# INTRODUCTION

Web applications are interactive applications that run on web browsers and adopt a client-server architecture. They consist of front-end components that run on the client-side and back-end components that run on the server-side. They provide organizations with the ability to host powerful applications accessible worldwide. Web applications can be developed by any web developer and hosted online, leading to a vast number of applications available on the internet. They offer dynamic content, unlike static websites, and can perform various functionalities for end-users. Web applications can run on any operating system and are not installed on the user's system, making them platform-independent. However, they face security risks and require frequent testing for vulnerabilities. Common methods for testing web applications include reviewing front-end components and assessing core functionality. Attacks on web applications can lead to significant damage, emphasizing the need for web application penetration testing and secure coding practices. A deep understanding of web applications and their vulnerabilities is crucial for effective security testing and protection.

# WEB APPLICATION LAYOUT

Web applications can have different infrastructure setups, including client-server, one server, many servers - one database, and many servers - many databases. The client-server model is the most common, where a server hosts the web application and distributes it to clients. In the one server architecture, the entire web application is hosted on a single server, while the many servers - one database model separates the database onto its own server. The many servers - many databases architecture further separates each web application's data into separate databases. These infrastructure models have different advantages in terms of security, redundancy, and scalability.

# FRONT END VS BACKEND

Front-end web development focuses on creating the user interface and experience of a web application. It involves using HTML, CSS, and JavaScript to build the components that users interact with in their browsers. Front-end development requires optimization for different platforms and devices. On the other hand, back-end development handles the core functionality of a web application, processing requests and managing databases. It involves working with servers, databases, and development frameworks. Both front-end and back-end development have their own set of tasks and responsibilities. Securing both components is crucial to prevent vulnerabilities and protect against attacks. Common mistakes made by web developers can lead to vulnerabilities, which are often categorized in the OWASP Top 10 list. Understanding these vulnerabilities is important for penetration testers and ensuring the security of web applications.

# HTML

HTML (Hypertext Markup Language) is a standard markup language used for creating the structure and content of web pages. It provides a set of tags that define the elements and layout of a web document. HTML tags are used to mark up different parts of the content, such as headings, paragraphs, links, images, tables, forms, and more. These tags define the structure and semantics of the content, allowing web browsers to interpret and display it correctly. HTML is the backbone of every web page and is essential for front-end web development.

## QUESTION

What is the HTML tag used to show an image? <img>

# CASCADING STYLE SHEETS(CSS)

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation and appearance of a document written in HTML or XML. CSS allows web developers to control the visual aspects of a web page, such as the layout, colours, fonts, and other design elements. By separating the presentation from the content, CSS provides flexibility and consistency across multiple web pages.

## QUESTION

What is the CSS "property: value" used to make an HTML element's text aligned to the left? text-align: left;

# JAVASCRIPT

JavaScript is a programming language used for client-side web development. It enhances web pages by adding interactivity and dynamic behavior. JavaScript syntax resembles other programming languages, and it is embedded within HTML or included as external files. It manipulates the DOM, handles events, and enables asynchronous communication with servers. JavaScript has a vast ecosystem of libraries and frameworks and can be used both on the client and server sides. It is widely supported by modern browsers and is crucial for creating interactive and responsive websites.

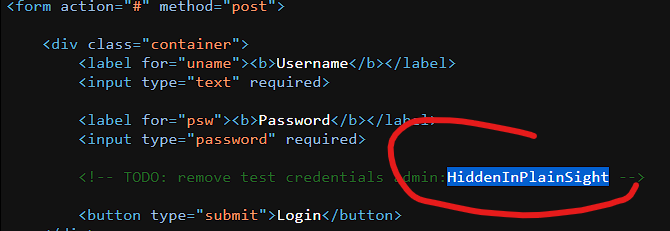
# SENSITIVE DATA EXPOSURE

Sensitive data exposure in web applications refers to the vulnerability where confidential information is inadequately protected, leading to unauthorized access and potential exposure. This can occur due to weak encryption, insecure storage practices, or improper handling of user data. To mitigate this risk, web applications should implement robust security measures such as strong encryption, secure authentication mechanisms, and regular vulnerability assessments to ensure the protection of sensitive data.

## TASKS

Check the above login form for exposed passwords. Submit the password as the answer.

HiddenInPlainSight



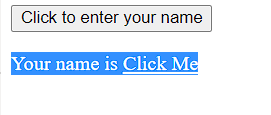
# HTML INJECTION

HTML injection, also known as client-side code injection, is a security vulnerability where untrusted user input is improperly handled and rendered as HTML on a web page. This can allow attackers to inject malicious code, such as scripts or tags, into the page, leading to various consequences like cross-site scripting (XSS) attacks. HTML injection typically occurs when user-supplied data is directly inserted into HTML templates without proper sanitization or validation. To prevent HTML injection, web applications should implement input validation and output encoding techniques to ensure that user input is treated as data rather than executable code.

