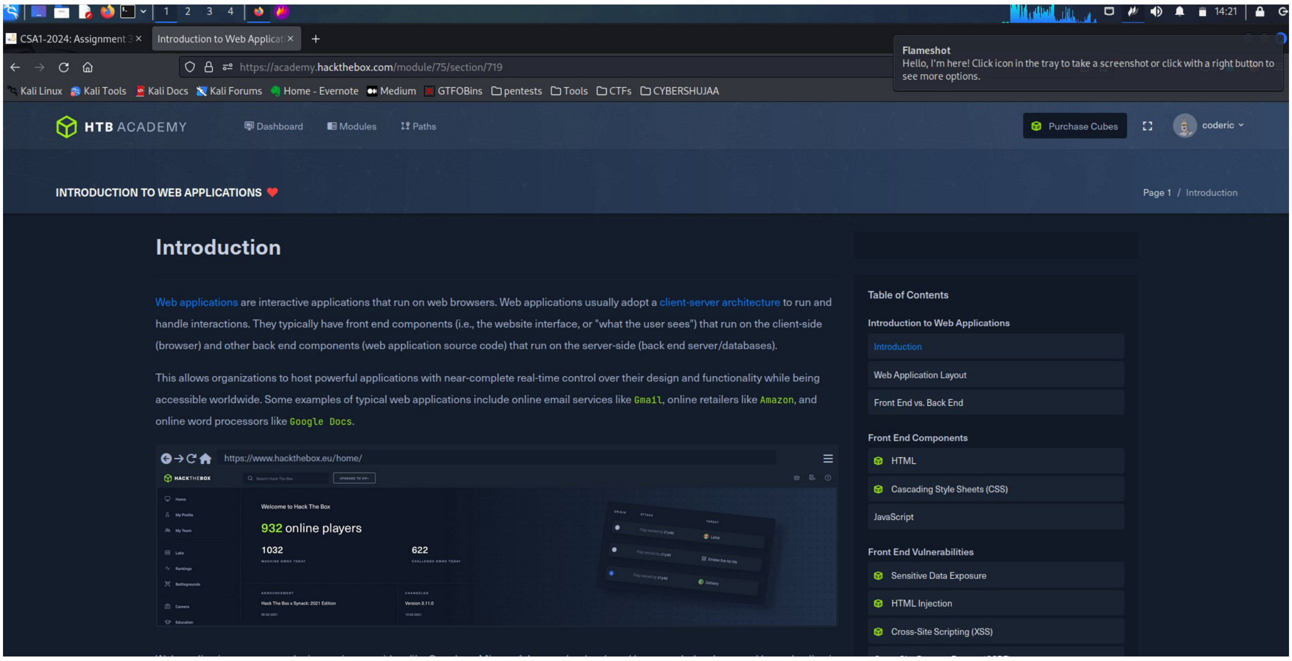
Eric Mwenda

Introduction to Web Applications

https://academy.hackthebox.com/achievement/596337/75



Flameshot
Hello, I'm here! Click icon in the tray to take a screenshot or click with a right button to see more options.

Introduction

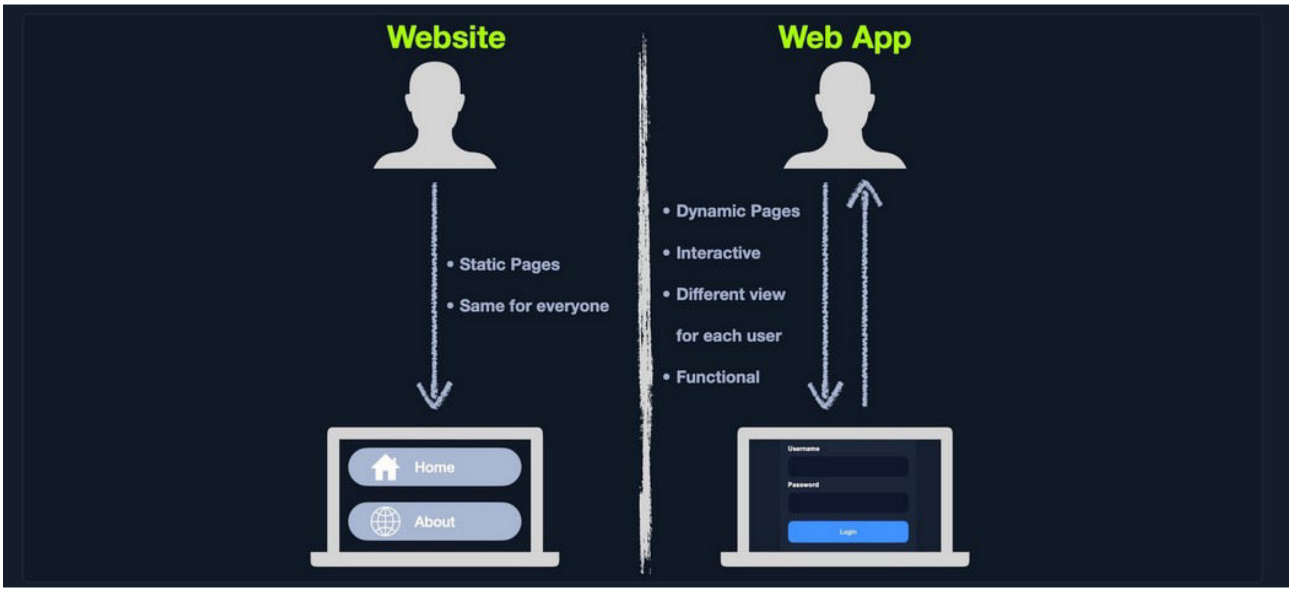
Web applications are interactive applications that run on web browsers. Web applications usually adopt a client-server architecture to run and handle interactions. They typically have front end components (i.e., the website interface, or "what the user sees") that run on the client-side (browser) and other back end components (web application source code) that run on the server-side (back end server/databases).

This allows organizations to host powerful applications with near-complete real-time control over their design and functionality while being accessible worldwide. Some examples of typical web applications include online email services like `Gmail`, online retailers like `Amazon`, and online word processors like `Google Docs`.

Web Applications vs Websites.

From this section am well able to differentiate between a web application and a website. A web application presents dynamic content based on the user interaction whereas a website is a static page that represents same information for all users, it does not change.

Web Applications have several vulnerabilities that can be exploited. Some of this vulnerabilities include:- SQL Injections, File Inclusion on the source code, Unrestricted File Uploads, Broken Access Controls among many other.



