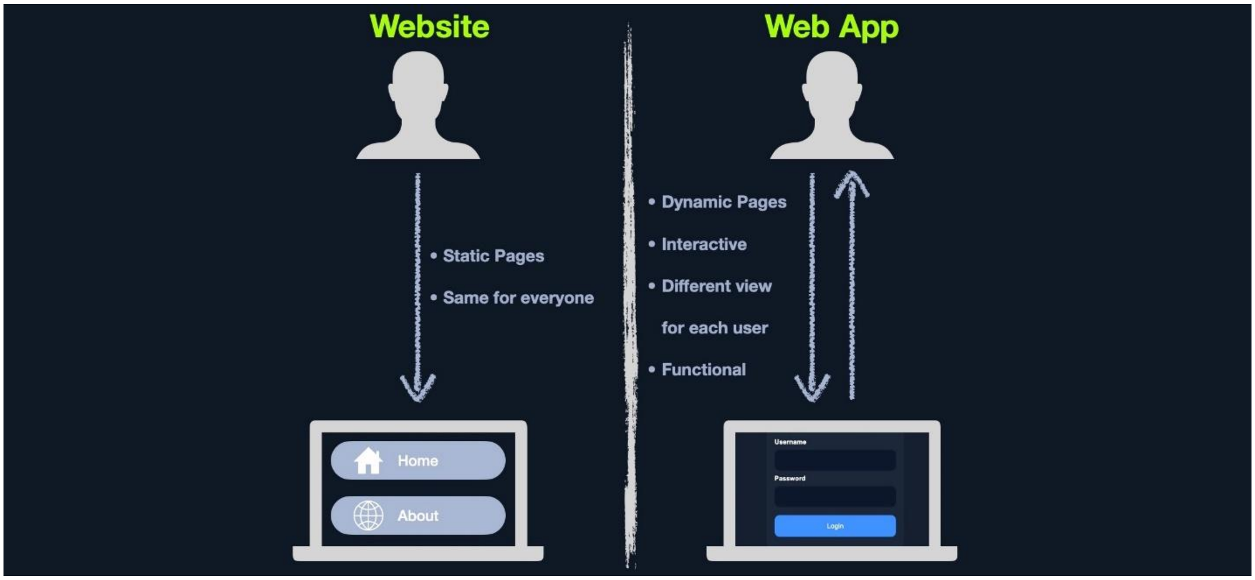
Week5: **Assignment 1:- Introduction to Web Applications**

Report by: **Aisha Khalifan, cs-cns04-23014**

Introduction

Web applications are the digital backbone of our lives. In this module, we will investigate their core elements, front-end and back-end distinctions, and the vital technologies behind them. We will also scrutinize vulnerabilities and how to address them. By the end, you will have the knowledge to confidently understand and secure web applications in our interconnected world.

1. Introduction to Web Applications

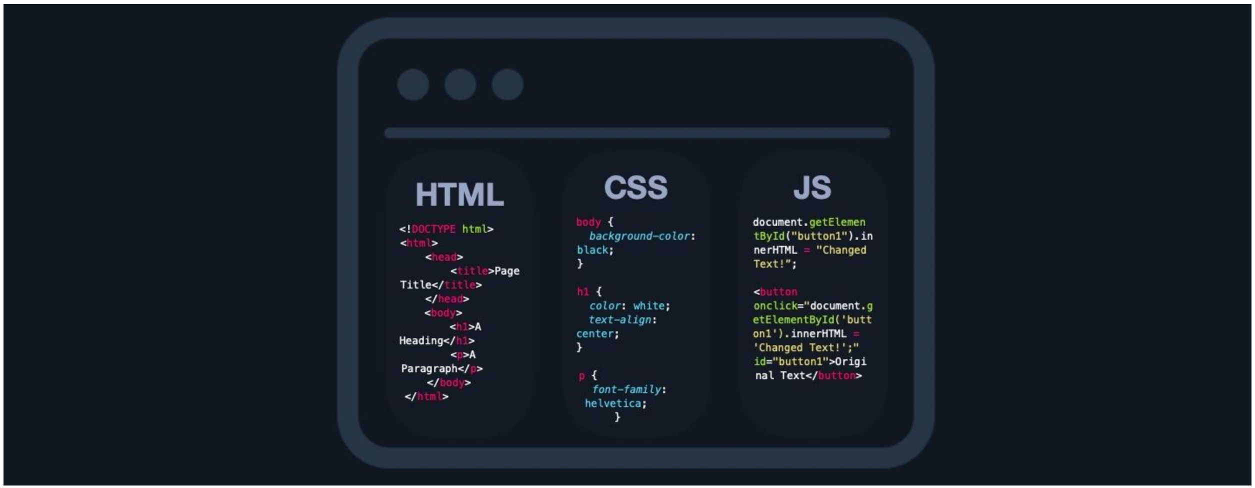


a. Web Application Layout

This consists of many different layers that can be summarized with the following three main categories:

Category	**Types**
Web Application Infrastructure	♦ Client-Server ♦ One Server ♦ Many Servers - One Database ♦ Many Servers - Many Databases
Web Application Components	♦ Client ♦ Server(Webserver, Web Application Logic, Database) ♦ Services (Microservices)- 3rd Party Integrations, Web Application Integrations ♦ Functions (Serverless)
Web Application Architecture	♦ Presentation, Application and Data Layers

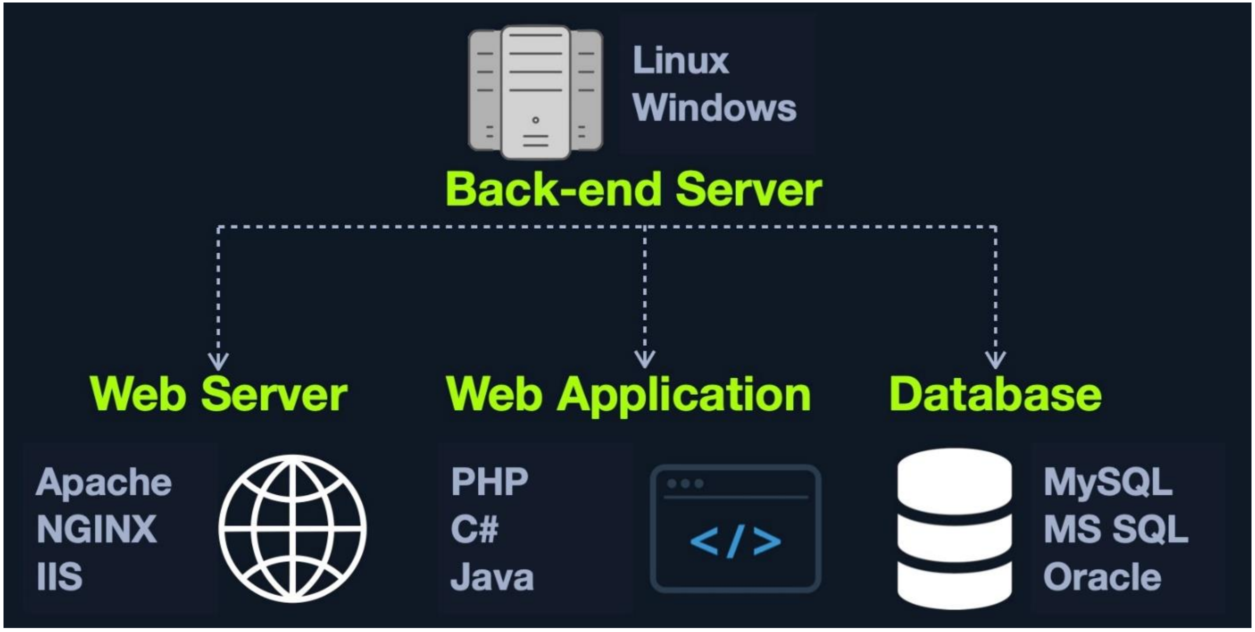
b. Front End vs. Back End



```html
<!DOCTYPE html>
<html>
    <head>
        <title>Page
Title</title>
    </head>
    <body>
        <h1>A
Heading</h1>
        <p>A
 Paragraph</p>
    </body>
 </html>
```

```css
body {
  background-color:
black;
}

h1 {
  color: white;
  text-align:
center;
}

 p {
  font-family:
 helvetica;
      }
```

