Introduction Web Application on the Hack the box

Here is the link to my completed module:

https://academy.hackthebox.com/achievement/1917469/75

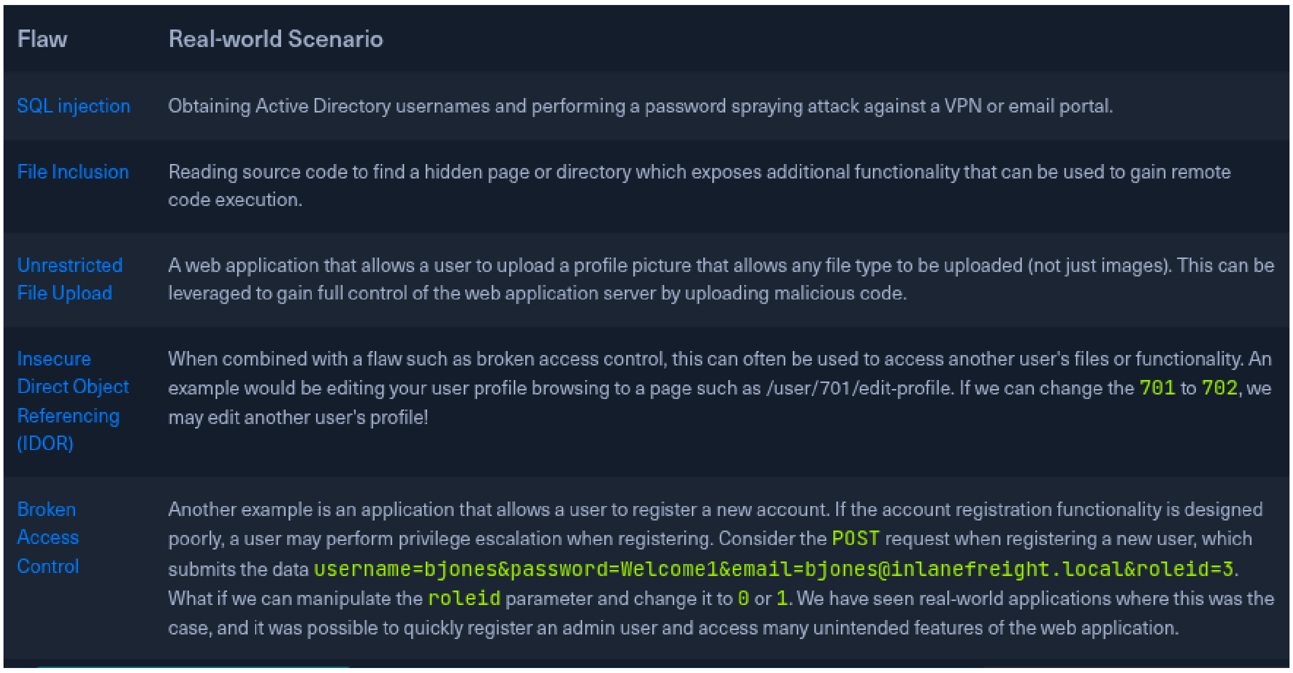
Overview

Web applications are interactive applications that run on web browser and usually adopts client server architecture to run and handle interactions I.e they have frontend and backend.

Web applications are also prone to attacks as they are accessible by anyone who has internet connection and from any country. These attacks can lead to significant losses and massive business interruption.