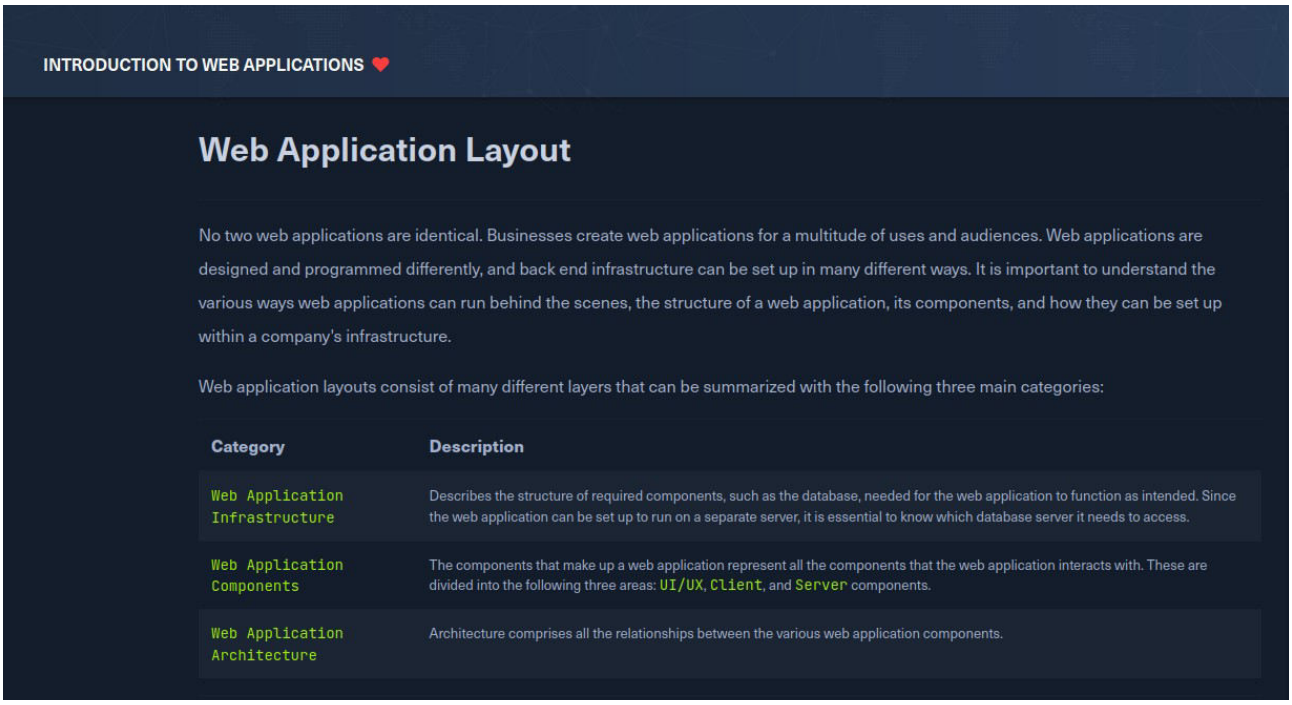
Web Application Layout.

Web Applications are not identical, every web application has a different design and program differently as well. Web Application Layouts have 3 main categories:-

- Web Application Infrastructure – Also called modules. Common types include client-server, one server, many servers- One database, Many servers-Many Databases.

- Web Application Components – Components of the modules include, Client, Server, Services, Functions.

- Web Application Architecture – The components are then divided into 3 different layers:- Presentation Layer, Application Layer, Data Layer.



INTRODUCTION TO WEB APPLICATIONS ❤

Web Application Layout

No two web applications are identical. Businesses create web applications for a multitude of uses and audiences. Web applications are designed and programmed differently, and back end infrastructure can be set up in many different ways. It is important to understand the various ways web applications can run behind the scenes, the structure of a web application, its components, and how they can be set up within a company's infrastructure.

Web application layouts consist of many different layers that can be summarized with the following three main 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 and Back End.

