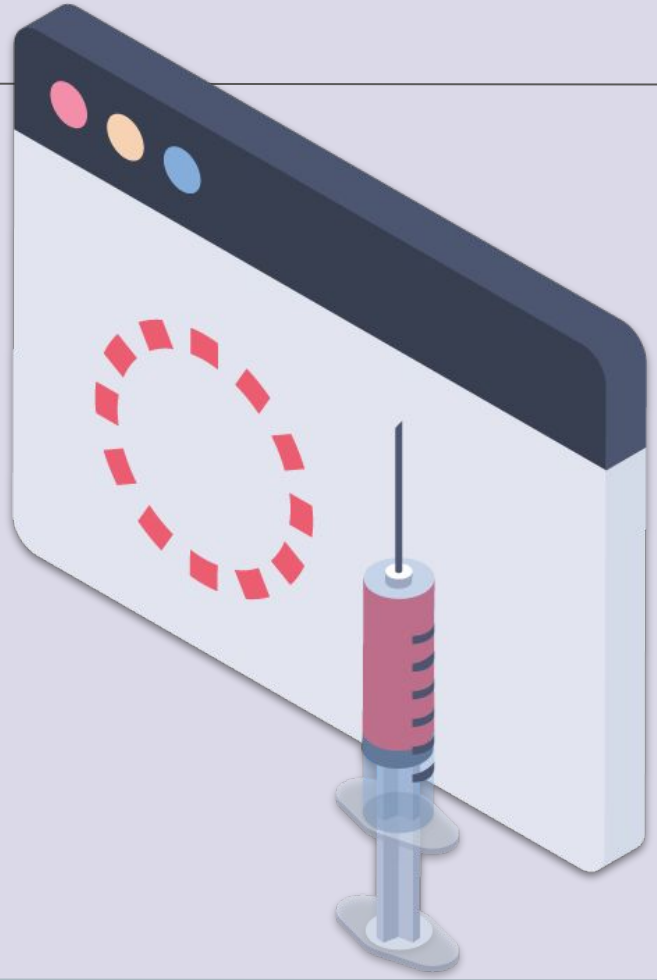
A client-side attack occurs when a user's computer downloads malicious content from the web. Client-side attacks can be difficult to mitigate if companies allow internet access to:

- Word processing software
- Web browsers
- Spreadsheets
- Email clients
- Media players
- File transfer clients

Prevalent Client Side Attacks

Firewalls often fail to prevent client-side attacks, which occur behind the firewall and from within the local network.

Cross-site scripting (XSS) allows attackers to **inject malicious code into a website** in order to intercept user sessions, vandalize websites, steal data, and control a user's browser.



Prevalent Client Side Attacks

Firewalls often fail to prevent client-side attacks, which occur behind the firewall and from within the local network.

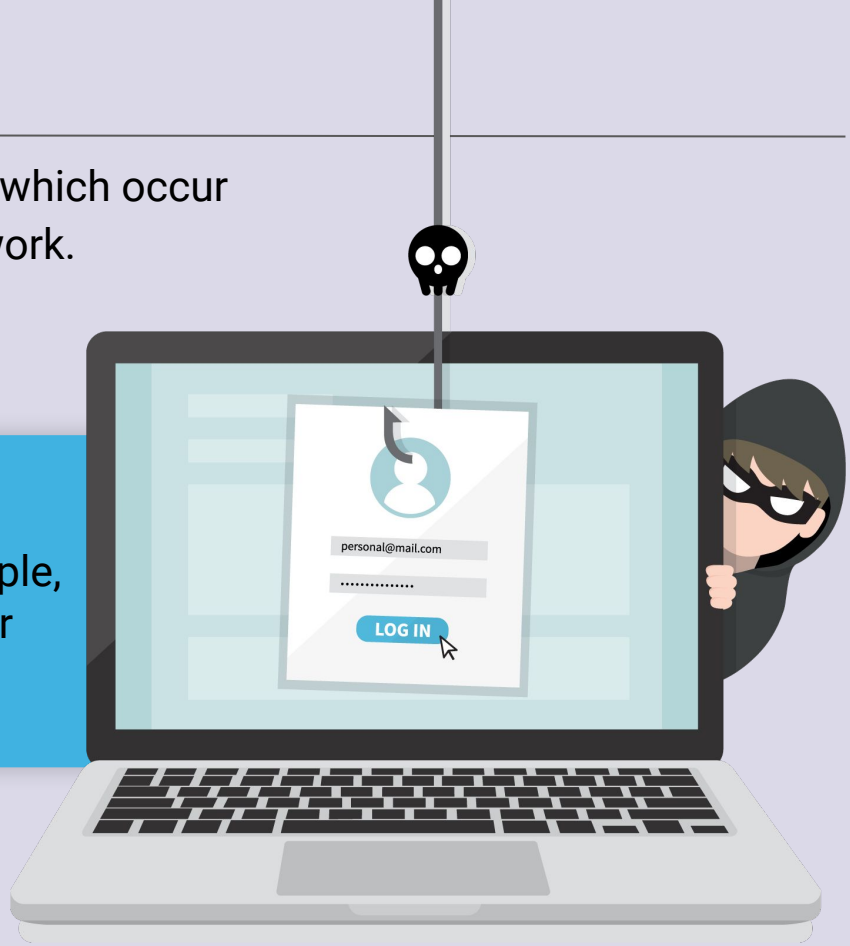
Clickjacking tricks users into clicking misleading graphics that will then trigger an exploit.

For example: A big fake Download button.

Prevalent Client Side Attacks

Firewalls often fail to prevent client-side attacks, which occur behind the firewall and from within the local network.

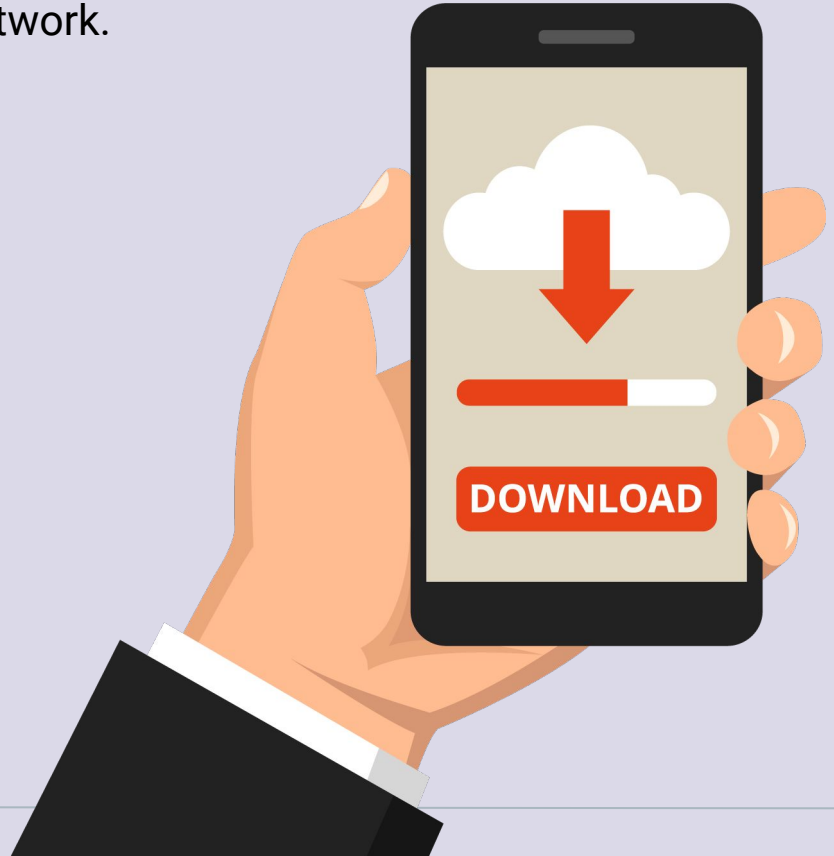
Content Spoofing tricks a user into believing that **a certain website is legitimate**. For example, a fake social media login page that steals your credentials after you submit them.



Prevalent Client Side Attacks

Firewalls often fail to prevent client-side attacks, which occur behind the firewall and from within the local network.