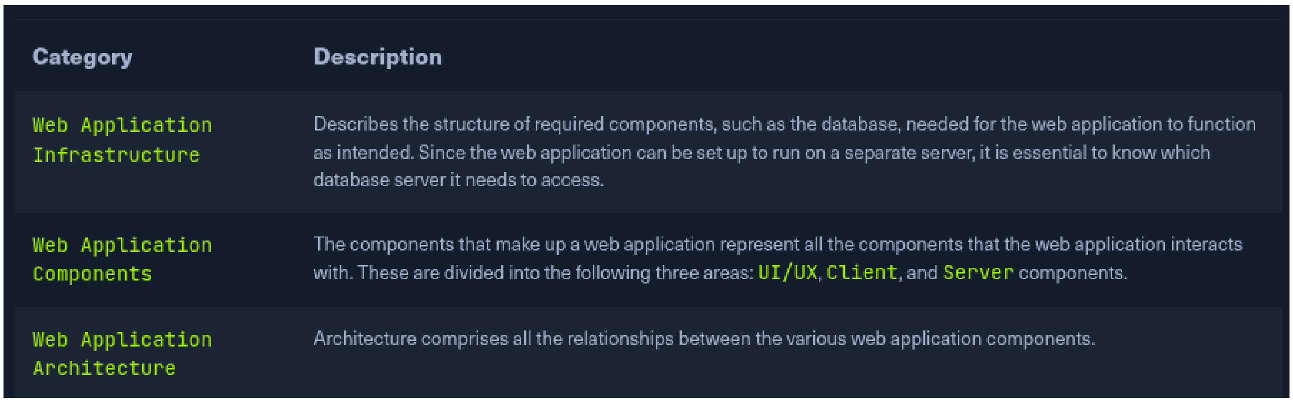
Any organization looking to secure internet facing web applications should undergo frequent web applications tests and implement secure coding practices at every development life cycle stage.

Examples of web applications attacks and impacts are as follows:



Flaw	Real-world Scenario
SQL injection	Obtaining Active Directory usernames and performing a password spraying attack against a VPN or email portal.
File Inclusion	Reading source code to find a hidden page or directory which exposes additional functionality that can be used to gain remote code execution.
Unrestricted File Upload	A web application that allows a user to upload a profile picture that allows any file type to be uploaded (not just images). This can be leveraged to gain full control of the web application server by uploading malicious code.
Insecure Direct Object Referencing (IDOR)	When combined with a flaw such as broken access control, this can often be used to access another user's files or functionality. An example would be editing your user profile browsing to a page such as /user/701/edit-profile. If we can change the `701` to `702`, we may edit another user's profile!
Broken Access Control	Another example is an application that allows a user to register a new account. If the account registration functionality is designed poorly, a user may perform privilege escalation when registering. Consider the `POST` request when registering a new user, which submits the data `username=bjones&password=Welcome1&email=bjones@inlanefreight.local&roleid=3`. What if we can manipulate the `roleid` parameter and change it to `0` or `1`. We have seen real-world applications where this was the case, and it was possible to quickly register an admin user and access many unintended features of the web application.

The layout of web applications consists of many different layers that can be summarized in following categories:



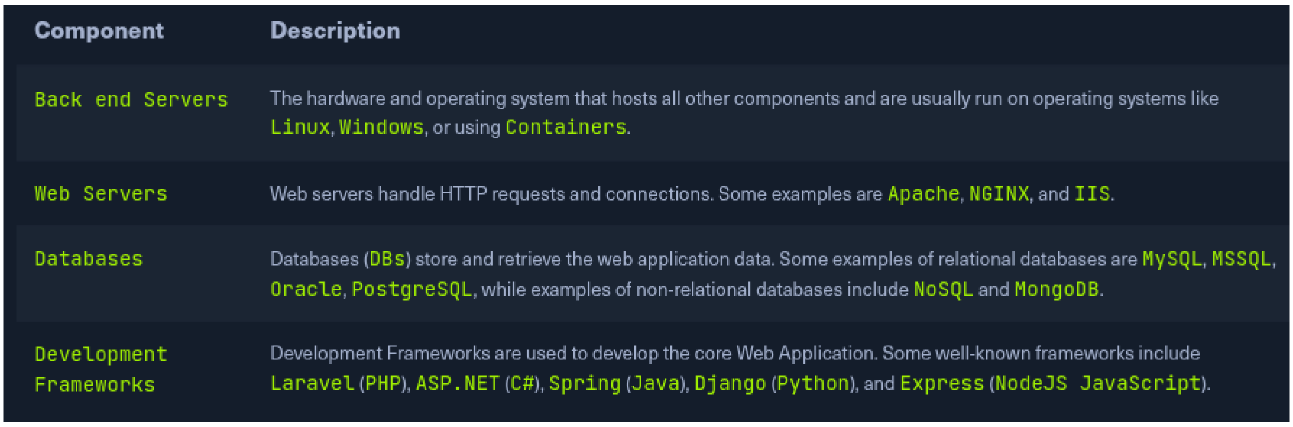
Category	Description
`Web Application Infrastructure`	Describes the structure of required components, such as the database, needed for the web application to function as intended. Since the web application can be set up to run on a separate server, it is essential to know which database server it needs to access.
`Web Application Components`	The components that make up a web application represent all the components that the web application interacts with. These are divided into the following three areas: `UI/UX`, `Client`, and `Server` components.
`Web Application Architecture`	Architecture comprises all the relationships between the various web application components.