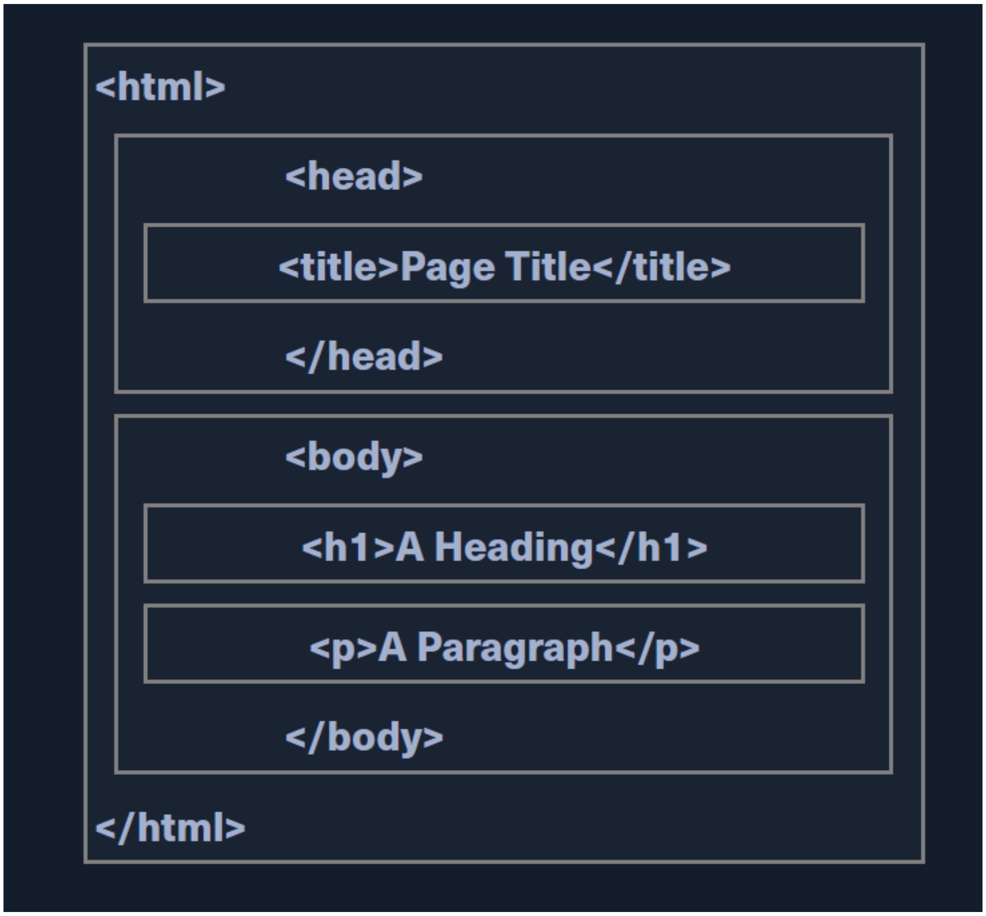
```javascript
document.getElemen
tById("button1").in
nerHTML = "Changed
Text!";

<button
onclick="document.g
etElementById('butt
on1').innerHTML =
'Changed Text!';"
id="button1">Origi
nal Text</button>
```



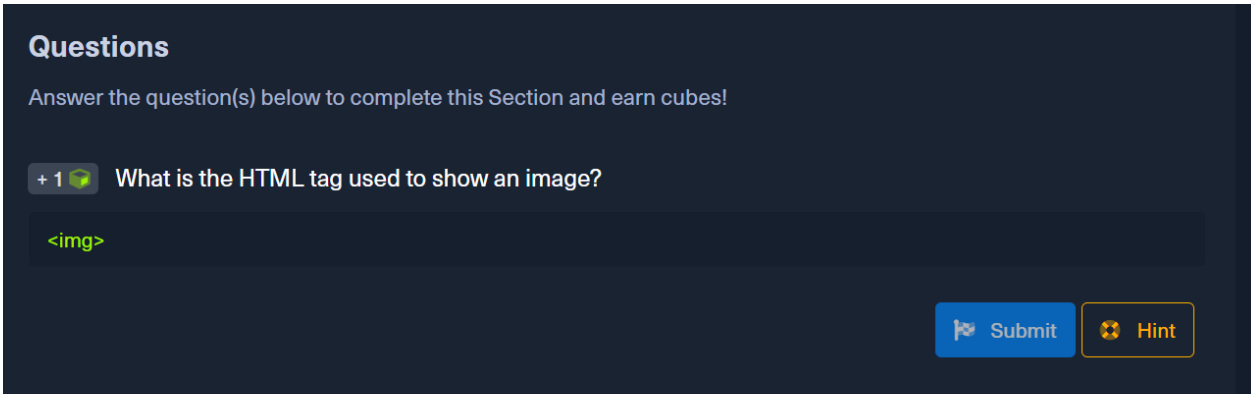
2. <u>Front End Components</u>

a. HTML

HTML is at the very core of any web page we see on the internet. It contains each page's basic elements, including titles, forms, images, and many other elements. The web browser, in turn, interprets these elements and displays them to the end-user.



```
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>
        <h1>A Heading</h1>
        <p>A Paragraph</p>
    </body>
</html>
```



Questions

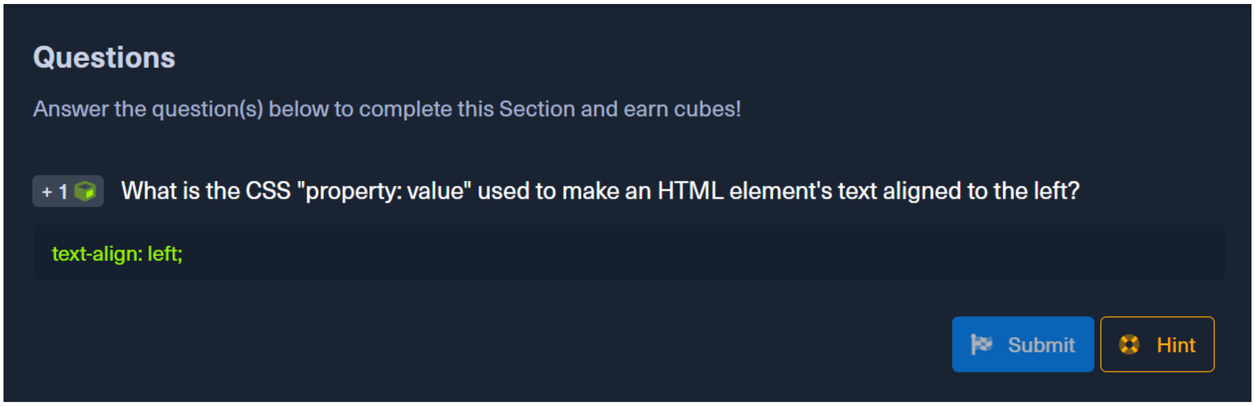
Answer the question(s) below to complete this Section and earn cubes!