In this section I got to understand that Full Stack means both front end and 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>
```

- Front end contains user components, whereby the user interact with the application directly through their web browsers. Usually the web application front end includes HTML, CSS and JavaScript which is interpreted in real-time in our browsers.

- Back end of a web application drives all the core application functionalities, all which are executed from a back end server. This is the part where users never see or interact directly with.

- I have also learnt during web application development, developers should be very keen to avoid various mistakes that leave an entry point for attackers to exploit.

This various mistakes include:-

1. Permitting invalid data.

2. Creating weak administrators passwords.

3. Remote file inclusion.

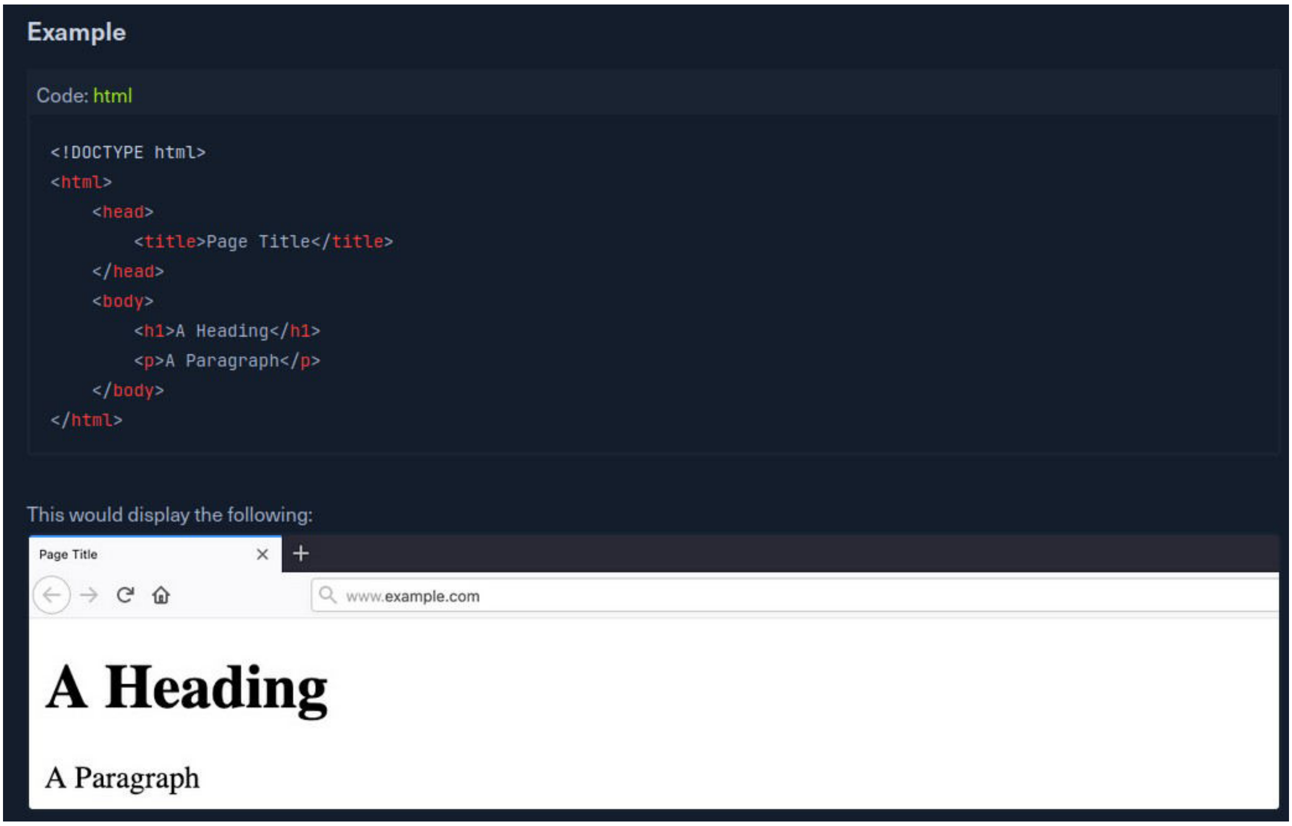
4. Storing unencrypted data.

5. Unverified SQL Injections.

Etc.

HTML

HTML is an abbreviation for Hyper Text Mark-up Language. HTML is the very core of any web page in the internet.



Example

Code: html

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

This would display the following:

A Heading

A Paragraph

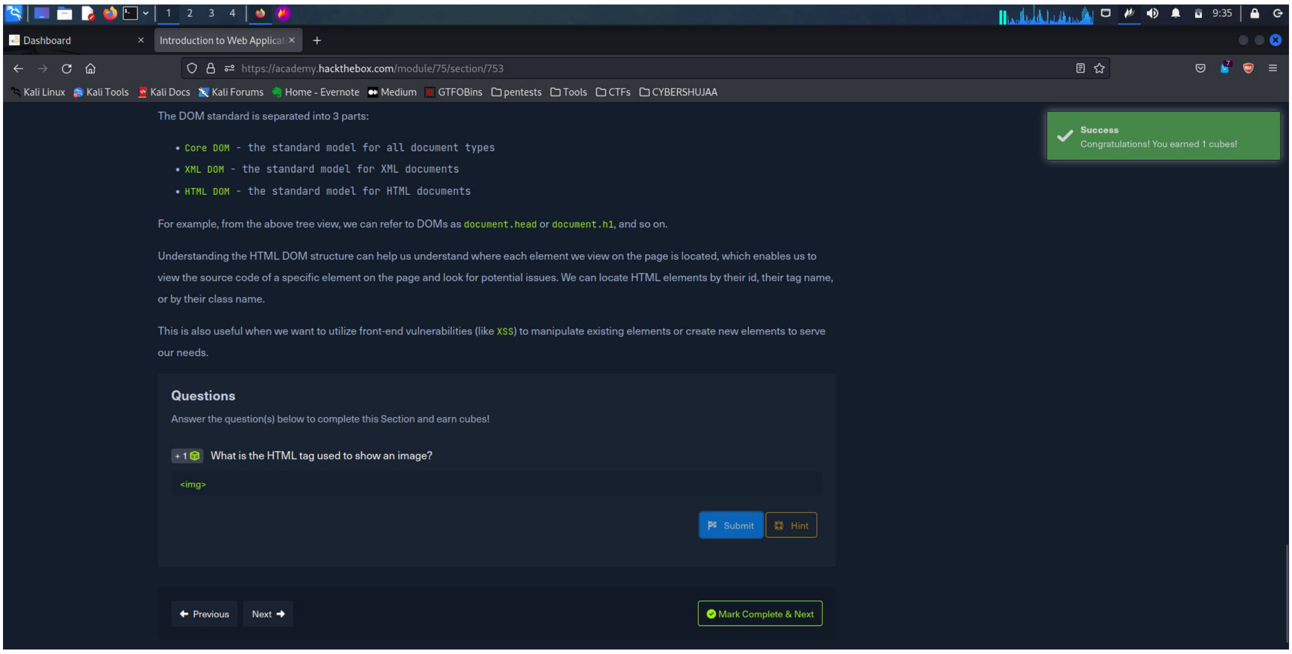
HTML has elements that have open and closed tags. Eg for a paragraph it is;

<p> This is an example of HTML Element </p>

HTML has several online encoders/decoders, a good example is the Burpsuit.

Questions

Answer the question(s) below to complete this Section and earn cubes! ANS: **<img>**



Success
Congratulations! You earned 1 cubes!

The DOM standard is separated into 3 parts:

- `Core DOM` - the standard model for all document types
- `XML DOM` - the standard model for XML documents
- `HTML DOM` - the standard model for HTML documents

For example, from the above tree view, we can refer to DOMs as `document.head` or `document.h1`, and so on.

Understanding the HTML DOM structure can help us understand where each element we view on the page is located, which enables us to view the source code of a specific element on the page and look for potential issues. We can locate HTML elements by their id, their tag name, or by their class name.

This is also useful when we want to utilize front-end vulnerabilities (like XSS) to manipulate existing elements or create new elements to serve our needs.

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>

Cascading Style Sheets (CSS)

CSS is the style sheet language used alongside HTML to format and set the style of HTML elements. CSS defines the style for each class or type of HTML.

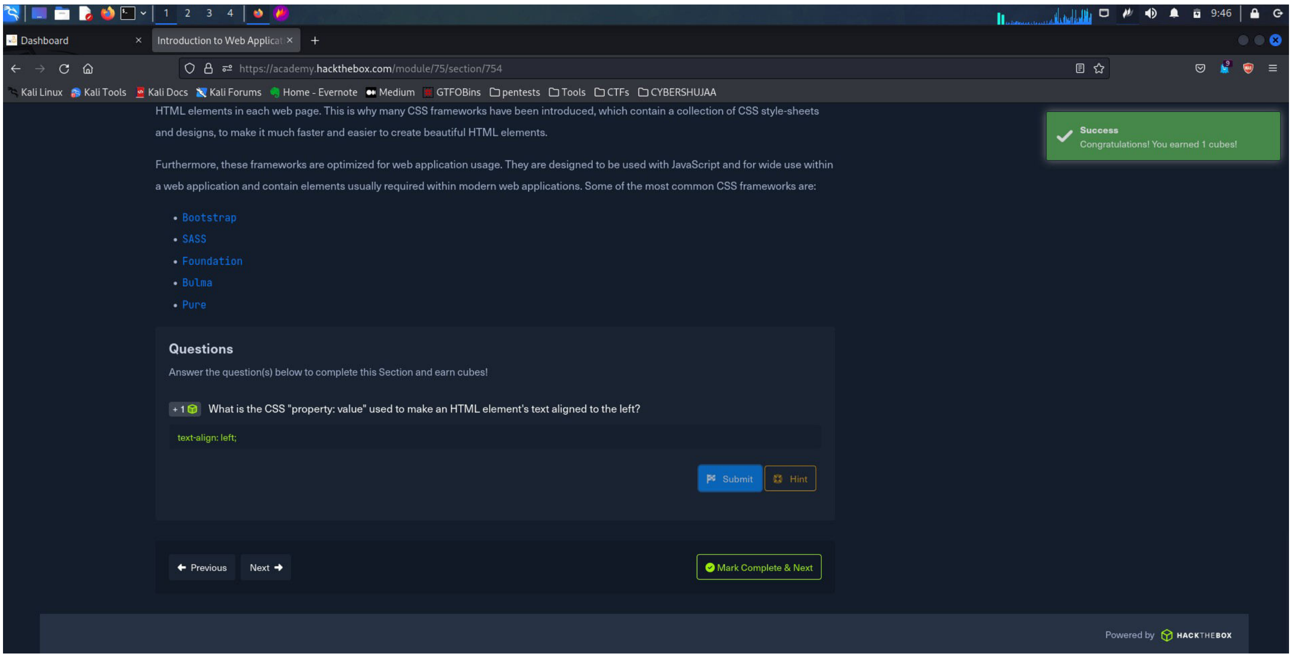
A simple code in CSS is:-