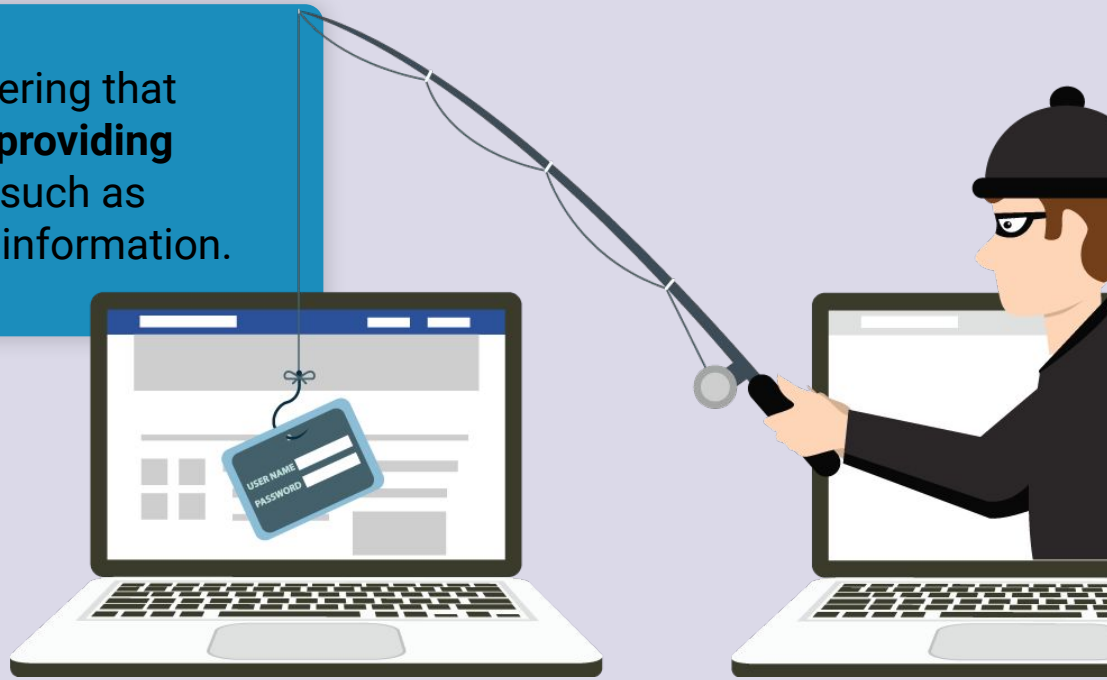
Drive-By Download are downloads triggered without the users knowledge upon visiting a webpage.



Prevalent Client Side Attacks

Firewalls often fail to prevent client-side attacks, which occur behind the firewall and from within the local network.

Phishing is a form of social engineering that **manipulates a computer user** into **providing personal confidential information**, such as user credentials and bank account information.



Server-Side Attacks

Attackers typically target the following software vulnerabilities and configuration errors in order to compromise a web server:

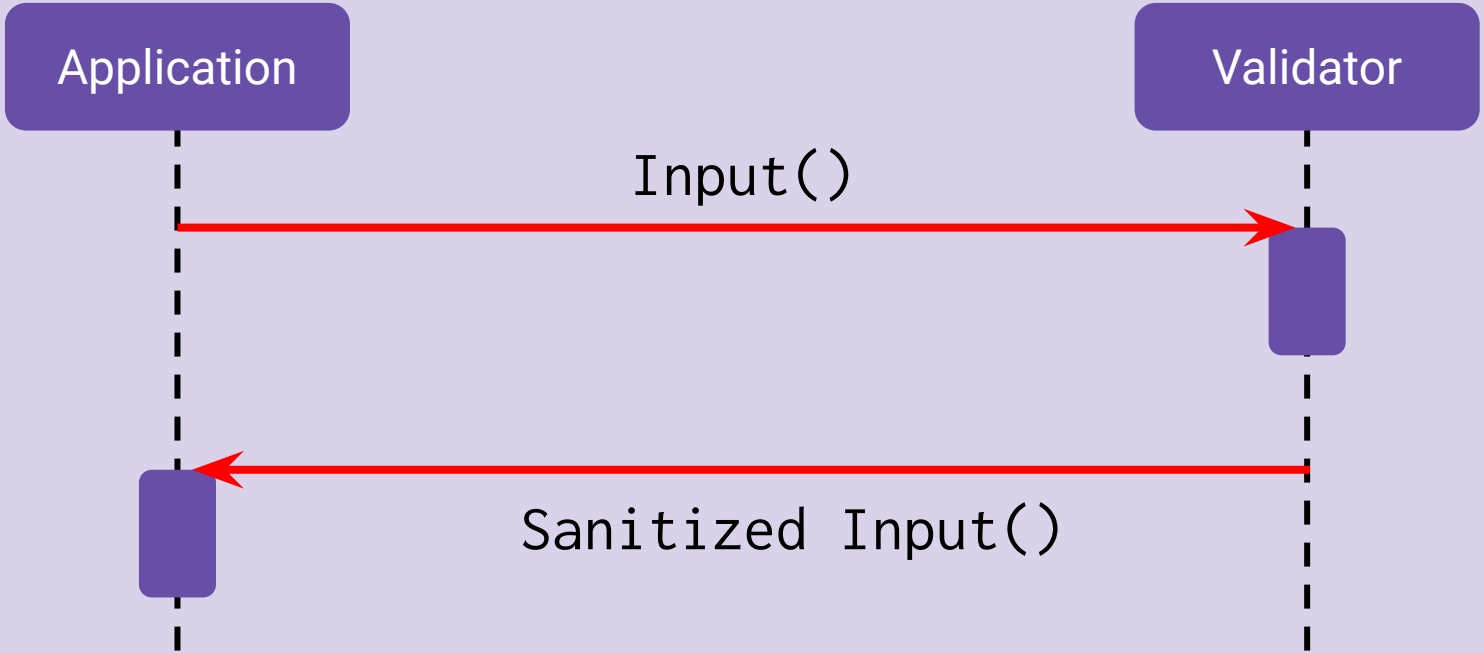
- Default accounts that contain easily acquired default usernames and passwords.
- Misconfigurations within a web server, web app, network, or operating system.
- Bugs in a web server from unpatched systems.
- Unnecessary services that lead to a larger attack service.
- Default settings on installed web server and apps that remain unchanged.
- Security conflicts with business cases and lack of budget.
- Improper permissions allowing overly lenient access to directory and file permissions.

Prevalent Server-Side Attacks

Website Defacement	HTTP Response Splitting	Web Cache Poisoning	Parameter or URL Tampering	Path or Directory Traversal (dot-dot-slash attack)
An attack against a website that alters its appearance or the information it contains.	The server does not properly sanitize input values, such as character returns (CRs) and line feeds (LFs), allowing for attacks such as cross-site scripting.	Involves replacing legitimate cached web pages with malicious content.	Manipulation of parameters in a URL passed to a web server.	Used to navigate into files and directories with dot-dot-slash (. . /) as if navigating through directories in a terminal, with the aim of reaching folders outside of the root directory.

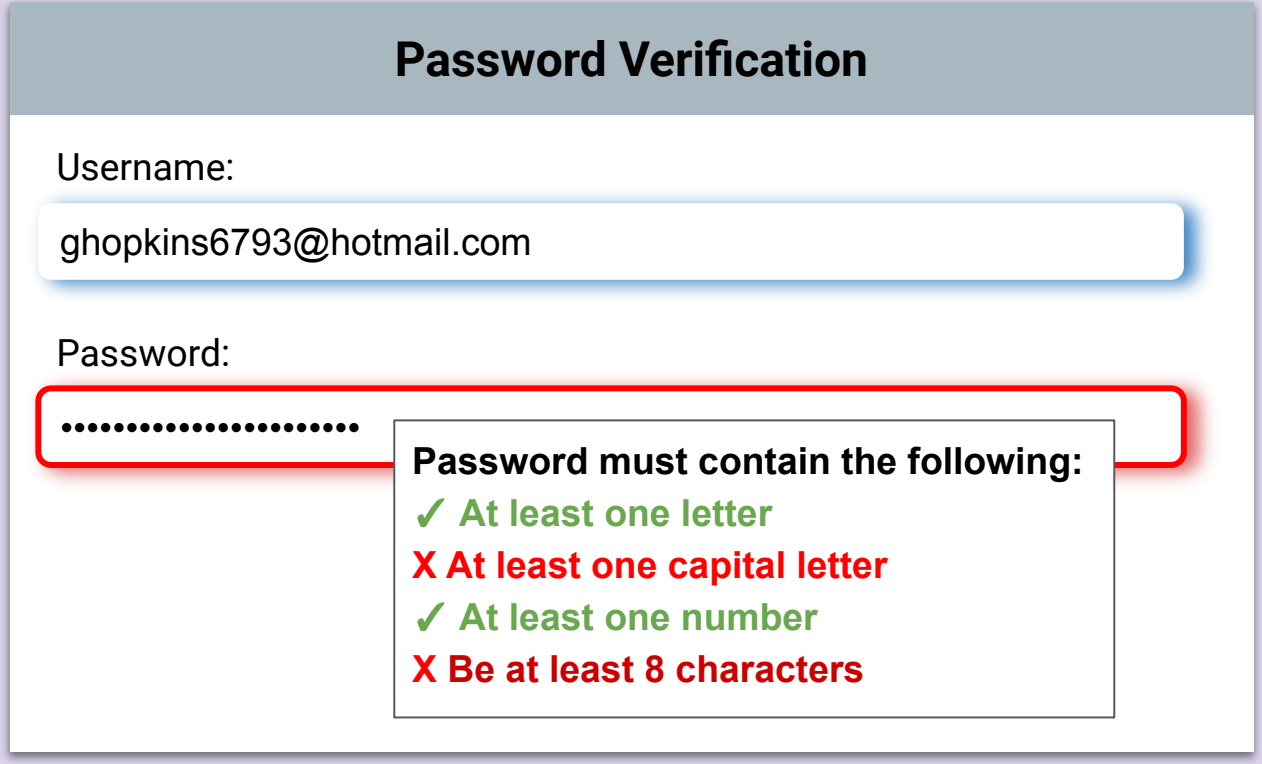
Risk Mitigation: Input sanitization

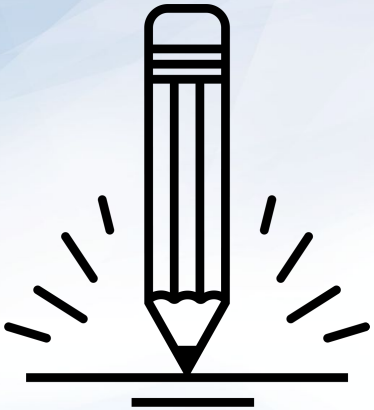
Web application (preferably server-side) actively modifies user input to an acceptable format or blocks the user's request.