Front end vs Back end

Front end of a web applications contains the user's components directly through their web browsers and makes up the source code of the web page we view when visiting a web applications and usually includes HTML,CSS and Javascripts.

The back end of a web application drives all of the core web application functionalities, all of which is executed at the back end server, which processes everything required for the web application to run correctly.

It consists of of 4 components(back end):

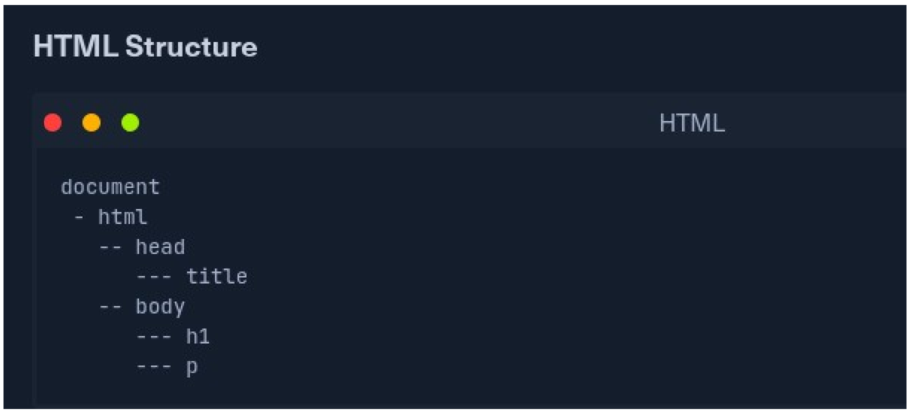


Component	Description
`Back end Servers`	The hardware and operating system that hosts all other components and are usually run on operating systems like `Linux`, `Windows`, or using `Containers`.
`Web Servers`	Web servers handle HTTP requests and connections. Some examples are `Apache`, `NGINX`, and `IIS`.
`Databases`	Databases (`DBs`) store and retrieve the web application data. Some examples of relational databases are `MySQL`, `MSSQL`, `Oracle`, `PostgreSQL`, while examples of non-relational databases include `NoSQL` and `MongoDB`.
`Development Frameworks`	Development Frameworks are used to develop the core Web Application. Some well-known frameworks include `Laravel` (`PHP`), `ASP.NET` (`C#`), `Spring` (`Java`), `Django` (`Python`), and `Express` (`NodeJS JavaScript`).

HTML

This is at very core of any web pages we see on the internet and contains each page's basic element including titles,forms,images and other elements.