+ 1 What is the HTML tag used to show an image?

<img>

b. Cascading Style Sheets (CSS)

- CSS (Cascading Style Sheets) is the stylesheet language used alongside HTML to format and set the style of HTML elements.
- Some of the most common CSS frameworks are: Bootstrap, SASS, Foundation, Bulma, Pure



Questions

Answer the question(s) below to complete this Section and earn cubes!

+1 What is the CSS "property: value" used to make an HTML element's text aligned to the left?

text-align: left;

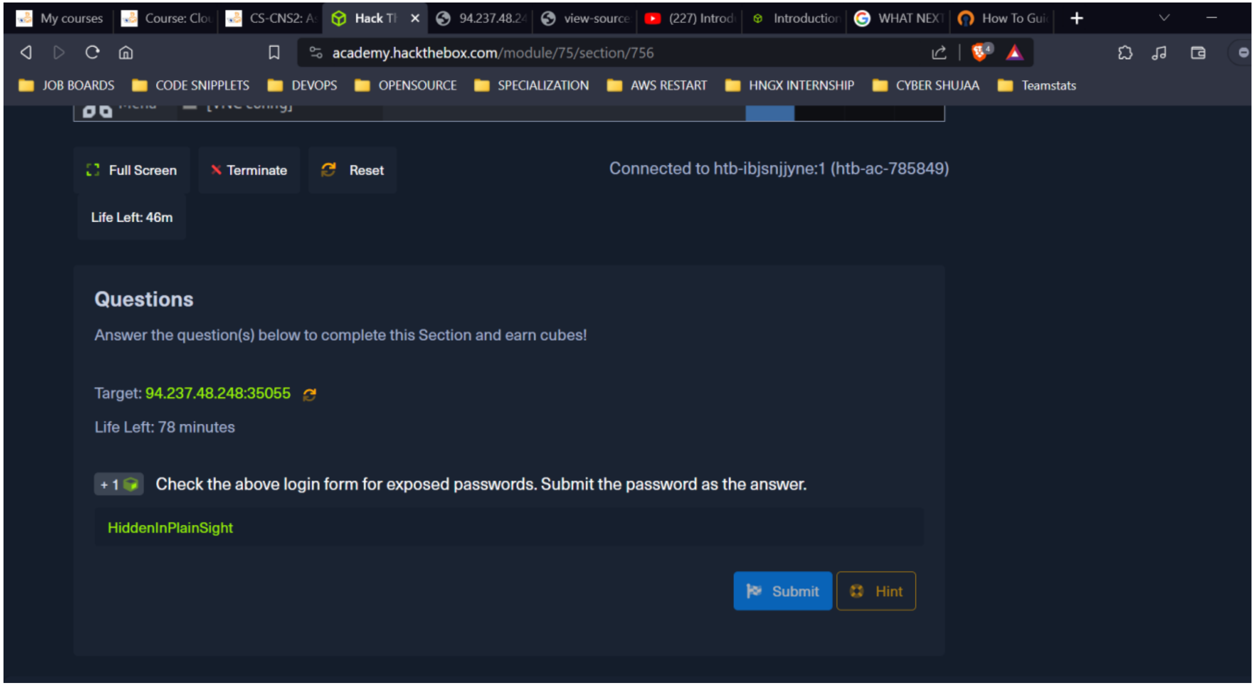
c. JavaScript

- JavaScript is usually used on the front end of an application to be executed within a browser.
- Some of the most common front end JavaScript frameworks are: Angular, React, Vue and jQuery

3. <u>Front End Vulnerabilities</u>

a. Sensitive Data Exposure

This refers to the availability of sensitive data in clear-text to the end-user. This is usually found in the source code of the web page or page source on the front end of web applications.



Connected to htb-ibjsnjjyne:1 (htb-ac-785849)

Life Left: 46m

Questions

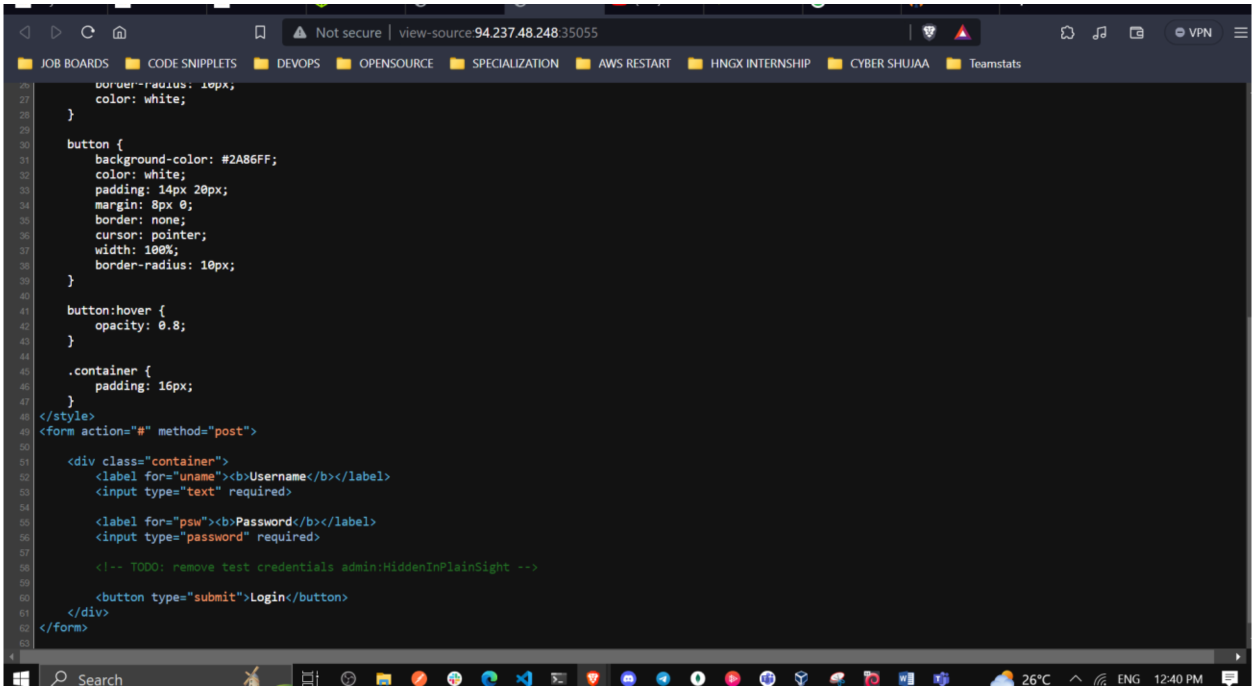
Answer the question(s) below to complete this Section and earn cubes!

Target: 94.237.48.248:35055

Life Left: 78 minutes

+1 Check the above login form for exposed passwords. Submit the password as the answer.