Risk Mitigation: Input validation

Checks that user input is in an acceptable format.





Activity: Client vs. Server Side Attacks

In this activity, you will explore the differences between client- and server-side attacks.

Suggested Time:
15 Minutes

Time’s Up! Let’s Review.

Countdown timer

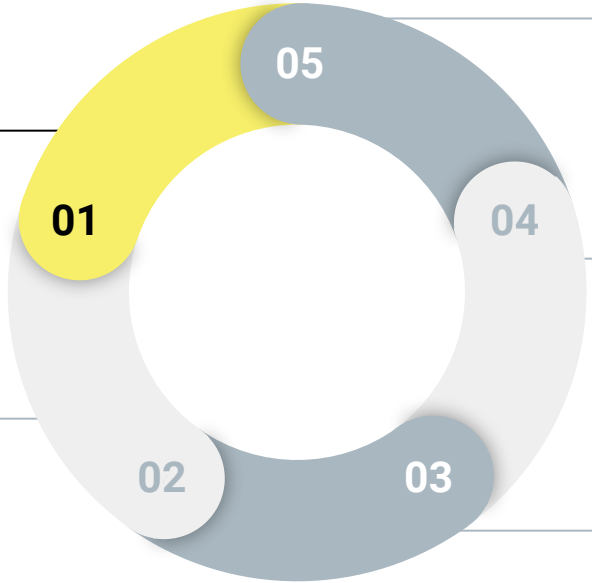
15:00

(with alarm)

Reconnaissance and Information Gathering

The Hybrid Kill Chain

It includes the following stages:



Reconnaissance
Information gathered against a target

Weaponization
Preparation of offensive operations against specific targets using information gathered during reconnaissance.

Exfiltration
Ultimate goal. The exfiltration of private, sensitive data that the target considers critically sensitive.

Exploitation
Active compromise of adversary's apps, servers, or network, and averting physical, logical, or administrative controls.

Delivery
Launch of the operation. Attacks carried out based on Red Team offensive strategies.

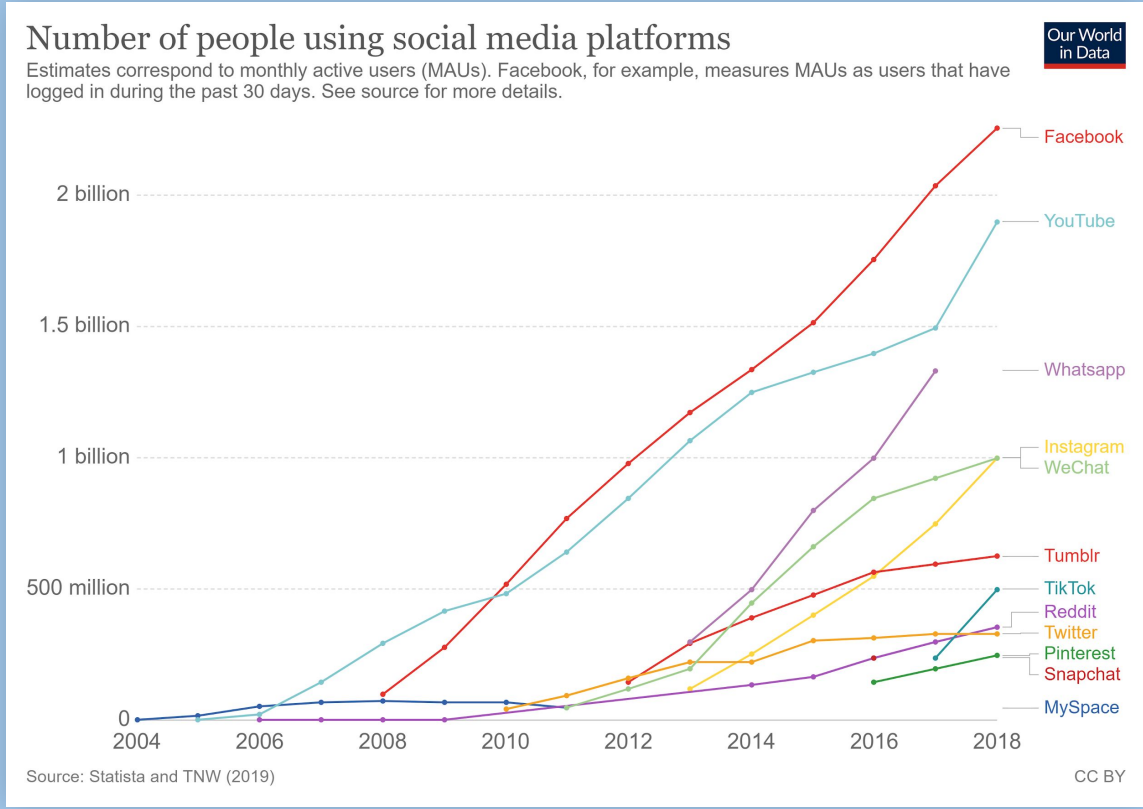
Reconnaissance

Facebook is the most popular online social media platform used today.

It can be thought of as an information wholesaler, making it incredibly desirable to attackers.

Reconnaissance

While not on the list, LinkedIn and its 500+ million users provide a wealth of information that can be used against employers and companies.



Reconnaissance

Social media has proven a profitable and effective way to build communities and engage customers. But to securely operate social media accounts, businesses must train employees to ensure they **do not**:

					
Engage with suspicious posts.	Share passwords.	Click on ads.	Use social media on public Wifi.	Use the same password for extended periods of time.	Follow accounts or people you don't know or haven't vetted.

Social Engineering

The attack surface is amplified for attackers who cross-reference information gathered from LinkedIn and Facebook, allowing them to carry out various social engineering attacks.

LinkedIn

- Types of hardware and software
- used by a company
- First and last names of employees
- Employee position information
- Length of employment
- Prior work history
- Education
- Skills and endorsements
- Previous projects

Facebook

- Names of friends and family
- Favorite hobbies
- Vacation spots
- Favorite books and movies
- Favorite foods, drinks and restaurants

Hackers also use more advanced forms of information gathering.

Web crawlers

Web crawlers, also known as spiders and spiderbots, index sites to help search engines find content.

Web crawlers: `robots.txt` file

01

To hide information from web crawlers, such as sensitive web server information, website owners can use the `robots.txt` file.

02

But excluding directories and pages indicates that they likely contain critical, sensitive data. This makes them attractive targets.

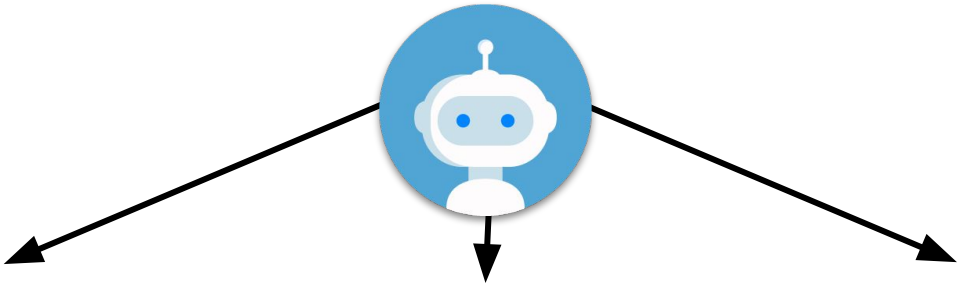
03

Experienced criminal hackers will attempt to harvest the `robots.txt` file to retrieve private data, such as content management system information and root directory structure.

robots.txt

A bot wants to visit the following URL: http://www.example.com/welcome.html

Prior to visiting the URL, it first checks http://www.example.com/robots.txt
It might receive any of the following responses:



```
User-agent: *
Disallow: /
```

Excludes all bots (indicated by the * wildcard symbol) from the entire server as indicated by disallowing access to the / root directory

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /tmp/
Disallow: /user/
```

Excludes all bots from specific directories.