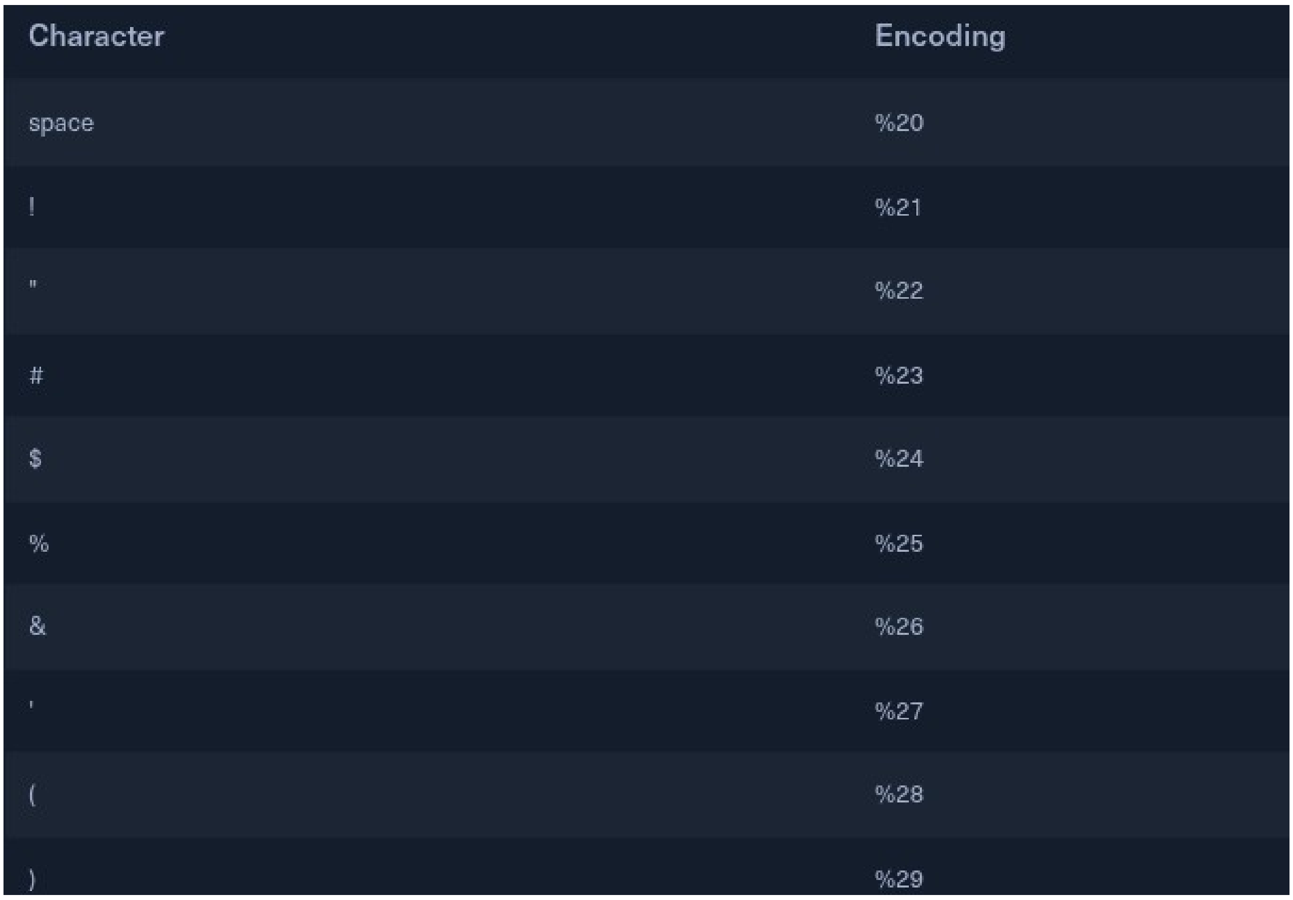
It has simple structure which look like this:


HTML Structure

```
document
 - html
   -- head
      --- title
   -- body
      --- h1
      --- p
```

Another important concept I found about html is url encoding which for a browser to properly display a page's contents it has to know its charset. URL encoding replaces unsafe ASCII characters with% sysmbol followed by two hexadecimal digits.

For example a single quote character '" is encoded to '%27'which can be understood by browsers as a single quote; below are some of the common characters:



Character	Encoding
space	%20
!	%21
"	%22
#	%23
$	%24
%	%25
&	%26
'	%27
(	%28
)	%29

Question:

What is the HTML tag used to show an image?

Answer is <img>

Cascading style sheet (css)

CSS is used along HTML to format and style elements.
CSS has its framework which are designed to be used with javascript and for wide use within the web application and contains elements usually required within modern web applications and common used framework is bootstrap.

Question:

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

Answer is text-align: left;

Javascript

This is widely used in web development and mobile development and in web pages to make it dynamic and interactive.
It makes web page to be viewed in real time,dynamically updating content in real time,accepting and processing user input.
It also contains framework which are vue,angular,react and jQuery