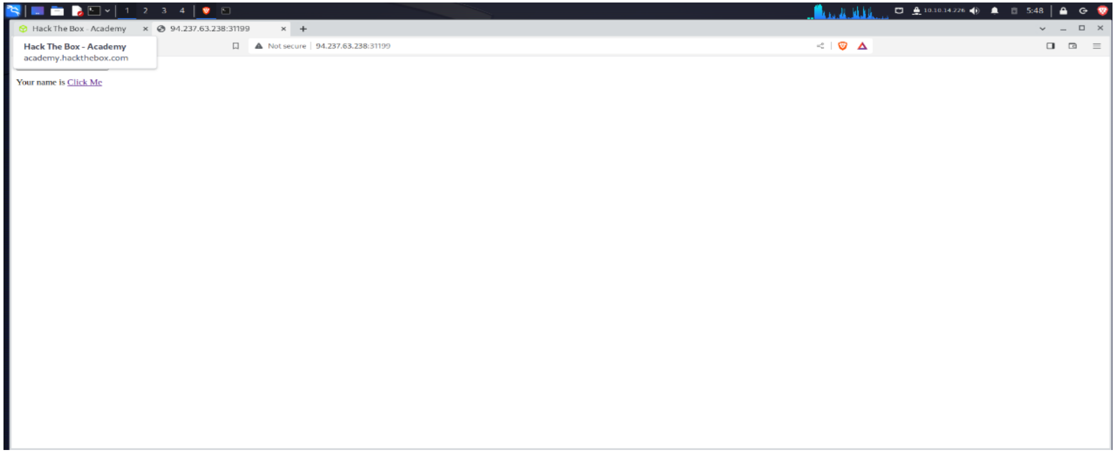
HiddenInPlainSight



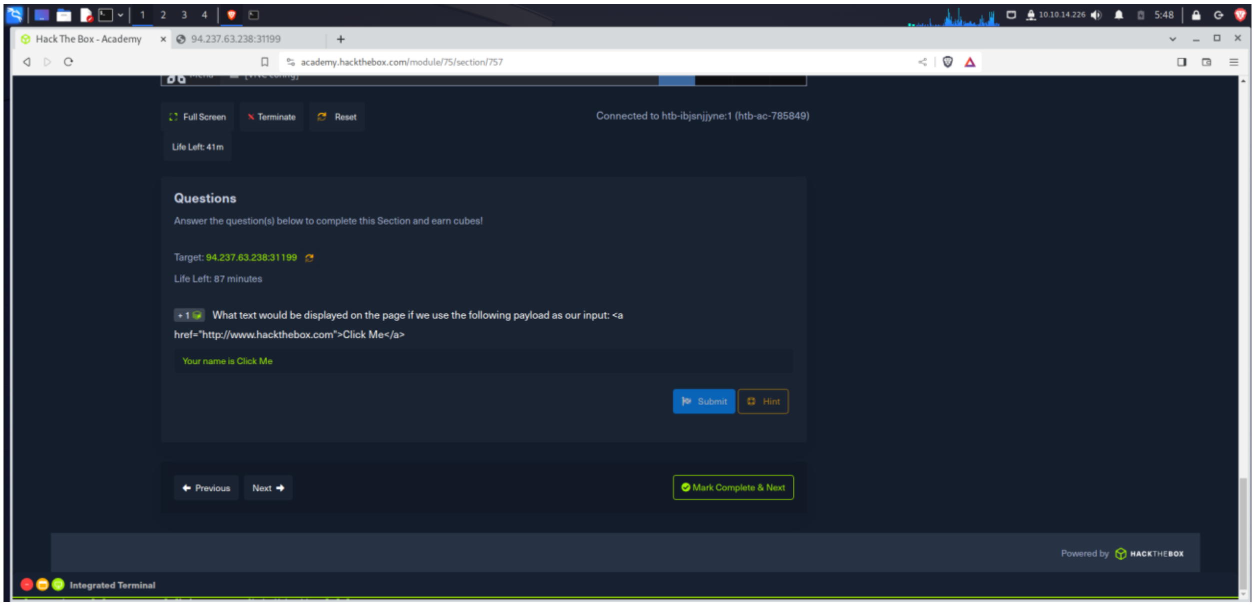
```
        border-radius: 10px;
        color: white;
    }

    button {
        background-color: #2A86FF;
        color: white;
        padding: 14px 20px;
        margin: 8px 0;
        border: none;
        cursor: pointer;
        width: 100%;
        border-radius: 10px;
    }

    button:hover {
        opacity: 0.8;
    }

    .container {
        padding: 16px;
    }
</style>
<form action="#" method="post">

    <div class="container">
        <label for="uname"><b>Username</b></label>
        <input type="text" required>

        <label for="psw"><b>Password</b></label>
        <input type="password" required>

        <!-- TODO: remove test credentials admin:HiddenInPlainSight -->

        <button type="submit">Login</button>
    </div>
</form>
```

b. HTML Injection

HTML injection occurs when unfiltered user input is displayed on the page. This can either be through retrieving previously submitted code, like retrieving a user comment from the back end database, or by directly displaying unfiltered user input through JavaScript on the front end.



Your name is Click Me



Connected to htb-ibjsnjjyne:1 (htb-ac-785849)

Questions

Answer the question(s) below to complete this Section and earn cubes!

Target: 94.237.63.238:31199

Life Left: 87 minutes

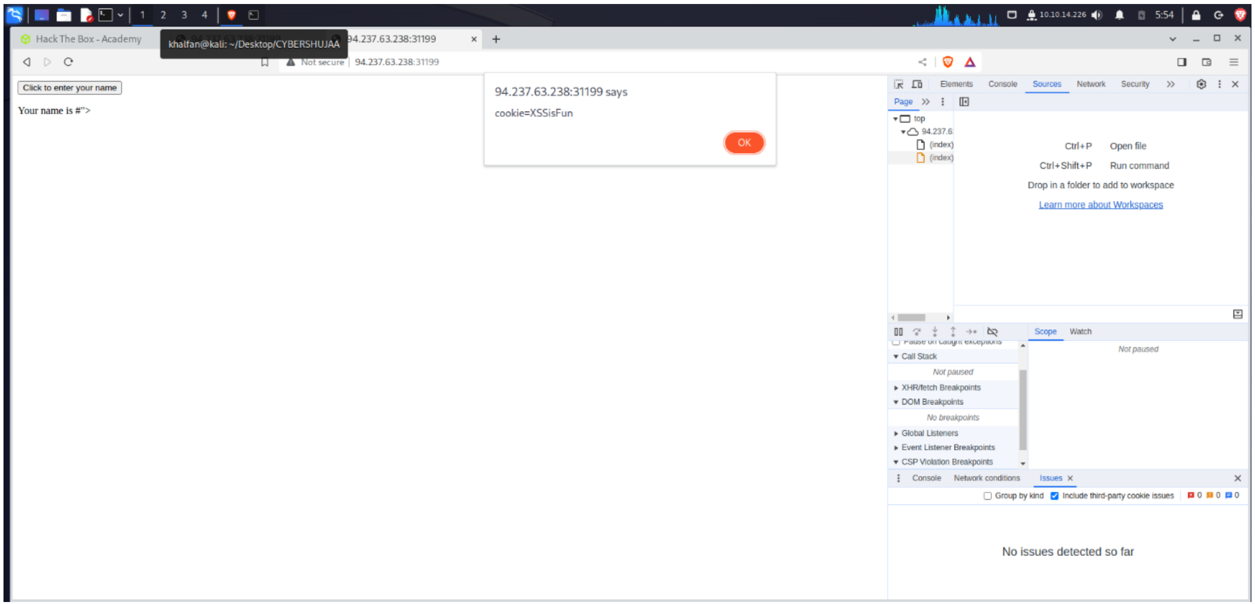
+1 What text would be displayed on the page if we use the following payload as our input: <a href="http://www.hackthebox.com">Click Me</a>

Your name is Click Me

c. **Cross-Site Scripting (XSS)**

HTML Injection vulnerabilities can often be utilized to also perform Cross-Site Scripting (XSS) attacks by injecting JavaScript code to be executed on the client-side. Once we can execute code on the victim's machine, we can potentially gain access to the victim's account or even their machine. XSS is very similar to HTML Injection in practice. However, XSS involves the injection of JavaScript code to perform more advanced attacks on the client-side, instead of merely injecting HTML code. There are three main types of XSS:

Type	Description
Reflected XSS	Occurs when user input is displayed on the page after processing (e.g., search result or error message).
Stored XSS	Occurs when user input is stored in the back end database and then displayed upon retrieval (e.g., posts or comments).
DOM XSS	Occurs when user input is directly shown in the browser and is written to an HTML DOM object (e.g., vulnerable username or page title).