```
User-agent: BadBot
Disallow: /
```

Excludes a single bot (BadBot) from the entire server.

Bots

Two important security implications that organizations must consider when using `/robots.txt` files:

01 Malicious Bots

Bots can ignore your `robots.txt` file. Especially malware robots, which scan the web searching for security vulnerabilities.

02 “Don’t Look Here!”

`robots.txt` is available to the public.

Be aware that files that have been disallowed may be seen as opportunities to find important sensitive data. Otherwise they wouldn’t have been disallowed in the first place.

WHOIS

Another effective reconnaissance tool is the WHOIS protocol. WHOIS is a query and response protocol used for querying the WHOIS database.

As you may recall from earlier units, a WHOIS database stores registered user information including IP address blocks, domain names, email and street addresses, and phone numbers.

Whois Demo

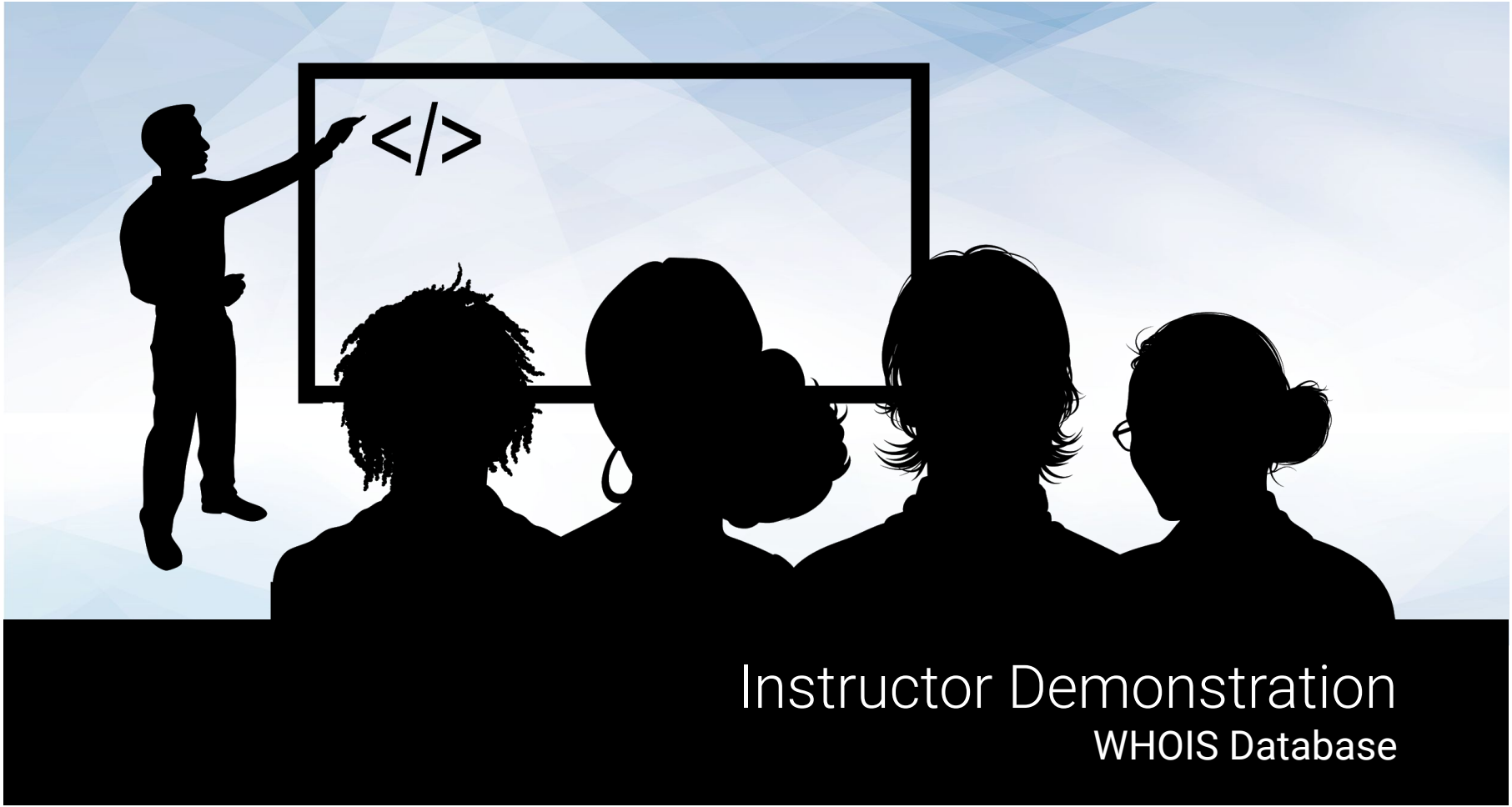
We'll demonstrate using the WHOIS database with the following scenario:

A hacktivist is trying to infiltrate the local newspaper's network, The Sacramento Bee, in order to deface their home web page.

The hacktivist is upset about a recently published op-ed article that favors a view they oppose.

They've decided to produce a phishing email to collect personal data from the newspaper's website registrant by falsely stating that their domain name
will be taken offline due to non-payment.

We will use a WHOIS registration database query to find registered user information, such as emails and phone numbers.

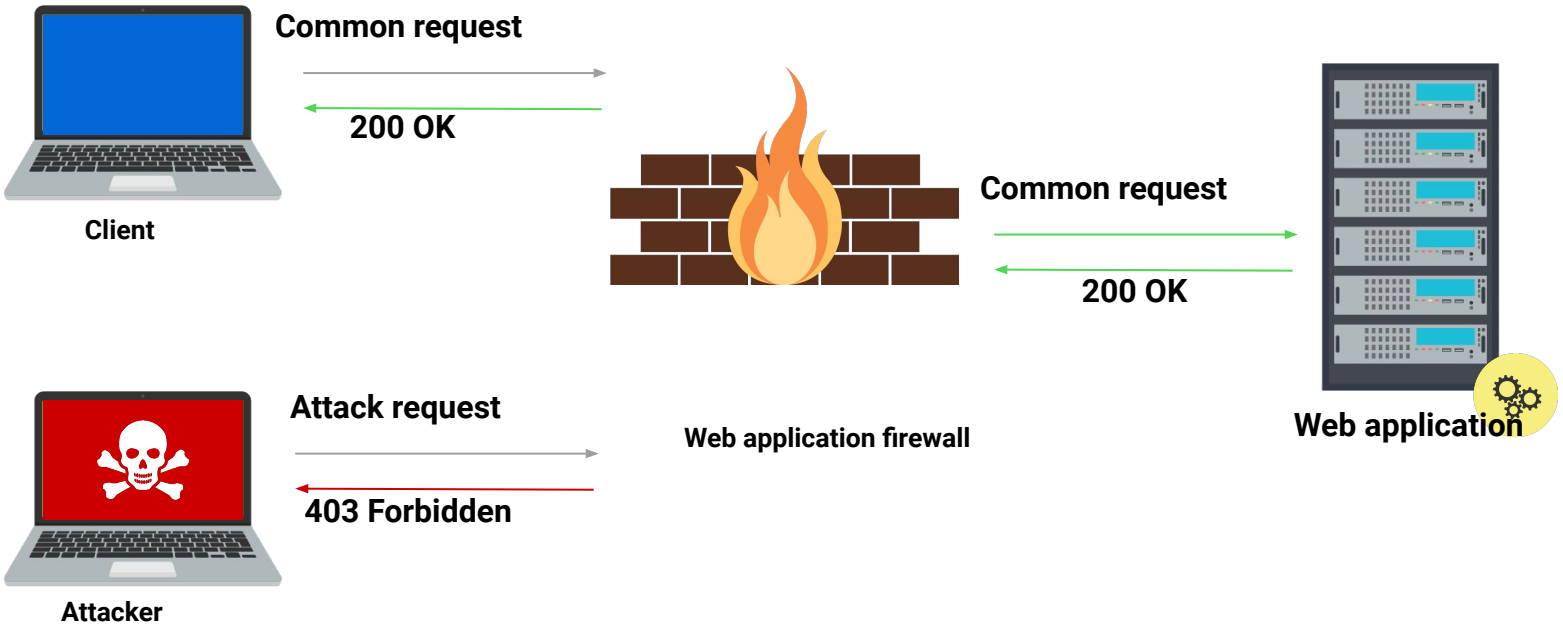


Instructor Demonstration

WHOIS Database

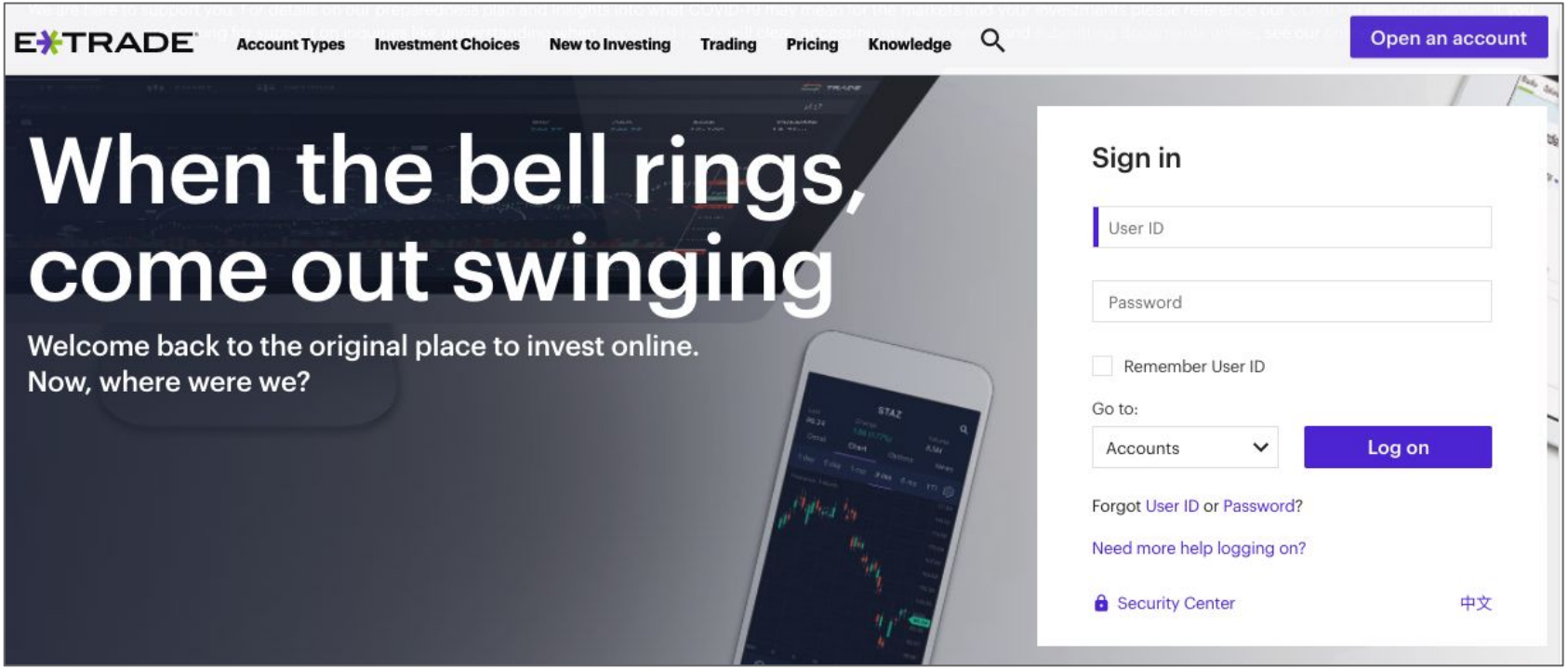
Web Application Firewalls

Web application firewalls (WAFs) are designed to defend against different types of HTTP attacks and various query types, such as SQLi and XSS.



Web Application Firewalls