```
body {
  background-color: blue;
}
h1 {
  color: white;
  text-align: left;
}
```

Question

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

ANS: **text-align: left;**



HTML elements in each web page. This is why many CSS frameworks have been introduced, which contain a collection of CSS style-sheets and designs, to make it much faster and easier to create beautiful HTML elements.

Furthermore, these frameworks are optimized for web application usage. They are designed to be used with JavaScript and for wide use within a web application and contain elements usually required within modern web applications. Some of the most common CSS frameworks are:

- `Bootstrap`
- `SASS`
- `Foundation`
- `Bulma`
- `Pure`

Questions

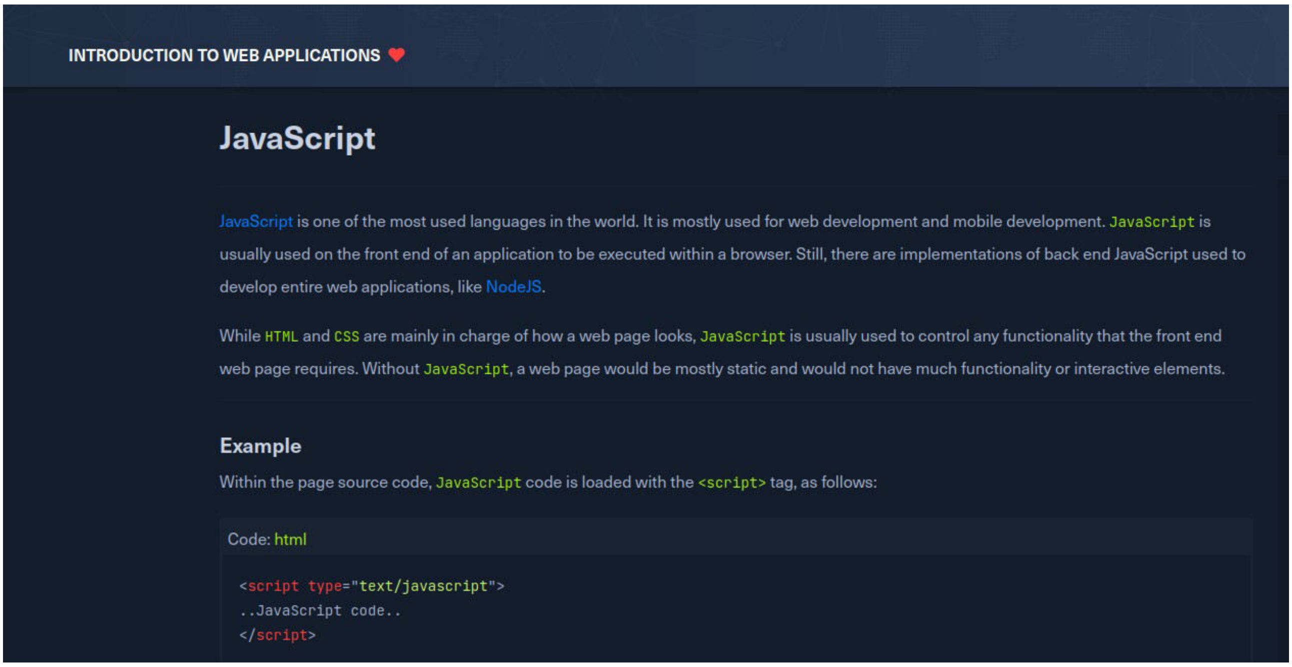
Answer the question(s) below to complete this Section and earn cubes!

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

text-align: left;

Success
Congratulations! You earned 1 cubes!

JavaScript



INTRODUCTION TO WEB APPLICATIONS

JavaScript

JavaScript is one of the most used languages in the world. It is mostly used for web development and mobile development. `JavaScript` is usually used on the front end of an application to be executed within a browser. Still, there are implementations of back end JavaScript used to develop entire web applications, like NodeJS.

While `HTML` and `CSS` are mainly in charge of how a web page looks, `JavaScript` is usually used to control any functionality that the front end web page requires. Without `JavaScript`, a web page would be mostly static and would not have much functionality or interactive elements.

Example

Within the page source code, `JavaScript` code is loaded with the `<script>` tag, as follows:

Code: html