## TASKS

What text would be displayed on the page if we use the following payload as our input: <a href="http://www.hackthebox.com">Click Me</a>

Your name is [Click Me](http://www.hackthebox.com/)



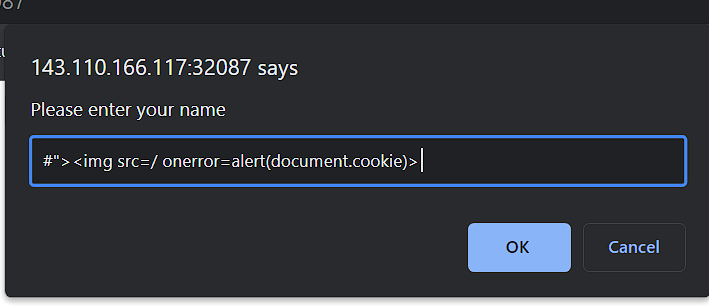
# CROSS-SITE SCRIPTING (XSS)

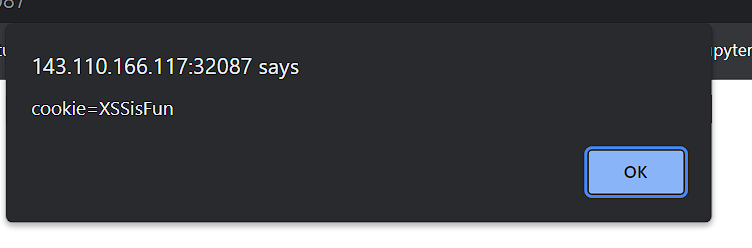
Cross-Site Scripting (XSS) is a type of security vulnerability commonly found in web applications. It occurs when an attacker is able to inject malicious scripts or code into a trusted website, which is then executed by the victim's browser. XSS attacks can have various forms, such as stored XSS, reflected XSS, or DOM-based XSS, but they all involve the injection of malicious code that is executed within the context of a legitimate website.

## TASKS

Try to use XSS to get the cookie value in the above page

cookie=XSSisFun





# BACKEND SERVERS

This is the hardware and OS ‘Behind-the-scenes’ hosting all applications used to run the webserver.

## QUESTION

What operating system is 'WAMP' used with?

Windows

# WEB SERVERS

The back end server, which manages all HTTP traffic from client-side browsers, directs it to the requested sites, and then responds to the client-side browser, is where a web server is an application that operates. online servers are in charge of connecting end users to various components of the online application and managing their varied replies. They typically run on TCP ports 80 or 443.

## QUESTION

If a web server returns an HTTP code 201, what does it stand for?

Created

# DATABASES

Back end databases are used by web applications to store diverse content and data connected to the online application. Core web application assets like files and photos, web application content like posts and updates, or user information like usernames and passwords can all be examples of this. This makes it possible for web applications to store and retrieve data fast and simply, as well as to provide dynamic content that is unique for each user.

## QUESTION

What type of database is Google's Firebase Database?

NoSQL

# DEVELOPMENT FRAMEWORKS & APIs

There are several popular web development frameworks that aid in creating the fundamental files and functionality of online applications. It could be difficult to build a cutting-edge web application from scratch given the growing complexity of online apps. As a result, the majority of well-known online apps are created utilizing web frameworks. Among the most popular web development frameworks are Rails (Ruby), Laravel (PHP), Express (Node.JS), and Django (Python).

## TASKS

Use GET request '/index.php?id=0' to search for the name of the user with id number 1?

superadmin



# COMMON WEB VULNERABILITIES

Common web vulnerabilities refer to the security weaknesses and flaws frequently found in web applications and websites. These vulnerabilities can be exploited by malicious actors to gain unauthorized access, manipulate data, or disrupt the normal functioning of the application. Some of the most prevalent web vulnerabilities include Cross-Site Scripting (XSS), SQL Injection, Cross-Site Request Forgery (CSRF), and Server-Side Request Forgery (SSRF). XSS involves injecting malicious scripts into web pages, allowing attackers to execute arbitrary code on users' browsers. SQL Injection occurs when attackers manipulate input fields to execute unauthorized database queries. CSRF exploits the trust between a user's browser and a website, tricking users into performing unintended actions. SSRF involves manipulating a web application to make arbitrary requests to internal resources. These vulnerabilities highlight the importance of implementing robust security measures, such as input validation, proper access controls, and secure coding practices, to protect web applications from potential attacks.

## QUESTION

To which of the above categories does public vulnerability 'CVE-2014-6271' belongs to?

Command Injection

# PUBLIC VULNERABILITIES

Common web vulnerabilities pose significant risks to the security of web applications. These vulnerabilities can be exploited by attackers to gain unauthorized access, manipulate data, or disrupt the normal functioning of the application. One such vulnerability is HTML injection, where an attacker can inject malicious HTML code into a web page, leading to various security risks. Another prevalent vulnerability is Cross-Site Scripting (XSS), which involves injecting malicious scripts into web pages viewed by users, allowing attackers to steal sensitive information or perform unauthorized actions on behalf of the user. These vulnerabilities highlight the importance of implementing robust security measures, such as input validation and secure coding practices, to mitigate the risk of exploitation and protect web applications and users from potential harm. Regular security assessments and patching of vulnerabilities are essential to maintain the security of web applications in the ever-evolving threat landscape.

## QUESTION

What is the CVSS score of the public vulnerability CVE-2017-0144?

9.3

# 16. CONCLUSION

This assignment has provided a comprehensive overview of web applications, covering various aspects such as their layout, front-end and back-end development, HTML, CSS, JavaScript, and common vulnerabilities. I have learned that web applications are interactive applications that run on web browsers and consist of front-end components and back-end components. Front-end development focuses on creating the user interface, while back-end development handles the core functionality and database management. HTML is used for creating the structure of web pages, CSS controls the presentation and appearance, and JavaScript adds interactivity. I have also gained knowledge about important security considerations, including sensitive data exposure, HTML injection, cross-site scripting (XSS), and the significance of secure coding practices. Understanding these concepts and vulnerabilities is crucial for effective security testing and protection of web applications. [**LINK**](https://academy.hackthebox.com/achievement/643478/75)