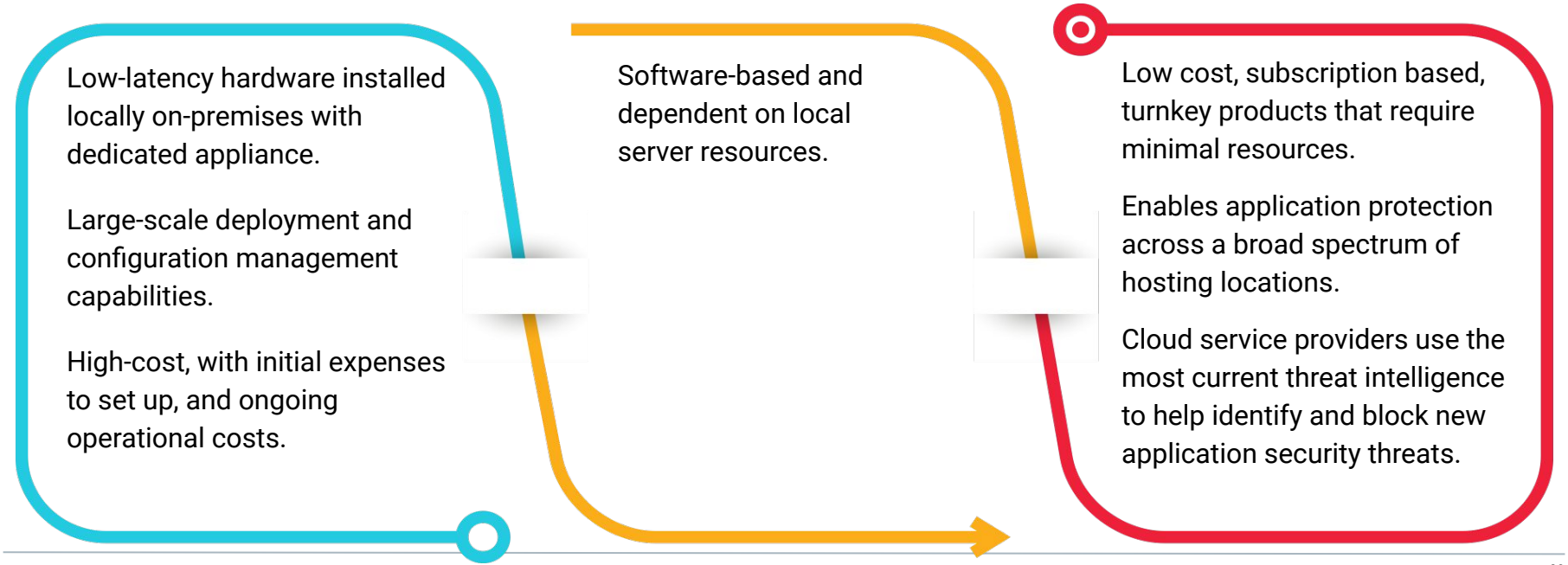
WAFs are typically present on web sites that use strict transport security mechanisms, such as online banking or ecommerce websites.



WAFs can provide adversaries crucial information regarding the security posture of an organization’s web infrastructure.

WAF Deployment

WAFs have three deployment strategies, each with their own sets of advantages and disadvantages.



WAF Analysis and Filtering

WAFs analyze and filter traffic based on three different techniques:

01 Allow lists

Only accepts traffic from sources that are known and trusted.

- Less-resource intensive than deny lists.
- May block benign traffic unintentionally.

02 Deny lists

Uses preset signatures to block malicious web traffic.

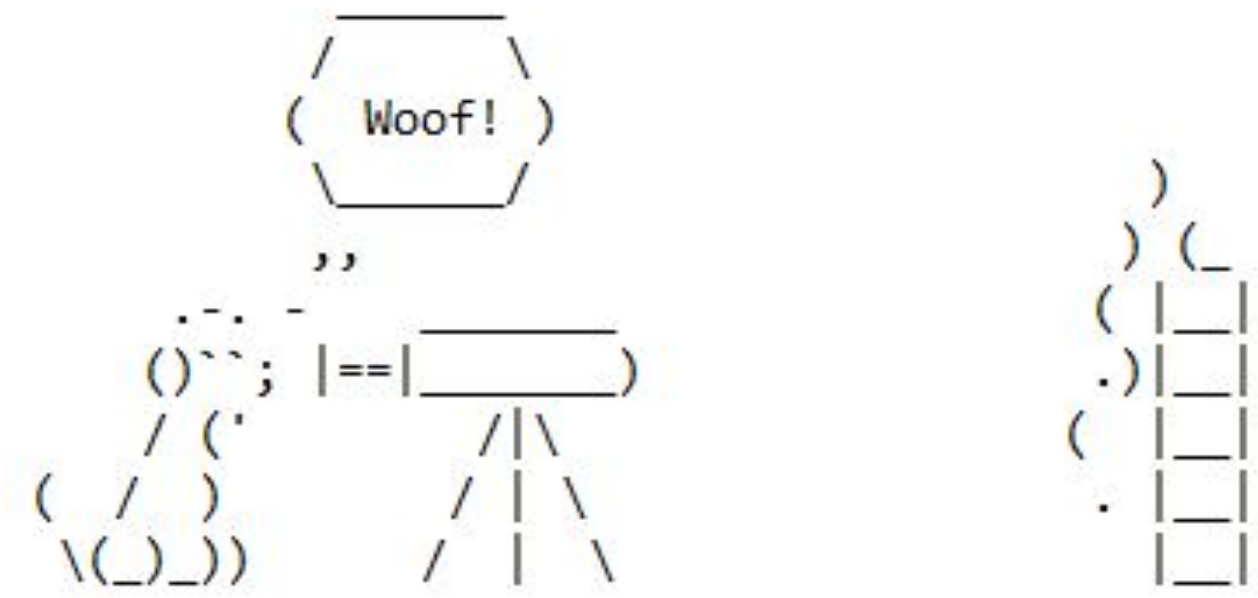
- Useful for public websites and web applications that receive lots of traffic from unknown IP addresses.
- Resource-intensive due to packet filtering based on a set of specific characteristics as opposed to IP addresses.

03 Hybrid

Uses a combination of allow and deny lists.

Wafw00f

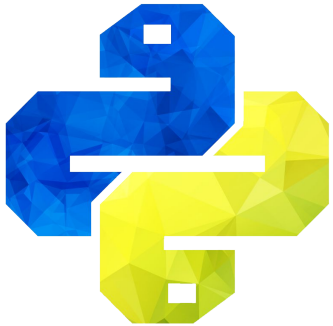
`Wafw00f` is an open source command-line WAF utility focused on web-based attacks that occur at the application `layer.afw00f`



Wafw00f

01 Python

It is written in Python and automates a carefully crafted set of procedures to determine if a website sits behind a web application firewall.



02 Vulnerabilities

Although WAFs are used to harden web server infrastructure, they do come with vulnerabilities of their own.

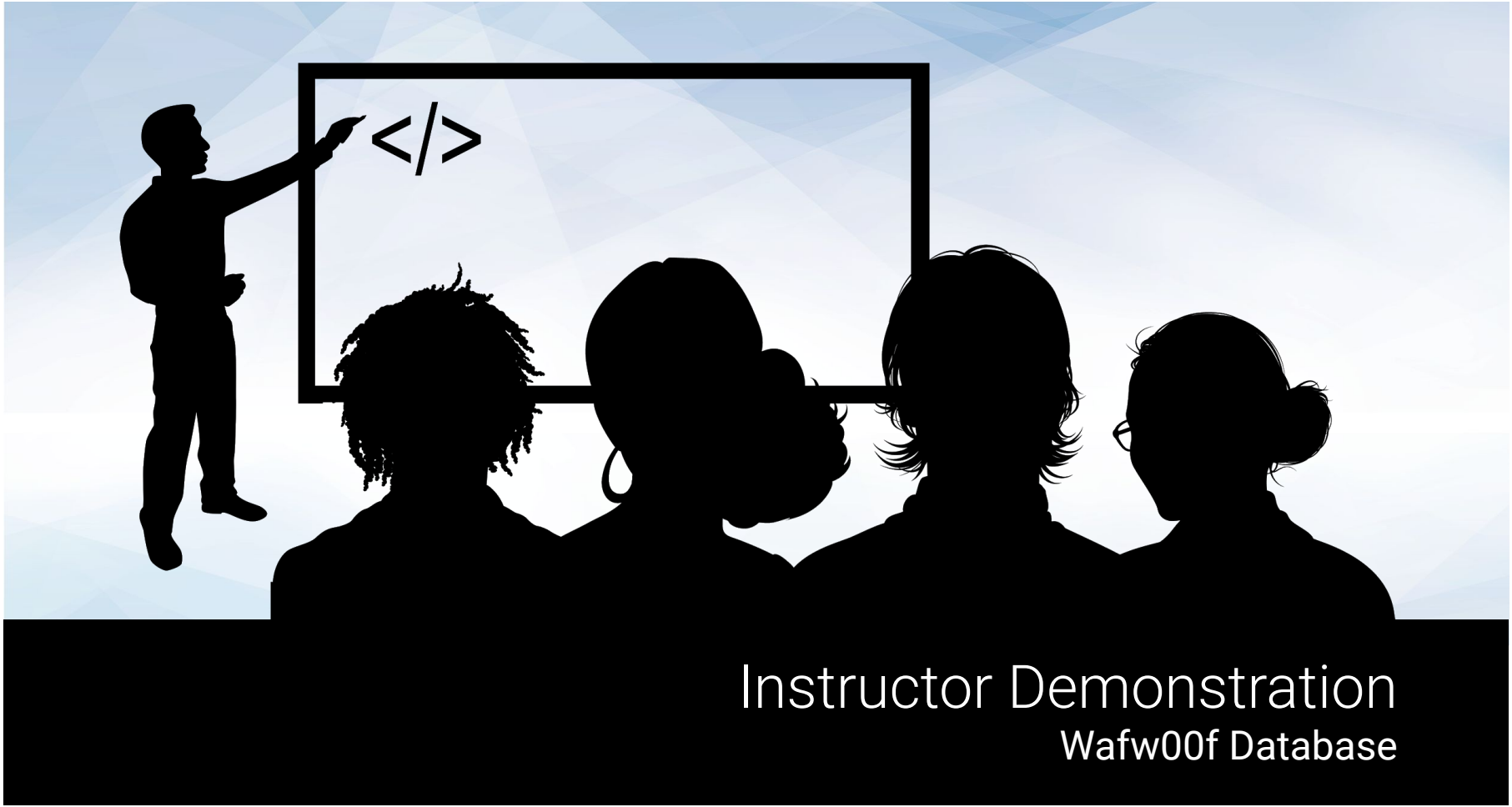
Wafw00f Demonstration

We'll demonstrate Wafw00f using the following scenario:

You're a criminal hacker looking to exploit any web vulnerability on a web site.

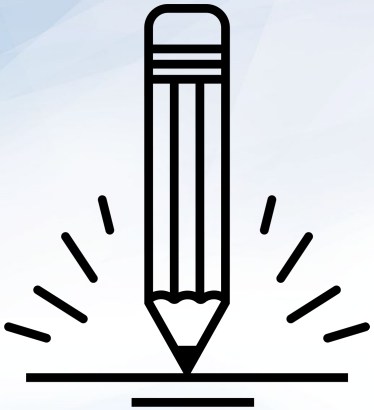
Before you can launch an attack, you need to know if the website is protected by a web application firewall and if so, what kind.