```html
<script type="text/javascript">
..JavaScript code..
</script>
```

This is one of the most popular language used in the world. In most cases Javascript is used in front end of applications but recently there are implementations of back end javascript used to develop entire web applications like NodeJS.

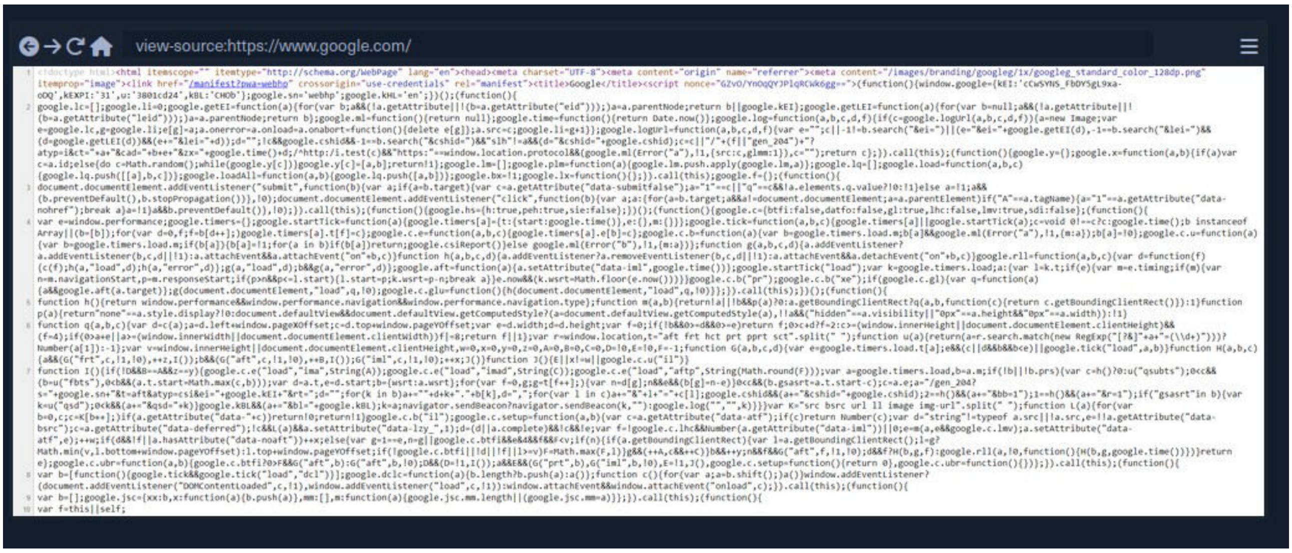
I got to understand that HTML and CSS are mainly in charge of how a web page looks while JavaScript is usually used to control any functionality that the front end web page requires. Without JavaScript in use, a web page would be mostly static and would not have much functionality or interactive elements.

Sensitive Data Exposure

Sensitive Data Exposure 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.

This is the HTML source code of the application, not to be confused with the back end code that is typically only accessible on the server itself.

To view the page source code most browsers you type CTRL + U or view the page source using Burp Suite.

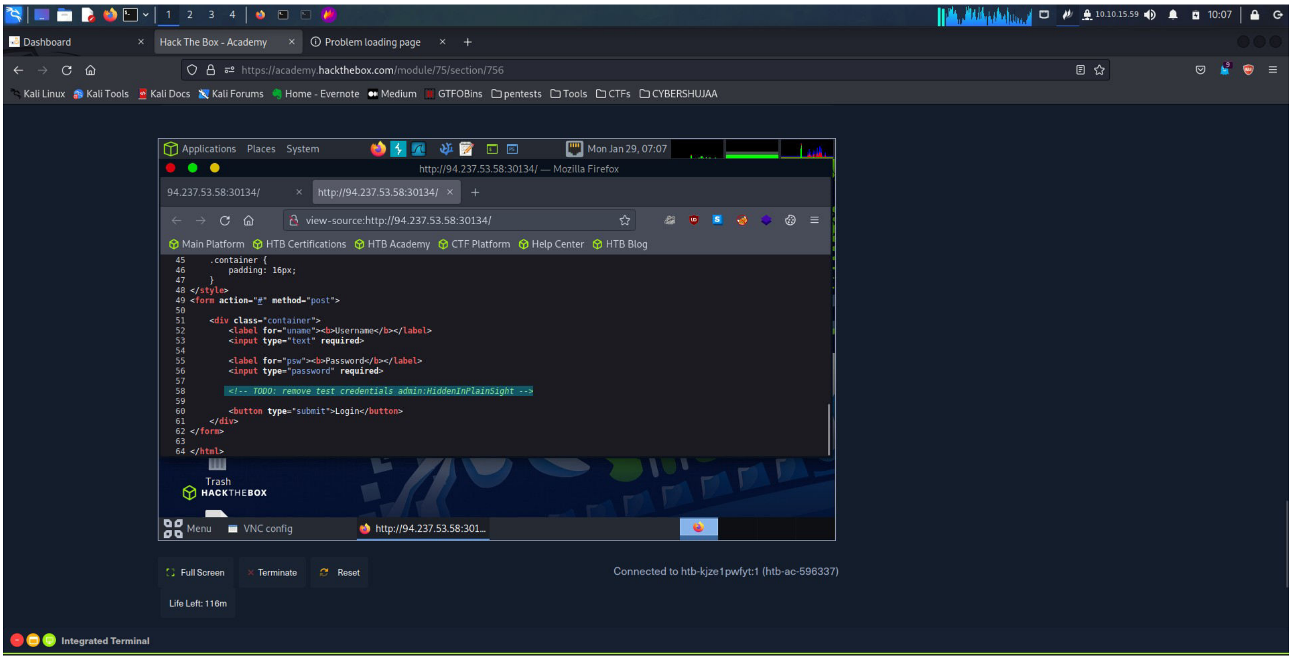