Sensitive data exposure

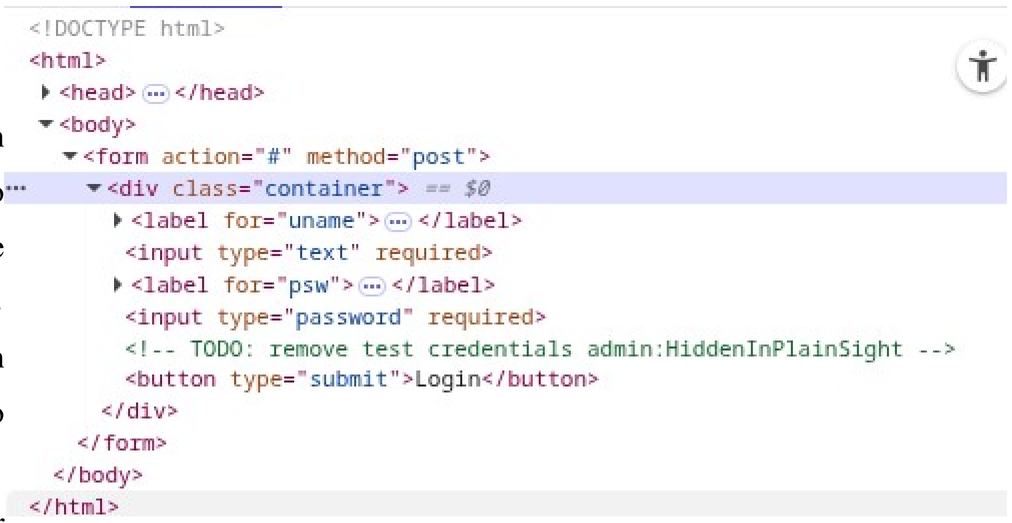
This refers to the availability of sensitive data in clear text to the end user and is usually found in the source code of the web page.
This can be avoided by reviewing the code that will be visible to end users through the page source.

Question:

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

Answer HiddenInPlainSight

This would allow an attacker to use these credentials to gain access to the website or system.



```
<!DOCTYPE html>
<html>
 ▸<head> … </head>
 ▾<body>
   ▾<form action="#" method="post">
…    ▾<div class="container"> == $0
       ▸<label for="uname"> … </label>
        <input type="text" required>
       ▸<label for="psw"> … </label>
        <input type="password" required>
        <!-- TODO: remove test credentials admin:HiddenInPlainSight -->
        <button type="submit">Login</button>
      </div>
    </form>
  </body>
</html>
```

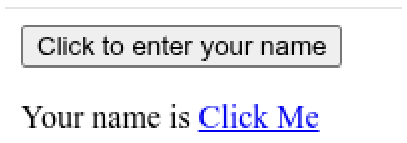
HTML Injection

This occurs when unfiltered user input is displayed on the page and 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.

Question:

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>

answer Your name is Click Me

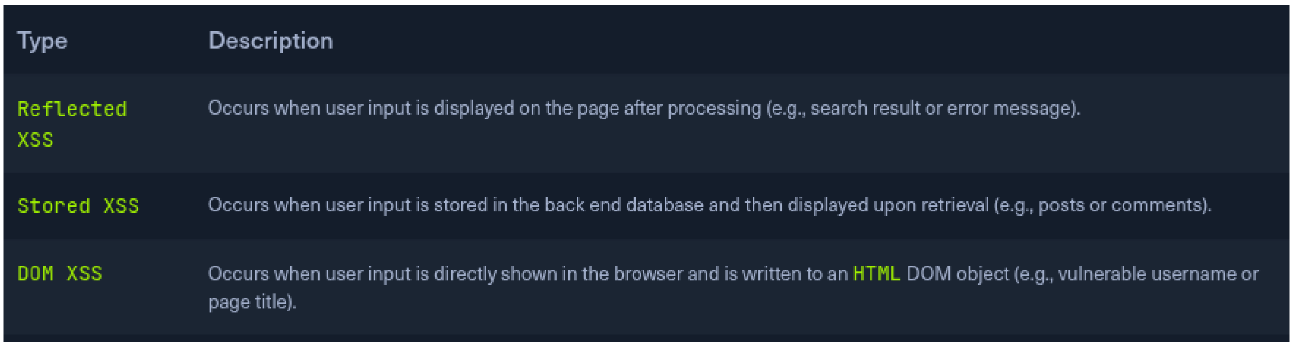


Your name is Click Me

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.