The results of this information gathering process will indicate the types of attacks that are available to you.



Instructor Demonstration

Wafw00f Database



Activity: Information Supermarket

In this activity, you will research how to mitigate data sourcing from web vulnerabilities that resulted in a surge of social engineering attacks.

Suggested Time:
15 Minutes

Time’s Up! Let’s Review.

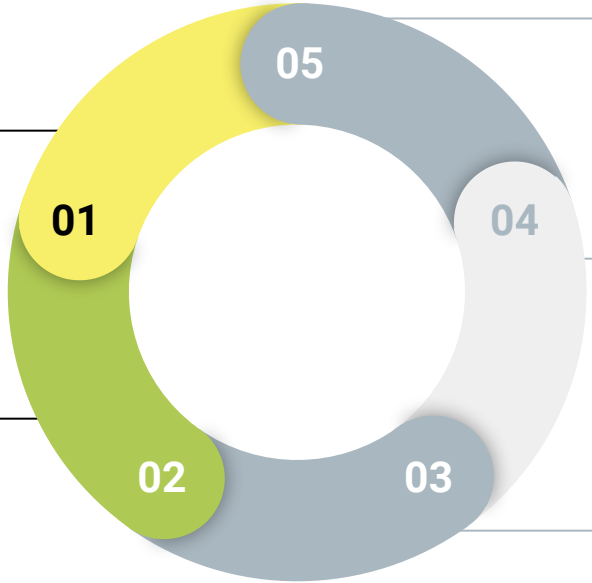
Executing Attacks

Stage 2: Execution

Now we will pivot from reconnaissance to the weaponization stage of the hybrid kill chain

Reconnaissance
Information gathered against a target

Weaponization
Preparation of offensive operations against specific targets using information gathered during reconnaissance.



Delivery
Launch of the operation. Attacks carried out based on Red Team offensive strategies.

Exploitation
Active compromise of adversary's apps, servers, or network, and averting physical, logical, or administrative controls.

Exfiltration
Ultimate goal. The exfiltration of private, sensitive data that the target considers critically sensitive.

Stage 2: Execution

By performing three specific attacks prevalent in today's cyber community:

01

Parameter tampering

02

Directory traversal

03

Cross-site scripting

Parameter Tampering

Parameter tampering is a web-based attack in which specific URL parameters or webpage form field data is modified from its original state.

Attackers use parameter tampering to take advantage of web vulnerabilities resulting from insecure web programming. This attack is used to obtain private personal or business information.

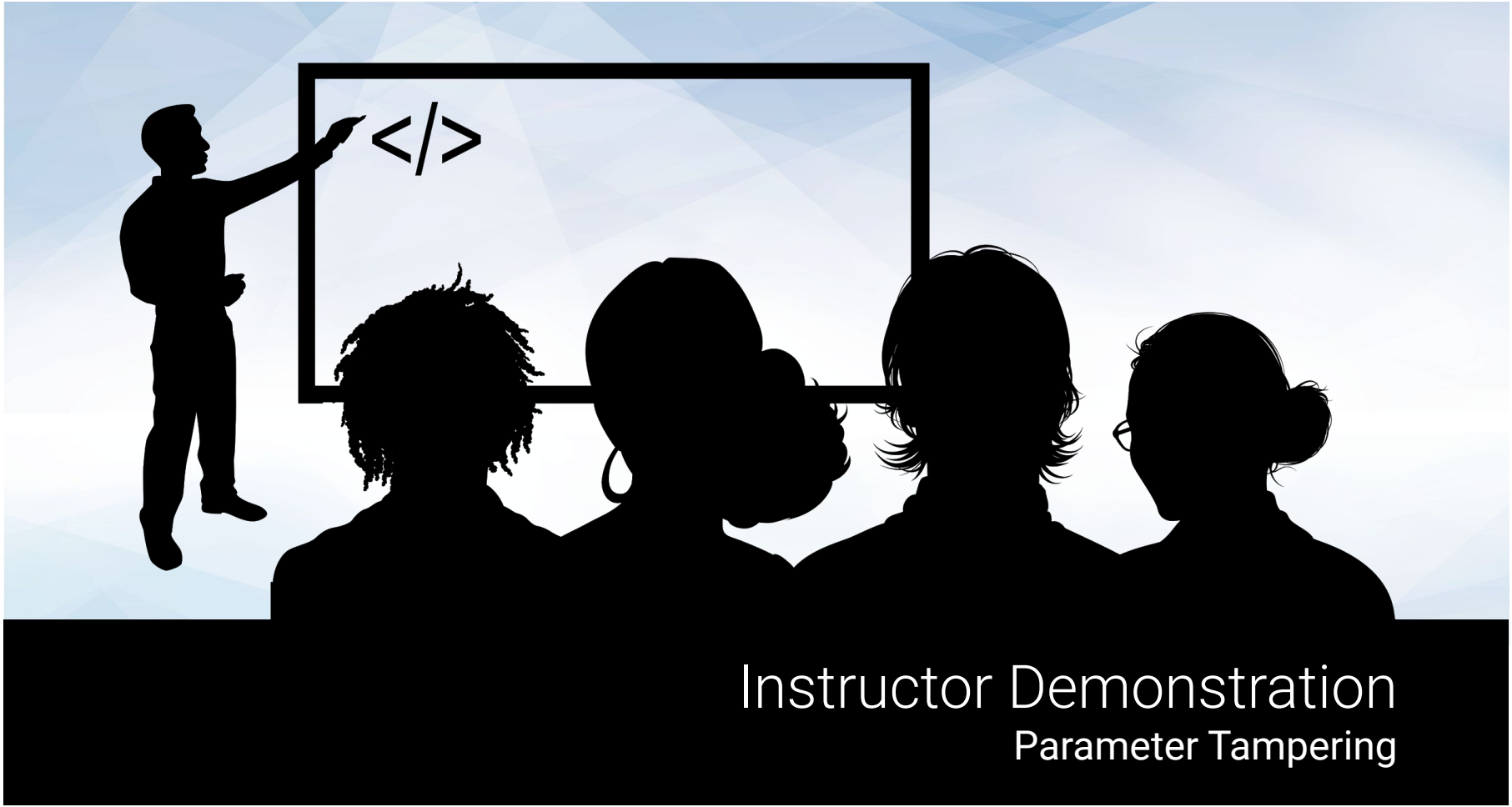
Parameter Tampering Demo Setup

We'll demonstrate how to execute a parameter tampering attack with the following scenario:

You visited a webpage of a major online retailer looking to buy a new laptop.