```
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"><head><meta charset="UTF-8"><meta content="origin" name="referrer"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png"
itemprop="image"><link href="/manifest?pwa=webhp" crossorigin="use-credentials" rel="manifest"><title>Google</title><script nonce="GZvO/YnOqQYJPlqRCWk6gg==">(function(){window.google={kEI:'cCWSYNS_FbDY5gL9xa-
oDQ',kEXPI:'31',u:'3801cd24',kBL:'CHOb'};google.sn='webhp';google.kHL='en';})();(function(){
google.lc=[];google.li=0;google.getEI=function(a){for(var b;a&&(!a.getAttribute||!(b=a.getAttribute("eid")));)a=a.parentNode;return b||google.kEI};google.getLEI=function(a){for(var b=null;a&&(!a.getAttribute||!
(b=a.getAttribute("leid")));)a=a.parentNode;return b};google.ml=function(){return null};google.time=function(){return Date.now()};google.log=function(a,b,c,d,f){if(c=google.logUrl(a,b,c,d,f)){a=new Image;var
e=google.lc,g=google.li;e[g]=a;a.onerror=a.onload=a.onabort=function(){delete e[g]};a.src=c;google.li=g+1}};google.logUrl=function(a,b,c,d,f){var e="";c||-1!=b.search("&ei=")||(e="&ei="+google.getEI(d),-1==b.search("&lei=")&&
(d=google.getLEI(d))&&(e+="&lei="+d));d="";!c&&google.cshid&&-1==b.search("&cshid=")&&"slh"!=a&&(d="&cshid="+google.cshid);c=c||"/"+(f||"gen_204")+"?
atyp=i&ct="+a+"&cad="+b+e+"&zx="+google.time()+d;/^http:/i.test(c)&&"https:"==window.location.protocol&&(google.ml(Error("a"),!1,{src:c,glmm:1}),c="");return c};}).call(this);(function(){google.y={};google.x=function(a,b){if(a)var
c=a.id;else{do c=Math.random();while(google.y[c])}google.y[c]=[a,b];return!1};google.lm=[];google.plm=function(a){google.lm.push.apply(google.lm,a)};google.lq=[];google.load=function(a,b,c)
{google.lq.push([[a],b,c])};google.loadAll=function(a,b){google.lq.push([a,b])};google.bx=!1;google.lx=function(){};}).call(this);google.f={};(function(){
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");a="1"==c||"q"==c&&!a.elements.q.value?!0:!1}else a=!1;a&&
(b.preventDefault(),b.stopPropagation())},!0);document.documentElement.addEventListener("click",function(b){var a;a:{for(a=b.target;a&&a!=document.documentElement;a=a.parentElement)if("A"==a.tagName){a="1"==a.getAttribute("data-
nohref");break a}a=!1}a&&b.preventDefault()},!0);}).call(this);(function(){google.hs={h:true,peh:true,sie:false};})();(function(){google.c={btfi:false,datfo:false,gl:true,lhc:false,lmv:true,sdi:false};(function(){
var e=window.performance;google.timers={};google.startTick=function(a){google.timers[a]={t:{start:google.time()},e:{},m:{}}};google.tick=function(a,b,c){google.timers[a]||google.startTick(a);c=void 0!==c?c:google.time();b instanceof
Array||(b=[b]);for(var d=0,f;f=b[d++];)google.timers[a].t[f]=c};google.c.e=function(a,b,c){google.timers[a].e[b]=c};google.c.b=function(a){var b=google.timers.load.m;b[a]&&google.ml(Error("a"),!1,{m:a});b[a]=!0};google.c.u=function(a)
{var b=google.timers.load.m;if(b[a]){b[a]=!1;for(a in b)if(b[a])return;google.csiReport()}else google.ml(Error("b"),!1,{m:a})};function g(a,b,c,d){a.addEventListener?
a.addEventListener(b,c,d||!1):a.attachEvent&&a.attachEvent("on"+b,c)}function h(a,b,c,d){a.addEventListener?a.removeEventListener(b,c,d||!1):a.attachEvent&&a.detachEvent("on"+b,c)}google.rll=function(a,b,c){var d=function(f)
{c(f);h(a,"load",d);h(a,"error",d)};g(a,"load",d);b&&g(a,"error",d)};google.aft=function(a){a.setAttribute("data-iml",google.time())};google.startTick("load");var k=google.timers.load;a:{var l=k.t;if(e){var m=e.timing;if(m){var
n=m.navigationStart,p=m.responseStart;if(p>n&&p<=l.start){l.start=p;k.wsrt=p-n;break a}}e.now&&(k.wsrt=Math.floor(e.now()))}}google.c.b("pr");google.c.b("xe");if(google.c.gl){var q=function(a)
{a&&google.aft(a.target)};g(document.documentElement,"load",q,!0);google.c.glu=function(){h(document.documentElement,"load",q,!0)}};}).call(this);})();(function(){
function h(){return window.performance&&window.performance.navigation&&window.performance.navigation.type};function m(a,b){return!a||!b&&p(a)?0:a.getBoundingClientRect?q(a,b,function(c){return c.getBoundingClientRect()}):1}function
p(a){return"none"==a.style.display?!0:document.defaultView&&document.defaultView.getComputedStyle?(a=document.defaultView.getComputedStyle(a),!!a&&("hidden"==a.visibility||"0px"==a.height&&"0px"==a.width)):!1}
function q(a,b,c){var d=c(a);a=d.left+window.pageXOffset;c=d.top+window.pageYOffset;var e=d.width;d=d.height;var f=0;if(!b&&0>=d&&0>=e)return f;0>c+d?f=2:c>=(window.innerHeight||document.documentElement.clientHeight)&&
(f=4);if(0>a+e||a>=(window.innerWidth||document.documentElement.clientWidth))f|=8;return f||1};var r=window.location,t="aft frt hct prt pprt sct".split(" ");function u(a){return(a=r.search.match(new RegExp("[?&]"+a+"=(\\d+)")))?
Number(a[1]):-1};var v=window.innerHeight||document.documentElement.clientHeight,w=0,x=0,y=0,z=0,A=0,B=0,C=0,D=!0,E=!0,F=-1;function G(a,b,c,d){var e=google.timers.load.t[a];e&&(c||d&&b&&b<e)||google.tick("load",a,b)}function H(a,b,c)
{a&&(G("frt",c,!1,!0),++z,I());b&&(G("aft",c,!1,!0),++B,I());G("iml",c,!1,!0);++x;J()}function J(){E||x!=w||google.c.u("il")}
function I(){if(!D&&B==A&&z==y){google.c.e("load","ima",String(A));google.c.e("load","imad",String(C));google.c.e("load","aftp",String(Math.round(F)));var a=google.timers.load,b=a.m;if(!b||!b.prs){var c=h()?0:u("qsubts");0<c&&
(b=u("fbts"),0<b&&(a.t.start=Math.max(c,b)));var d=a.t,e=d.start;b={wsrt:a.wsrt};for(var f=0,g;g=t[f++];){var n=d[g];n&&e&&(b[g]=n-e)}0<c&&(b.gsasrt=a.t.start-c);c=a.e;a="/gen_204?
s="+google.sn+"&t=aft&atyp=csi&ei="+google.kEI+"&rt=";d="";for(k in b)a+=""+d+k+"."+b[k],d=",";for(var l in c)a+="&"+l+"="+c[l];google.cshid&&(a+="&cshid="+google.cshid);2==h()&&(a+="&bb=1");1==h()&&(a+="&r=1");if("gsasrt"in b){var
k=u("qsd");0<k&&(a+="&qsd="+k)}google.kBL&&(a+="&bl="+google.kBL);k=a;navigator.sendBeacon?navigator.sendBeacon(k,""):google.log("","",k)}}}var K="src bsrc url ll image img-url".split(" ");function L(a){for(var
b=0,c;c=K[b++];)if(a.getAttribute("data-"+c))return!0;return!1}google.c.b("il");google.c.setup=function(a,b){var c=a.getAttribute("data-atf");if(c)return Number(c);var d="string"!=typeof a.src||!a.src,e=!!a.getAttribute("data-
bsrc");c=a.getAttribute("data-deferred");!c&&L(a)&&a.setAttribute("data-lzy_",1);d=(d||a.complete)&&!c&&!e;var f=!google.c.lhc&&Number(a.getAttribute("data-iml"))||0;e=m(a,e&&google.c.lmv);a.setAttribute("data-
atf",e);++w;if(d&&!f||a.hasAttribute("data-noaft"))++x;else{var g=1==e,n=g||google.c.btfi&&e&4&&f&&F<v;if(n){if(a.getBoundingClientRect){var l=a.getBoundingClientRect();l=g?
Math.min(v,l.bottom+window.pageYOffset):l.top+window.pageYOffset;if(!google.c.btfi||!d||!f||l>=v)F=Math.max(F,l)}g&&(++A,c&&++C)}b&&++y;n&&f&&G("aft",f,!1,!0);d&&f?H(b,g,f):google.rll(a,!0,function(){H(b,g,google.time())})}return
e};google.c.ubr=function(a,b){google.c.btfi?0>F&&G("aft",b):G("aft",b,!0);D&&(D=!1,I());a&&E&&(G("prt",b),G("iml",b,!0),E=!1,J(),google.c.setup=function(){return 0},google.c.ubr=function(){})};}).call(this);(function(){
var b=[function(){google.tick&&google.tick("load","dcl")}];google.dclc=function(a){b.length?b.push(a):a()};function c(){for(var a;a=b.shift();)a()}window.addEventListener?
(document.addEventListener("DOMContentLoaded",c,!1),window.addEventListener("load",c,!1)):window.attachEvent&&window.attachEvent("onload",c);}).call(this);(function(){
var b=[];google.jsc={xx:b,x:function(a){b.push(a)},mm:[],m:function(a){google.jsc.mm.length||(google.jsc.mm=a)}};}).call(this);(function(){
var f=this||self;
```

Questions

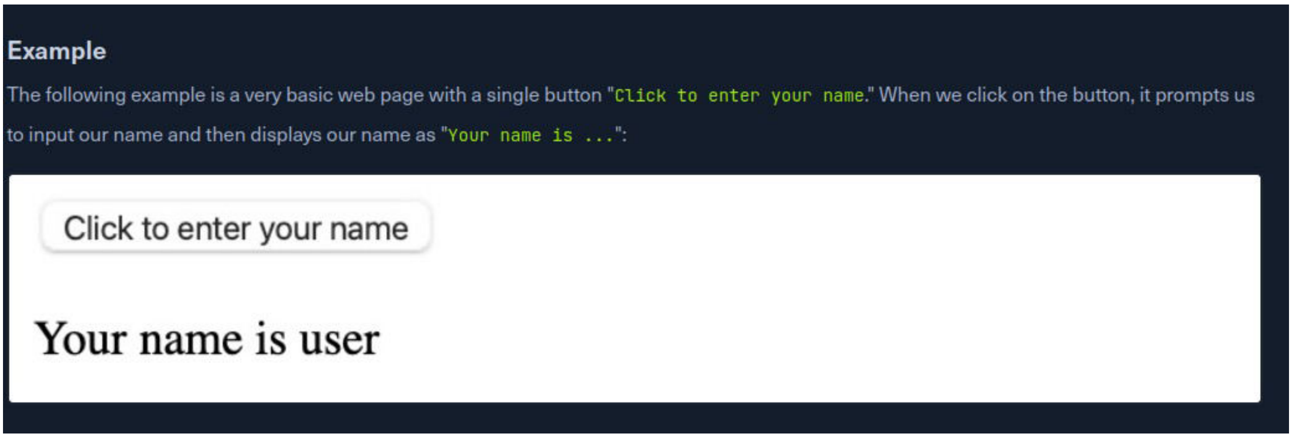
Check the above login form for exposed passwords. Submit the password as the answer.

ANS: **HiddenInPlainSight**