Types of XSS are:

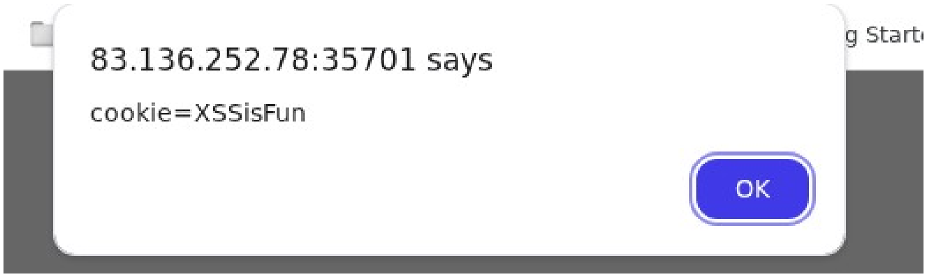


Type	Description
`Reflected XSS`	Occurs when user input is displayed on the page after processing (e.g., search result or error message).
`Stored XSS`	Occurs when user input is stored in the back end database and then displayed upon retrieval (e.g., posts or comments).
`DOM XSS`	Occurs when user input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

Question:

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

Answer XSSisFun

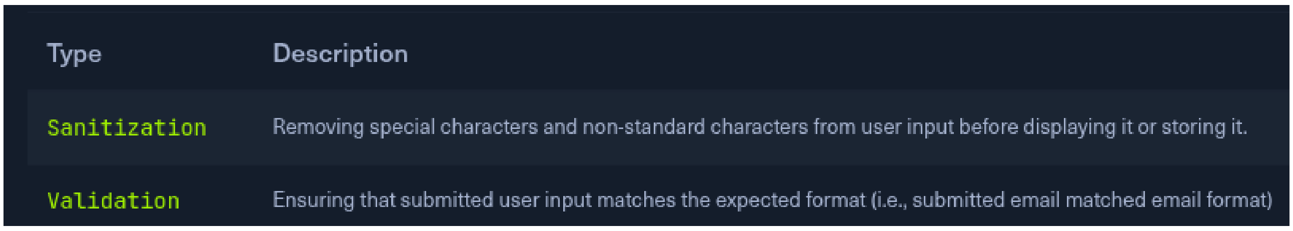


83.136.252.78:35701 says

cookie=XSSisFun

Cross-Site Request Forgery(CSRF)

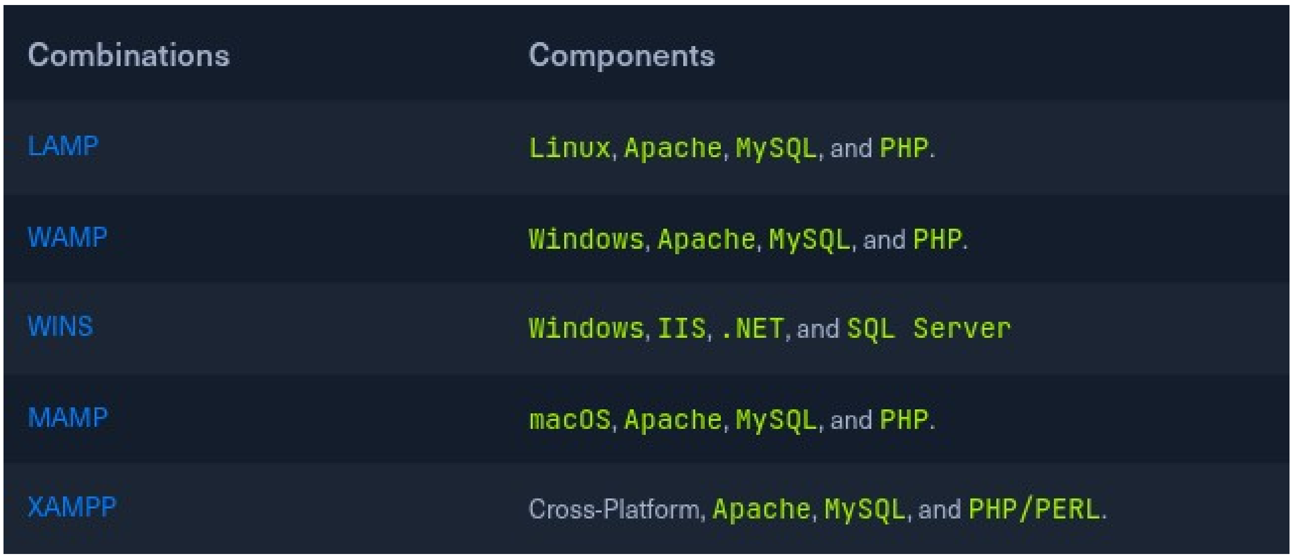
CSRF attacks may utilize XSS vulnerabilities to perform certain queries, and API calls on a web application that the victim is currently authenticated to. This would allow the attacker to perform actions as the authenticated user.
A common CSRF attack to gain higher privileged access to a web application 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.
It can also be leveraged to attack admins and gain access to their accounts.
To prevent this it is important to filter and sanitize user input on the front end before it reaches the back end and especially if this code may be displayed directly on the client-side without communicating with the back end.
Two main controls must be applied when accepting user input:



Type	Description
`Sanitization`	Removing special characters and non-standard characters from user input before displaying it or storing it.
`Validation`	Ensuring that submitted user input matches the expected format (i.e., submitted email matched email format)

Back end servers

This is the hardware and OS on the back end that hosts all of the applications necessary to run the web application and contains the web server,database and development framework.



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.

Question:

What operating system is 'WAMP' used with?

Windows

This can be shown by the above figure.

Web Servers

This 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. They usually runs on TCP port 80 or 443 and are responsible for connecting end users to various part of the web application.

This part also explained various HTTP responses codes and some common used web servers which are Apache,IIS and NGINX.

Question:

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

Response = created

Databases

These are used to store various content and information related to web applications.

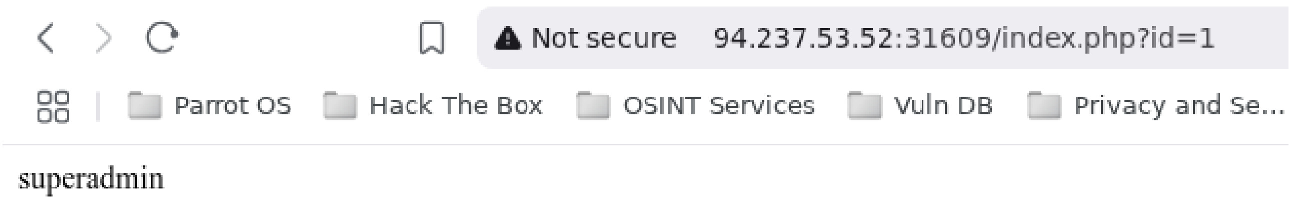
This part also explained various types of database such as relational and non relational sql and one thing developers choose database is by looking at their characteristic such as speed in storing and retrieving data,size and scalability when storing large amounts of data.

Question:

What type of database is Google's Firebase Database? NoSQL

Development Framework & APIs

These framework helps in developing core web application files and functionalities.
APIs helps to connect the front end and back end to be able to send data back and forth between the front end and back end components and carry out various functions within the web applications.
Question:
Use GET request '/index.php?id=0' to search for the name of the user with id number 1?
answer is superadmin


Not secure 94.237.53.52:31609/index.php?id=1

superadmin

Common Web Vulnerabilities

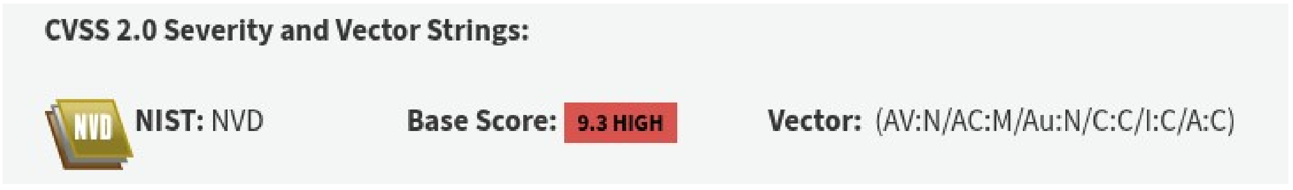
This part explained the following vulnerabilities which are broken authentication/access controls,malicious file upload,sql injection and command injection.
Question:
To which of the above categories does public vulnerability 'CVE-2014-6271' belongs to?