You found the perfect laptop but cannot afford the price tag of $1,999.00.

You've decided to take advantage of a potential web vulnerability and change the sales price of the laptop by altering a form field on the webpage.



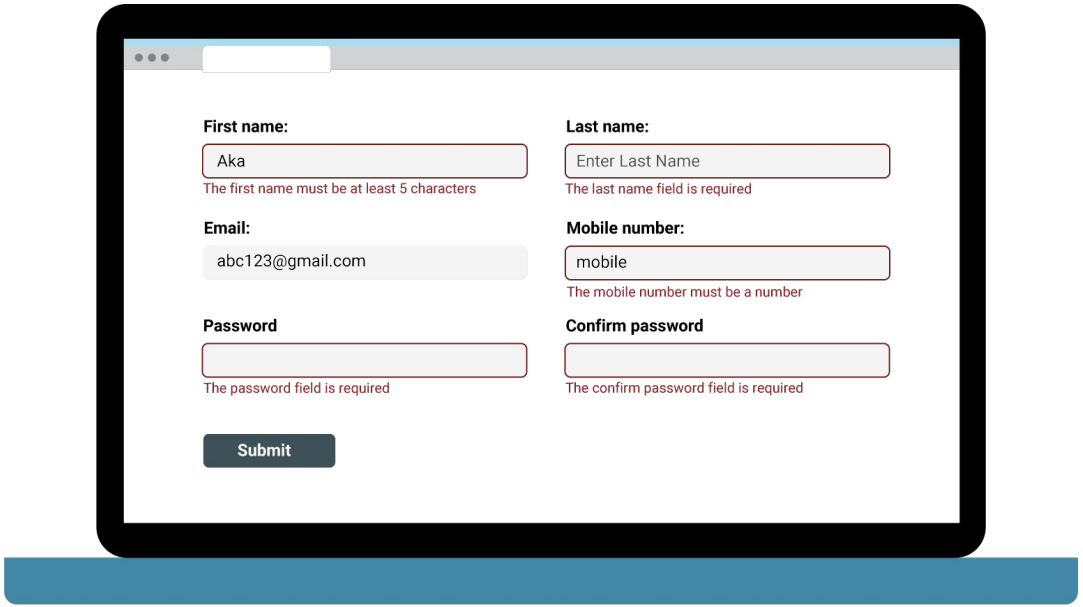
Instructor Demonstration

Parameter Tampering

Parameter Tampering Hardening

While less common today, this web vulnerability results from a lack of input validation.

- **Input validation** is the process of testing the supplied input of a user or program, which detects and prevents improperly formatted data from entering an information system.
- Additional mitigation strategies include:
  - Restricted numeric range
  - Restricted character sequence and patterns
  - Restricted use of `null`
  - Restricted parameter use of unused form fields



Path/Directory Traversal

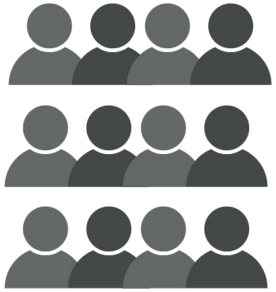
Path traversal is an attack that exploits web vulnerabilities by referencing files with a dot-dot-slash (../) sequence in the URL.

A vulnerable web server can allow an attacker to access arbitrary files and directories stored on a web server. This can result in the loss of application source code, system configuration information, and/or critical system files.

Path Traversal Demo Setup

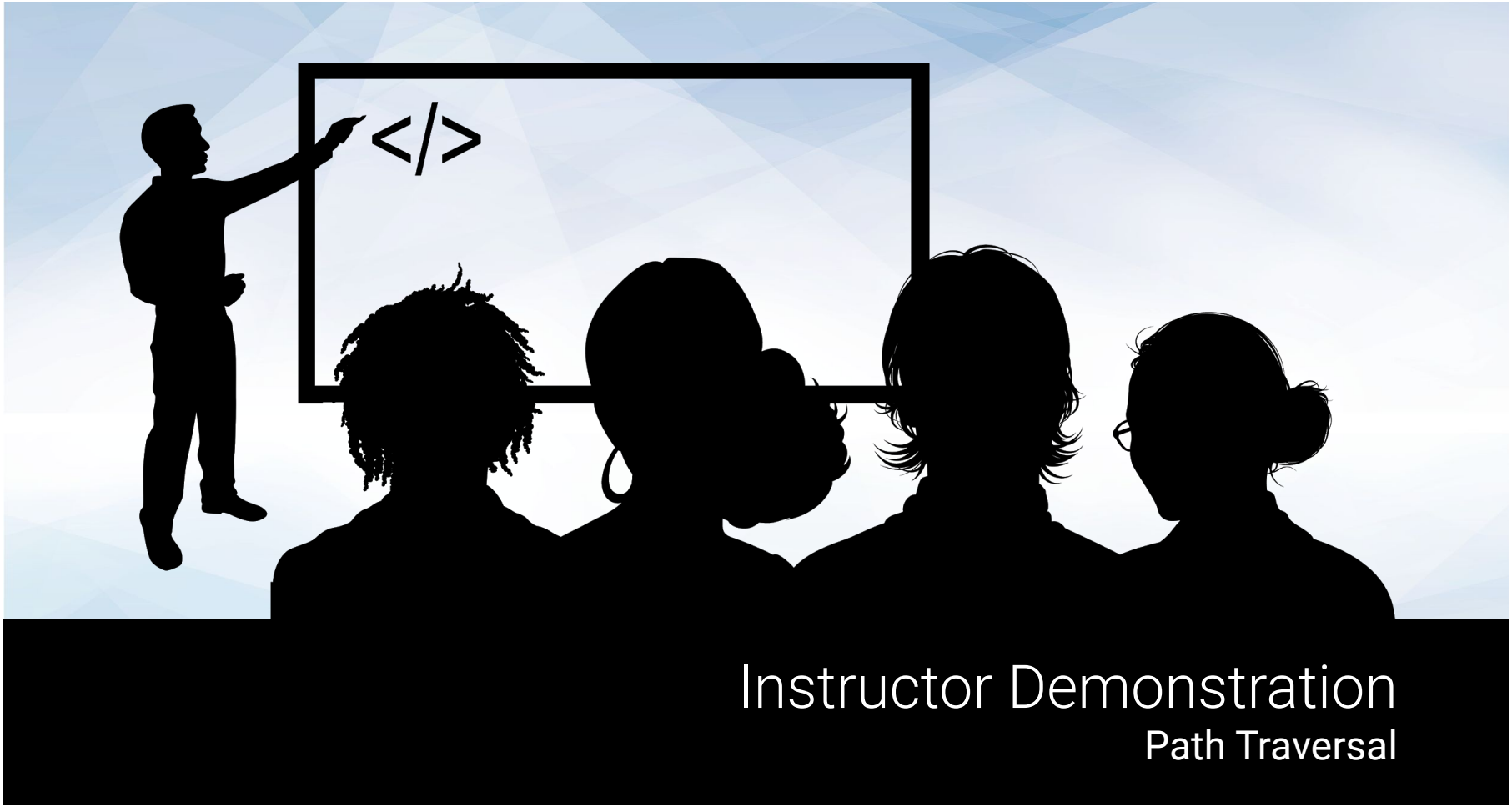
We'll demonstrate how to execute a path traversal attack with the following scenario:

You're a hacktivist who's politically motivated to shame an organization with opposing views.



You've decided to hack into the opposition groups email server with and make private emails publicly available. To carry out this mission, you must harvest a list of all users.

You will attempt to exploit a possible web vulnerability by using path traversal.

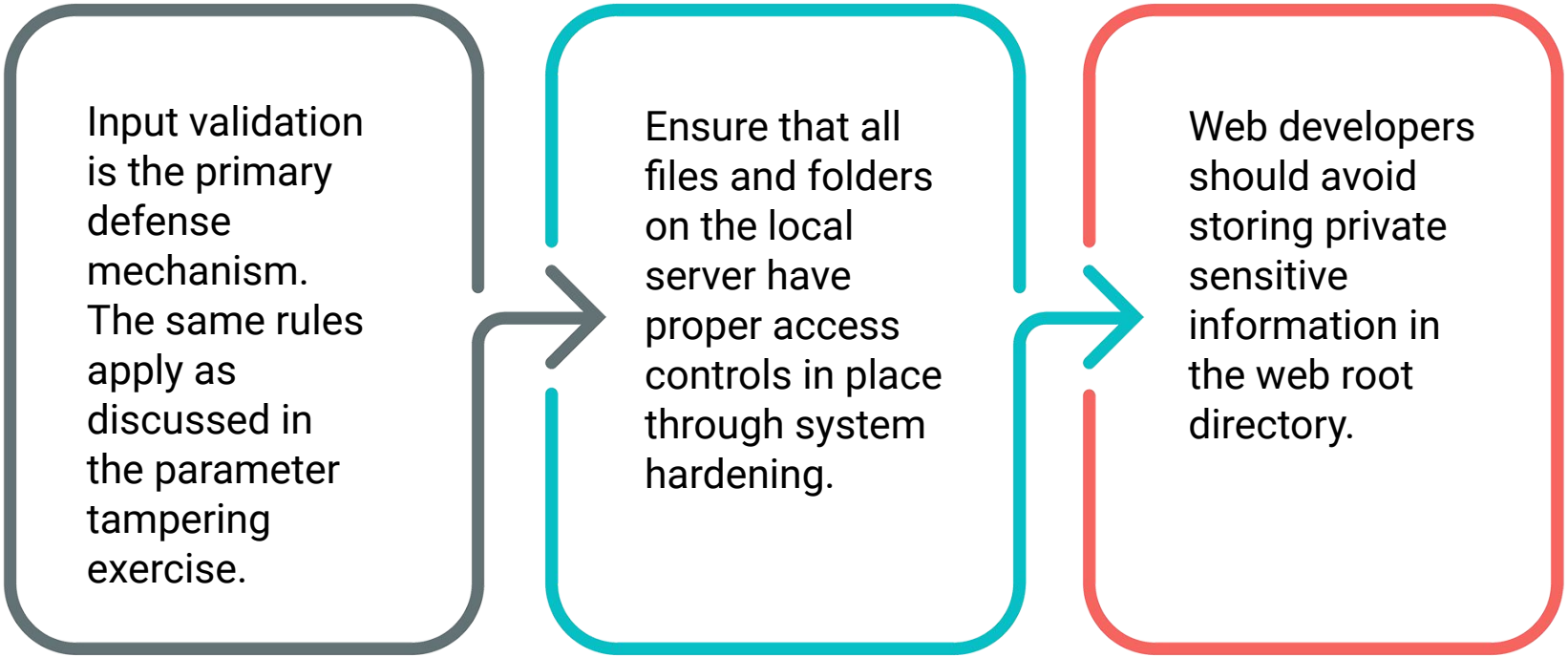


Instructor Demonstration

Path Traversal

Path Traversal Mitigation

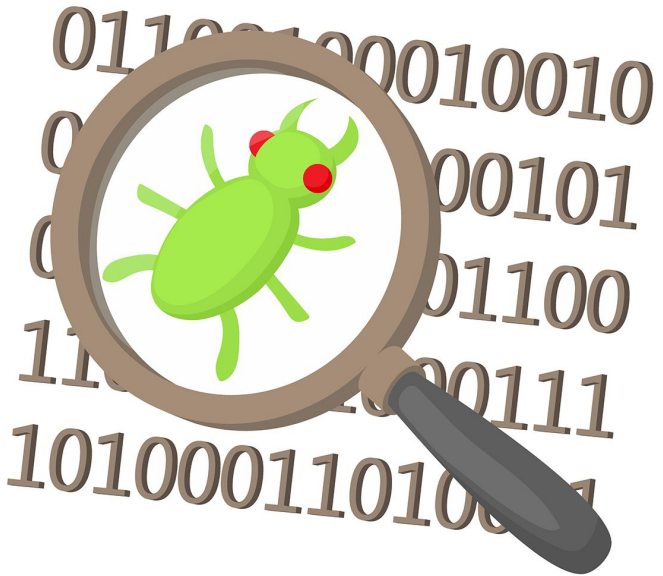