```
    .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>

</html>
```

HTML Injection



Example

The following example is a very basic web page with a single button "`Click to enter your name`." When we click on the button, it prompts us to input our name and then displays our name as "`Your name is ...`":

Click to enter your name

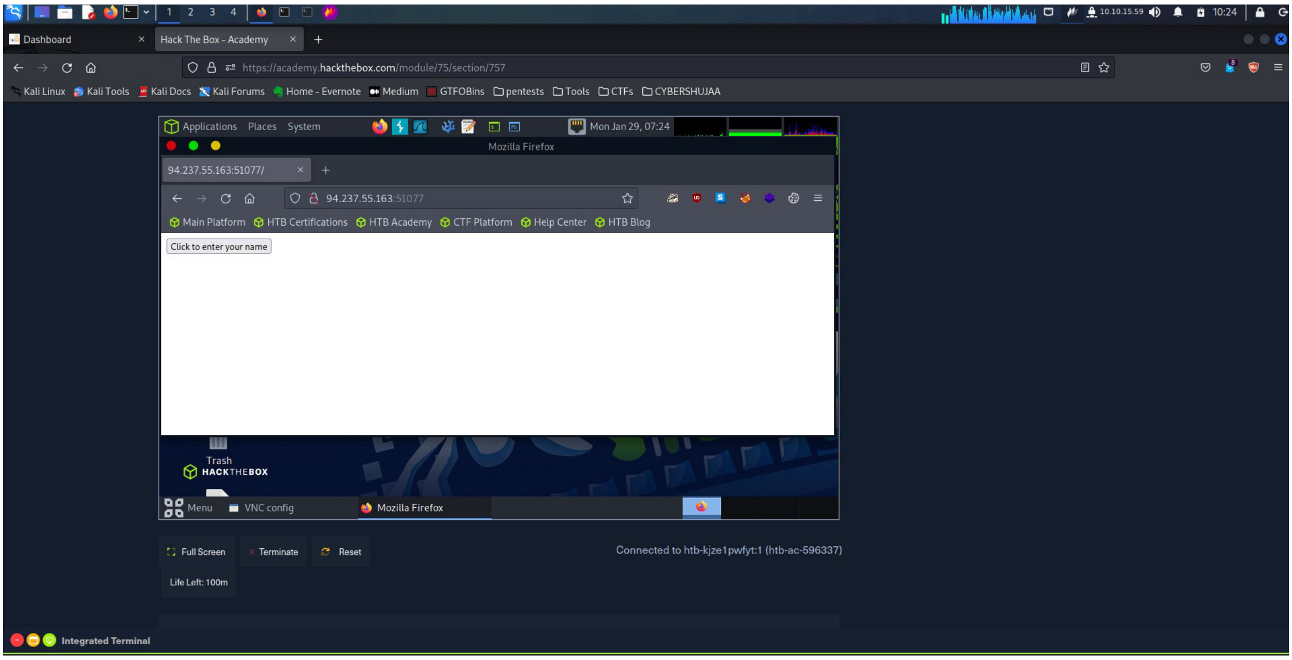
Your name is user

HTML Injection occurs when unfiltered user input is displayed on the page. I get also to understand, an Injection 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.

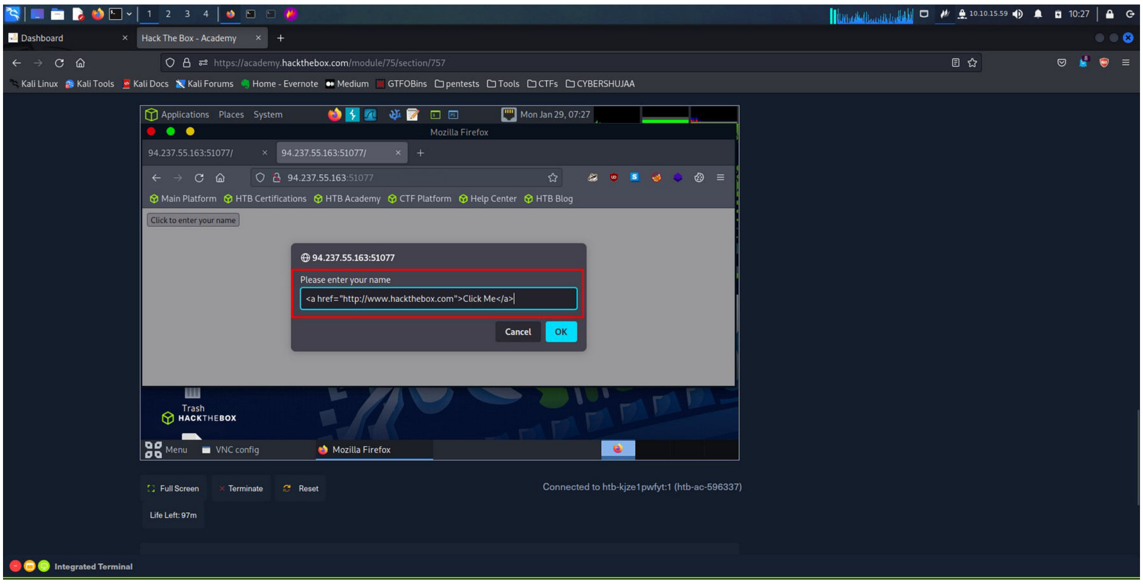
Question.

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> ANS: **Your name is Click Me**

My target IP adress is: 94.237.55.163:51077



After Clicking the button Click to enter your name, I keyed in the HTML injection code “<a href="http://www.hackthebox.com">Click Me</a>” which revealed the answer for the above exercise.



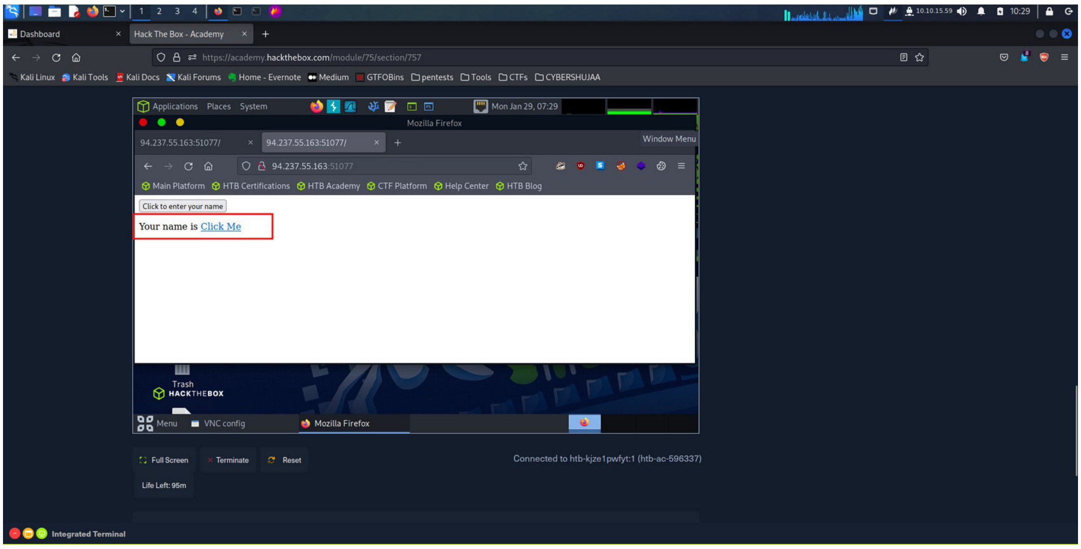
94.237.55.163:51077

Please enter your name