To solve this I searched the CVE in the CVE database which I found it belongs to Command Injection.
Answer is Command Injection

Public Vulnerabilities

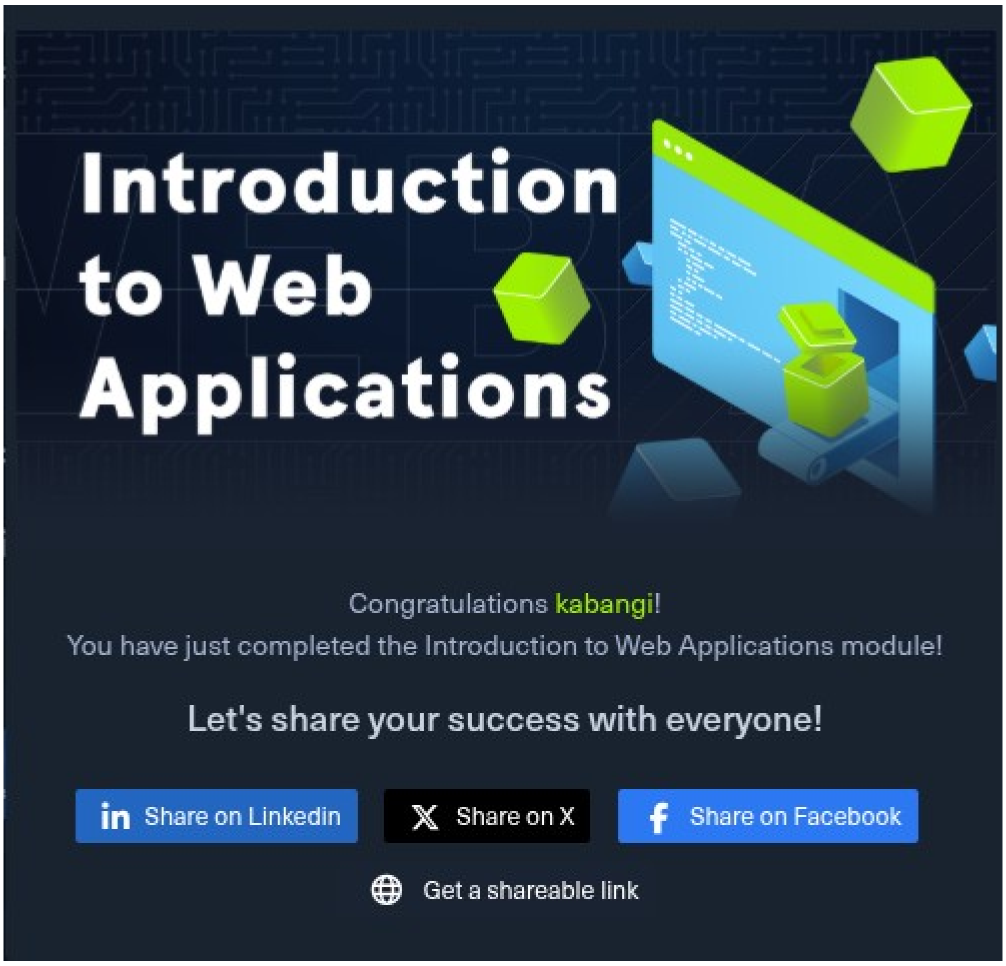
This mostly concentrate on public CVE and their scoring like CVSS v2 and CVSS v3.
Question:
What is the CVSS v2.0 score of the public vulnerability CVE-2017-0144?

9.3which was high


CVSS 2.0 Severity and Vector Strings:

NIST: NVD **Base Score:** 9.3 HIGH **Vector:** (AV:N/AC:M/Au:N/C:C/I:C/A:C)

That was the last part of the module and I successfully finished it.


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

Let's share your success with everyone!

Conclusion

This module provided a valuable reminder of the various programming languages and frameworks commonly used in web application development. It also allowed me to step into the mindset of an attacker, helping me understand the vulnerabilities they look for and the techniques they use to exploit systems. Most importantly, it highlighted the critical role developers play in securing applications and the measures they must implement to prevent unauthorized access and reduce security risks.