Click to enter your name

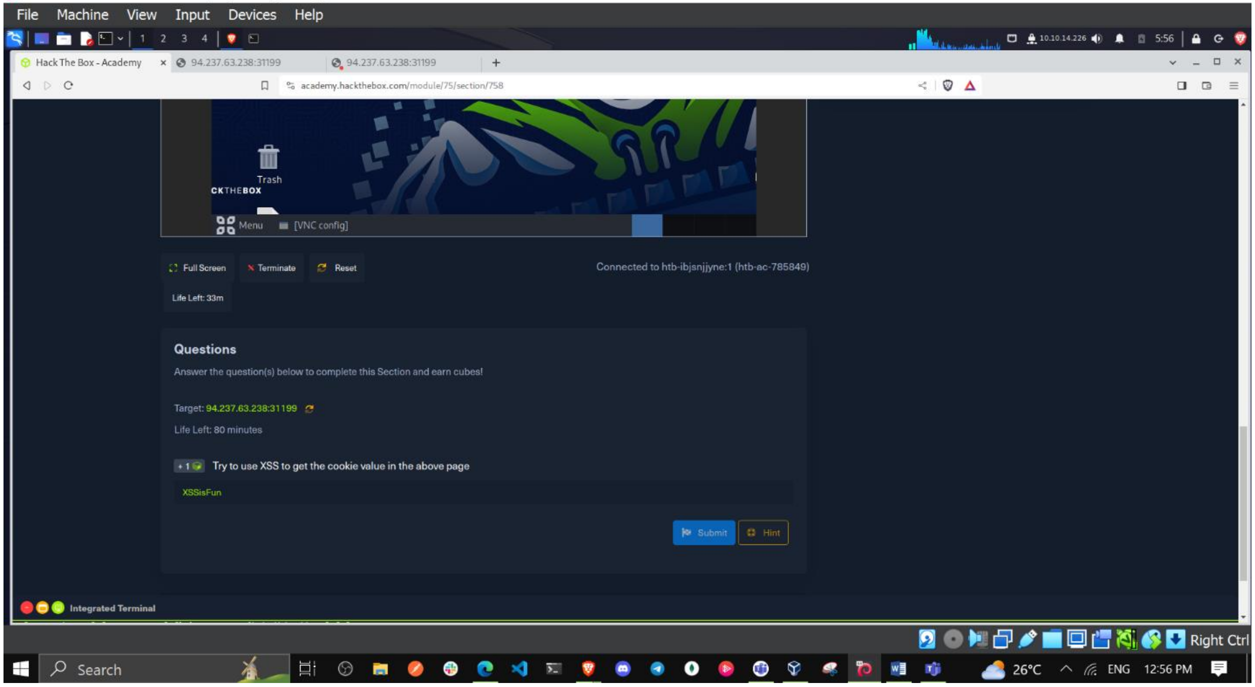
Your name is #">

94.237.63.238:31199 says

cookie=XSSisFun

OK

No issues detected so far



Connected to htb-ibjsnjjyne:1 (htb-ac-785849)

Life Left: 33m

Questions

Answer the question(s) below to complete this Section and earn cubes!

Target: 94.237.63.238:31199

Life Left: 80 minutes

+1 Try to use XSS to get the cookie value in the above page

XSSisFun

d. Cross-Site Request Forgery (CSRF)

The third type of front end vulnerability that is caused by unfiltered user input is Cross-Site Request Forgery (CSRF). CSRF attacks may utilize XSS vulnerabilities to perform certain queries, and API calls on a web application that the victim is currently authenticated to. This would allow the attacker to perform actions as the authenticated user. It may also utilize other vulnerabilities to perform the same functions, like utilizing HTTP parameters for attacks.

A common CSRF attack to gain higher privileged access to a web application is to craft a JavaScript payload that automatically changes the victim's password to the value set by the attacker. Once the victim views the payload on the vulnerable page (e.g., a malicious comment containing the JavaScript CSRF payload), the JavaScript code would execute automatically. It would use the victim's logged-in session to change their password. Once that is done, the attacker can log in to the victim's account and control it.

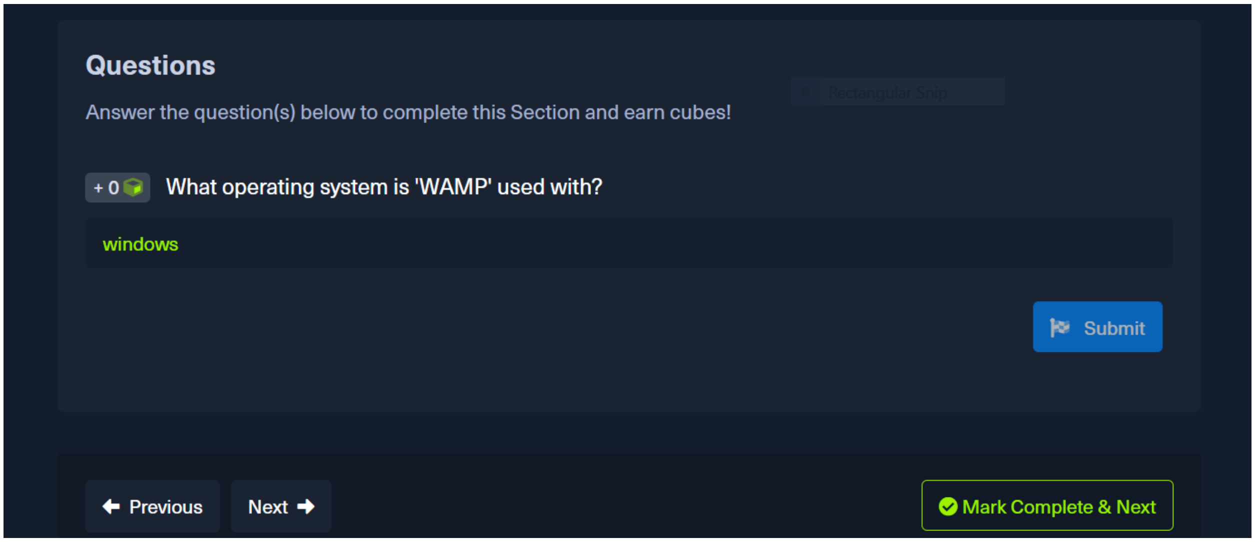
4. Back End Components

a. Back End Servers

A back-end server is the hardware and operating system on the back end that hosts all of the applications necessary to run the web application**.**

There are many popular combinations of "stacks" for back-end servers, which contain a specific set of back end components. Some common examples include:

Combinations	**Components**
LAMP	Linux, Apache, MySQL, and PHP.
WAMP	Windows, Apache, MySQL, and PHP.
WINS	Windows, IIS, .NET, and SQL Server
MAMP	macOS, Apache, MySQL, and PHP.
XAMPP	Cross-Platform, Apache, MySQL, and PHP/PERL.