Path traversal is usually the result of a lack of input validation.



Cross-Site Scripting (XSS)

Cross-site scripting (XSS) attacks occur when an attacker takes advantage of a web vulnerability by injecting malicious code into a web browser that's stored on the back-end server. This is typically in the form of a script, which then affects subsequent visitors to that same web page.

- An end user's browser doesn't have the capability to detect such attacks because of the default trust relationship it has with the source.
- As a result, the affected end user's web browser allows the malicious script free access to session tokens, cookies, and any other sensitive information stored by the browser.



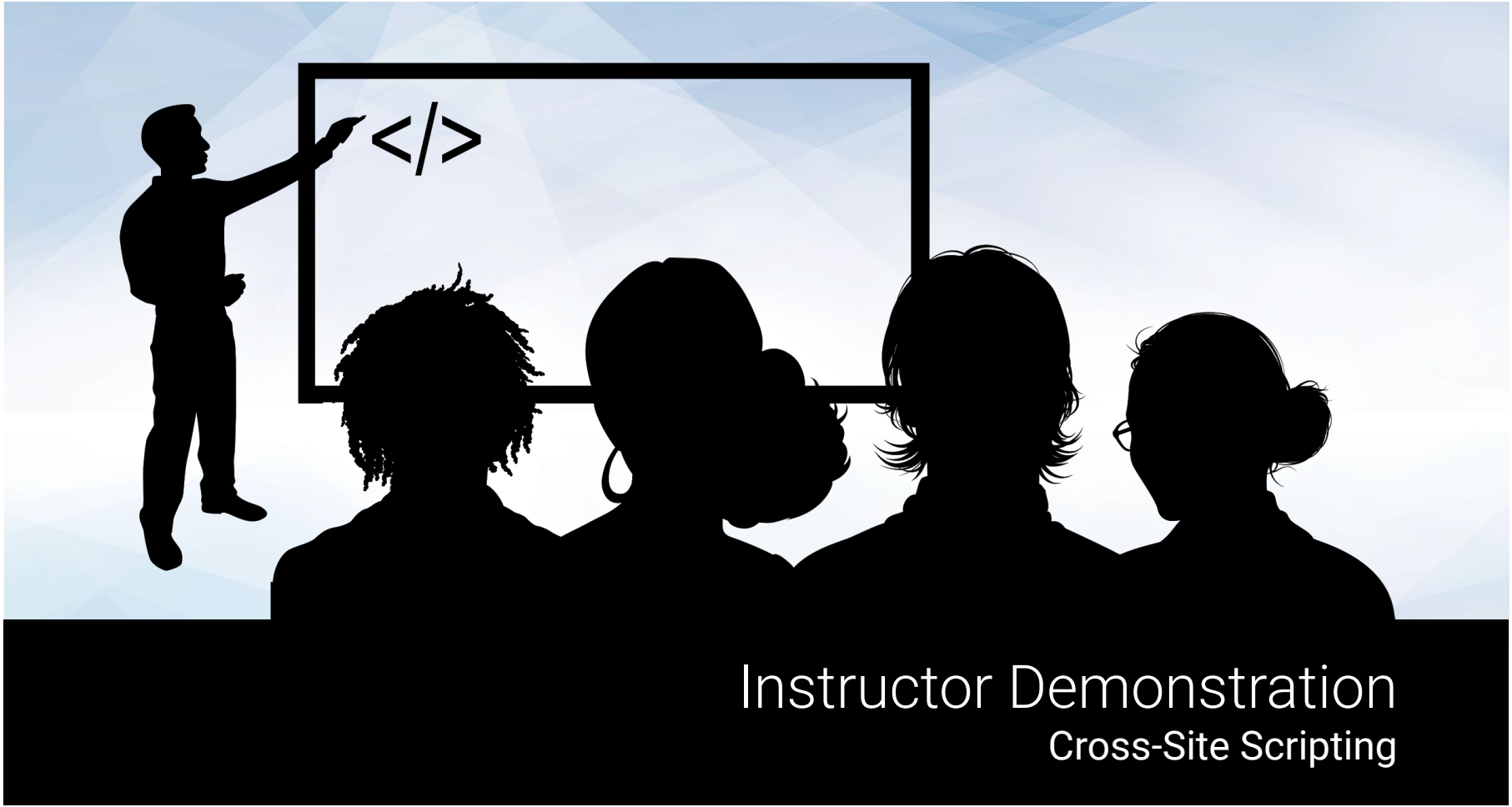
XSS Demo Setup

We'll demonstrate how to execute a cross-site scripting attack with the following scenario:

You're a criminal hacker whose mission is to gain root access to a web server.

Before you can do that, you need to steal a session cookie.

You will perform a cross-site scripting attack, but first will need to determine if the attack is possible.



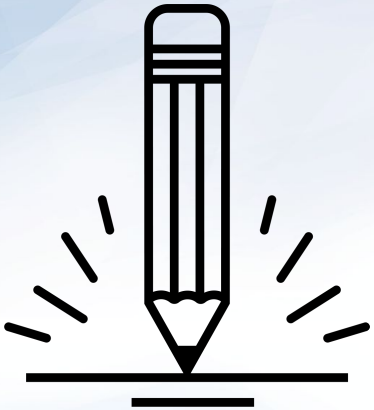
Instructor Demonstration

Cross-Site Scripting

XSS Mitigation

Mitigation strategies include:

- Data Output Encoding: Encodes user-controlled data before its output in an HTTP response. This prevents it from being misinterpreted as active content.
- Data Input Filters: Filters data at the point where user input is received, based on expected or valid input criteria.
- Content Security Policy (CSP): Detects and mitigates specific types of attacks, including XSS and other injection attacks. (Not all browsers support CSP.)



Activity: Executing Exploits

In this activity, you will use OWASP's Broken Web Apps to demonstrate the various ways a website can be exploited.

Suggested Time:
20 Minutes

Time’s Up! Let’s Review.


The End