Questions

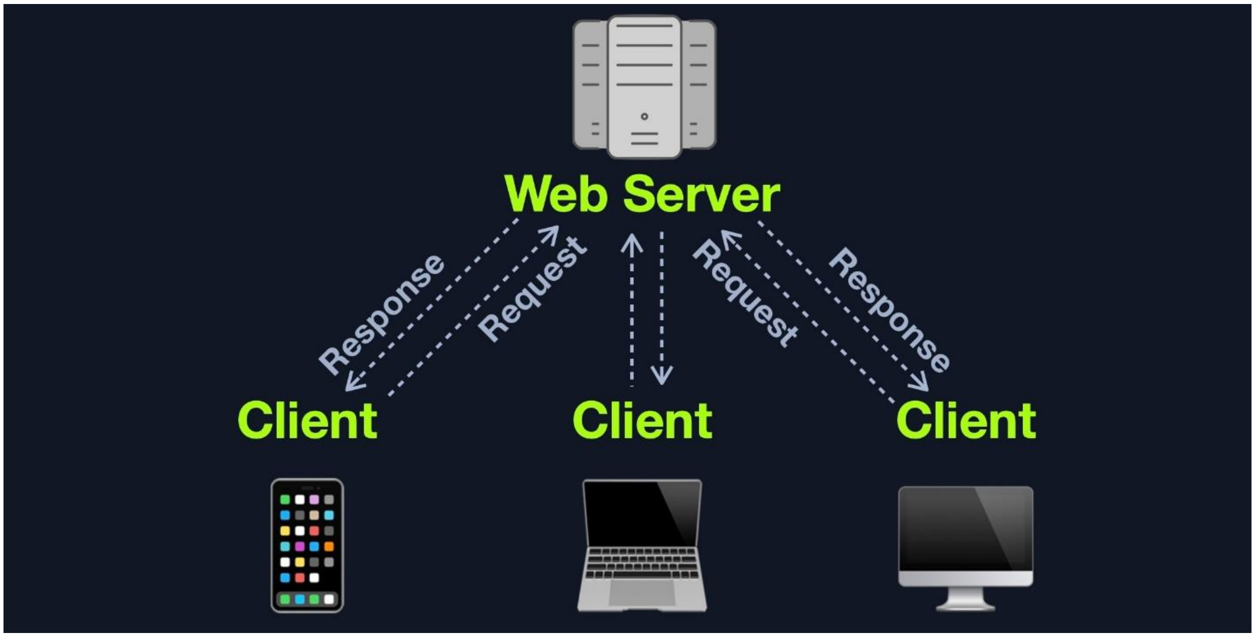
Answer the question(s) below to complete this Section and earn cubes!

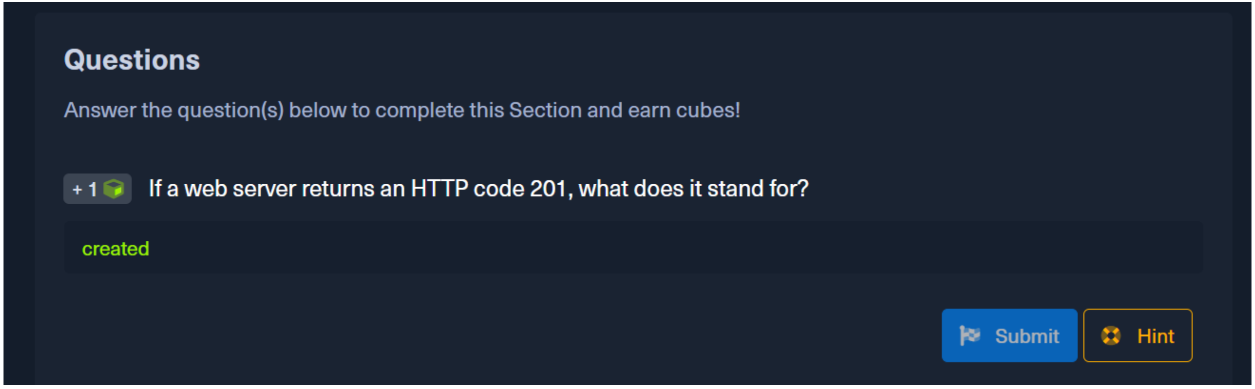
+ 0 What operating system is 'WAMP' used with?

windows

b. Web Servers

A web server is an application that runs on the back end server, which handles all of the HTTP traffic from the client-side browser, routes it to the requested pages, and finally responds to the client-side browser.





Questions

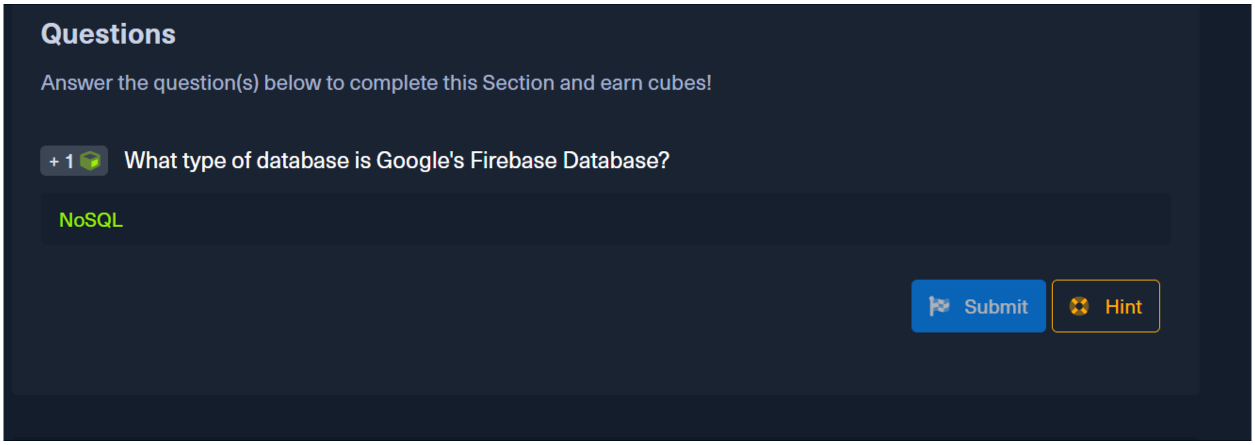
Answer the question(s) below to complete this Section and earn cubes!

+1 If a web server returns an HTTP code 201, what does it stand for?

created

c. Databases

- There are many different types of databases, each of which fits a certain type of use.
- Relational (SQL) databases store their data in tables, rows, and columns. Eg **MySQL, MSSQL, Oracle, PostgreSQL, SQLite, MariaDB, Amazon Aurora, and Azure SQL.**
- Non-relational (NoSQL) database does not use tables, rows, columns, primary keys, relationships, or schemas. Instead, a NoSQL database stores data using various storage models, depending on the type of data stored. E.g MongoDB, **ElasticSearch, Apache Cassandra, Redis, Neo4j, CouchDB, and Amazon DynamoDB.**



Questions

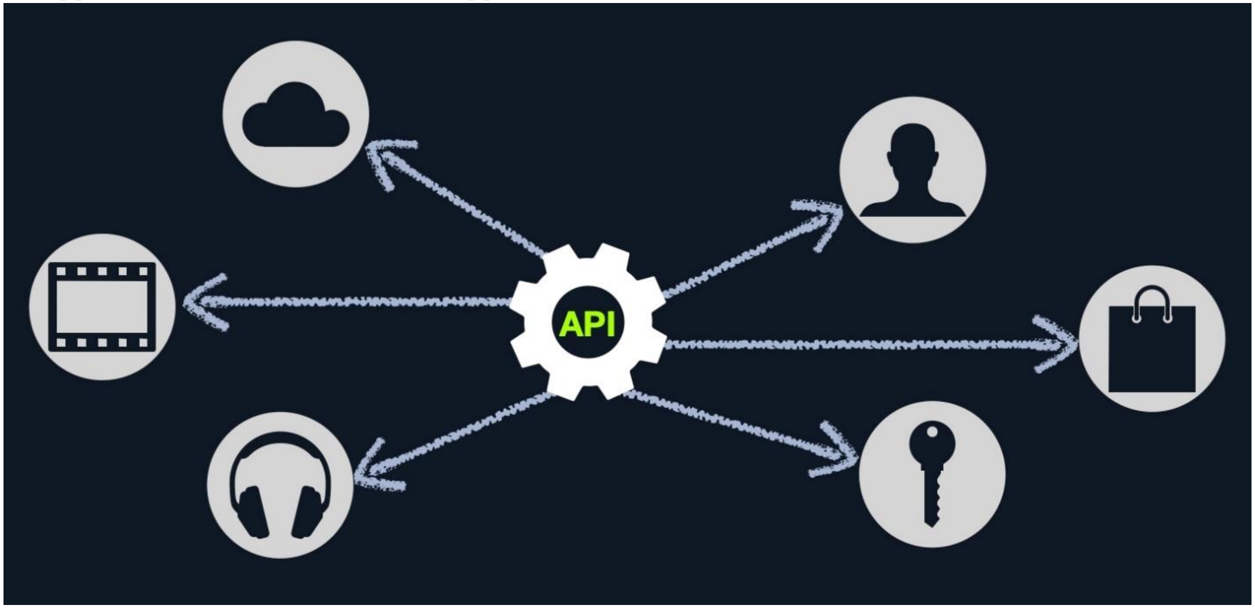
Answer the question(s) below to complete this Section and earn cubes!

+1 What type of database is Google's Firebase Database?

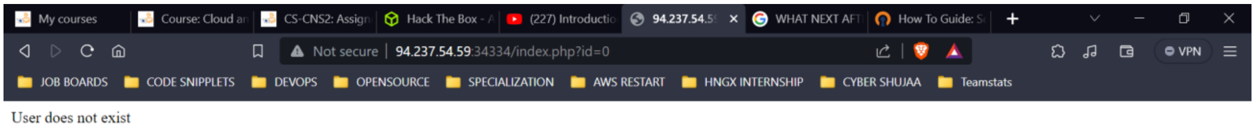
NoSQL

d. Development Frameworks & APIs

An API (Application Programming Interface) is an interface within an application that specifies how the application can interact with other applications.

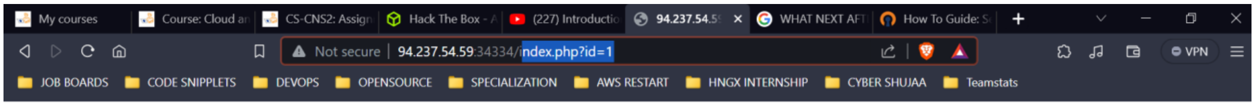


- To enable the use of APIs within a web application, the developers have to develop this functionality on the back end of the web application by using the API standards like **SOAP or REST.**
- **SOAP (Simple Objects Access)** standard shares data through XML, where the request is made in XML through an HTTP request, and the response is also returned in XML. Front end components are designed to parse this XML output properly
- REST (Representational State Transfer) standard shares data through the URL path 'i.e. search/users/1', and usually returns the output in JSON format 'i.e. userid 1'.



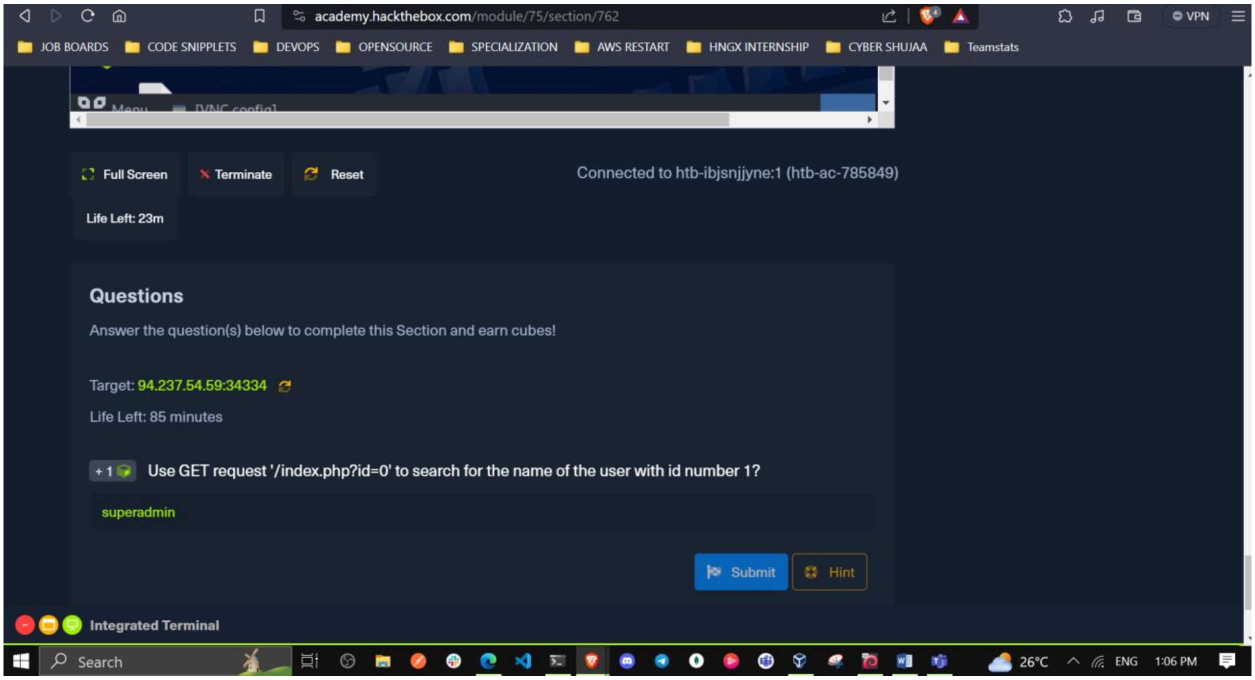
Not secure | 94.237.54.59:34334/index.php?id=0

User does not exist



Not secure | 94.237.54.59:34334/index.php?id=1

superadmin



academy.hackthebox.com/module/75/section/762

Connected to htb-ibjsnjjyne:1 (htb-ac-785849)

Life Left: 23m

Questions

Answer the question(s) below to complete this Section and earn cubes!

Target: 94.237.54.59:34334

Life Left: 85 minutes

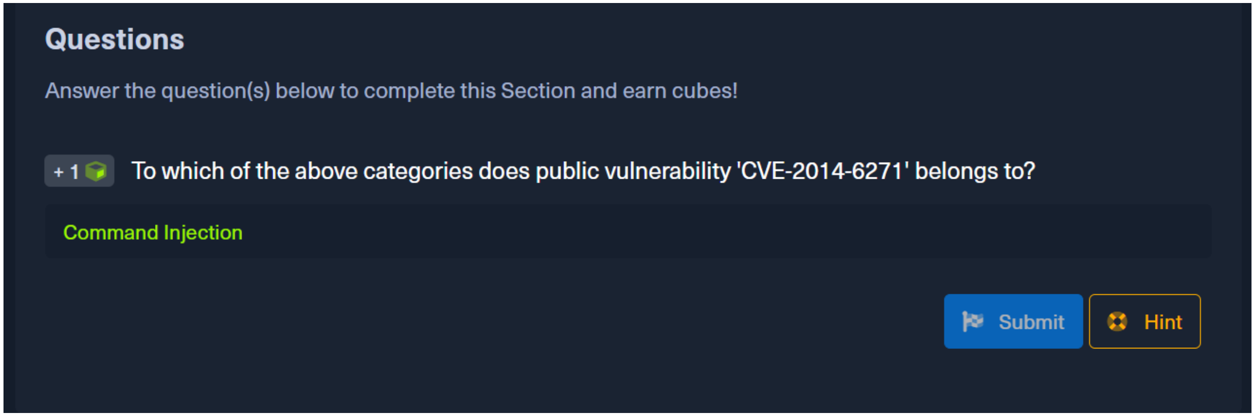
+1 Use GET request '/index.php?id=0' to search for the name of the user with id number 1?

superadmin

Common Web Vulnerabilities

The below examples are some of the most common vulnerability types for web applications, part of OWASP Top 10 vulnerabilities for web applications.

- Broken Authentication/Access Control
- Malicious File Upload
- Command Injection
- SQL Injection (SQLi)



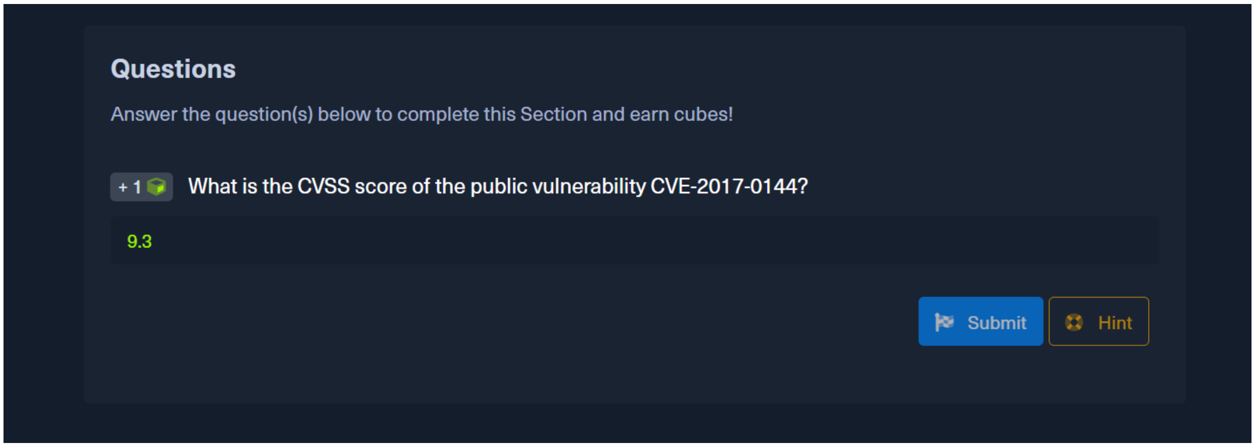
Questions

Answer the question(s) below to complete this Section and earn cubes!

+1 To which of the above categories does public vulnerability 'CVE-2014-6271' belongs to?

Command Injection

a. **Public Vulnerabilities**

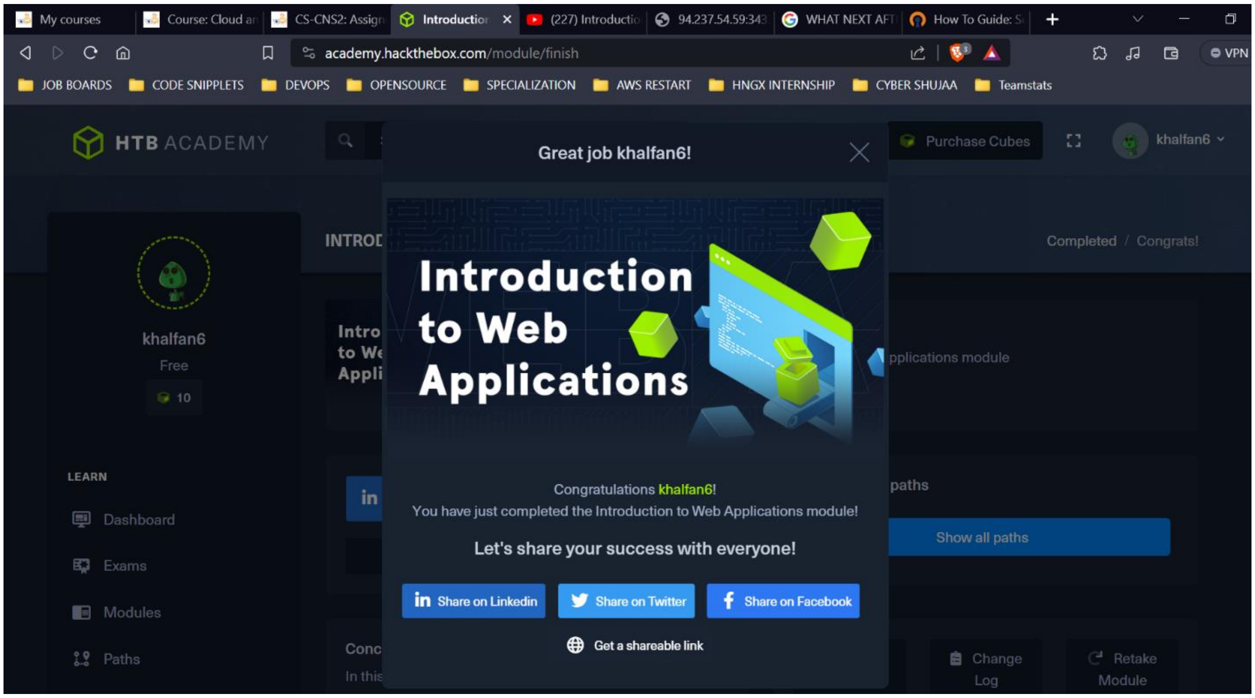


Questions

Answer the question(s) below to complete this Section and earn cubes!

+ 1 What is the CVSS score of the public vulnerability CVE-2017-0144?

9.3



academy.hackthebox.com/module/finish

Great job khalfan6!

Introduction to Web Applications

Congratulations khalfan6!
You have just completed the Introduction to Web Applications module!

Let's share your success with everyone!

https://academy.hackthebox.com/achievement/785849/75

CONCLUSION

Web applications are the backbone of the modern internet, powering everything from e-commerce websites to social media platforms. However, they are also a prime target for attackers, who exploit vulnerabilities to steal data, disrupt operations, and launch further attacks. To protect web applications, it is important to understand the common security risks and how to mitigate them. This includes securing the front-end and back-end components, as well as addressing public vulnerabilities and following best practices in web application development and testing.