```
<a href="http://www.hackthebox.com">Click Me</a>
```

Connected to htb-kjze1pwfyt:1 (htb-ac-596337)

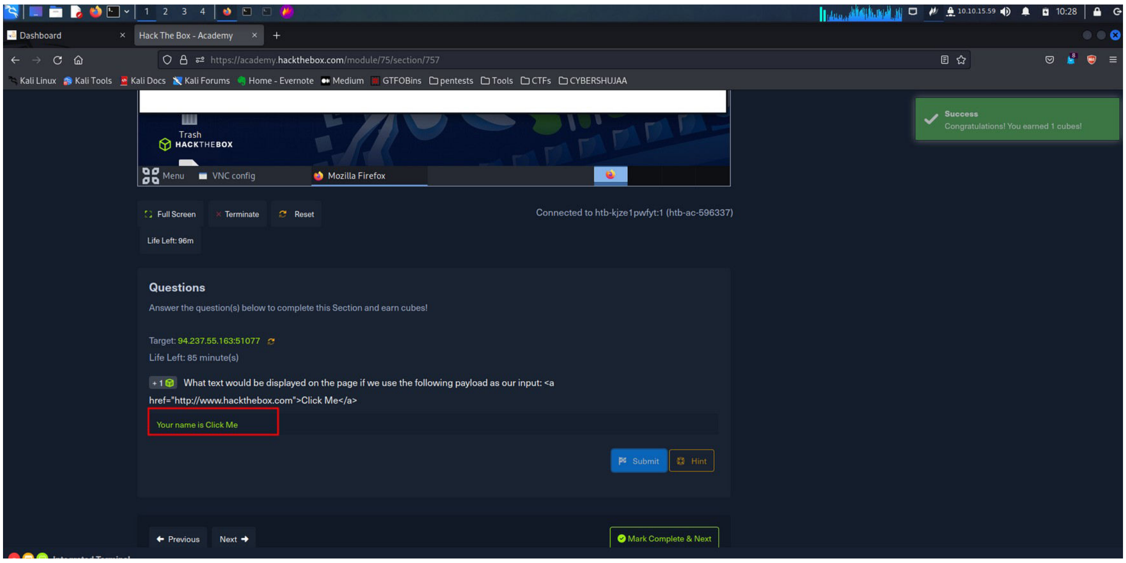
Life Left: 97m



Your name is Click Me

Connected to htb-kjze1pwfyt:1 (htb-ac-596337)

Life Left: 95m



Success
Congratulations! You earned 1 cubes!

Connected to htb-kjze1pwfyt:1 (htb-ac-596337)

Life Left: 96m

Questions

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

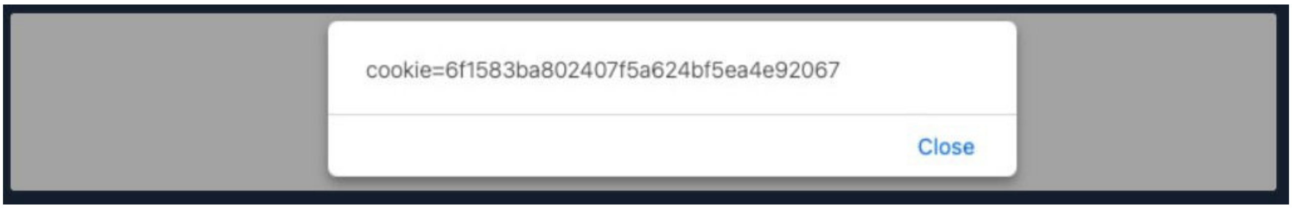
Target: 94.237.55.163:51077

Life Left: 85 minute(s)

+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

Cross-Site Scripting (XSS)



cookie=6f1583ba802407f5a624bf5ea4e92067

In this section I got to understand that HTML Injection vulnerabilities can be utilized to 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.

I also learnt that XSS is similar to HTML Injection in practice only that XSS involves the injection of JavaScript code to perform more advanced attacks on the client-side, instead of merely injecting HTML code.

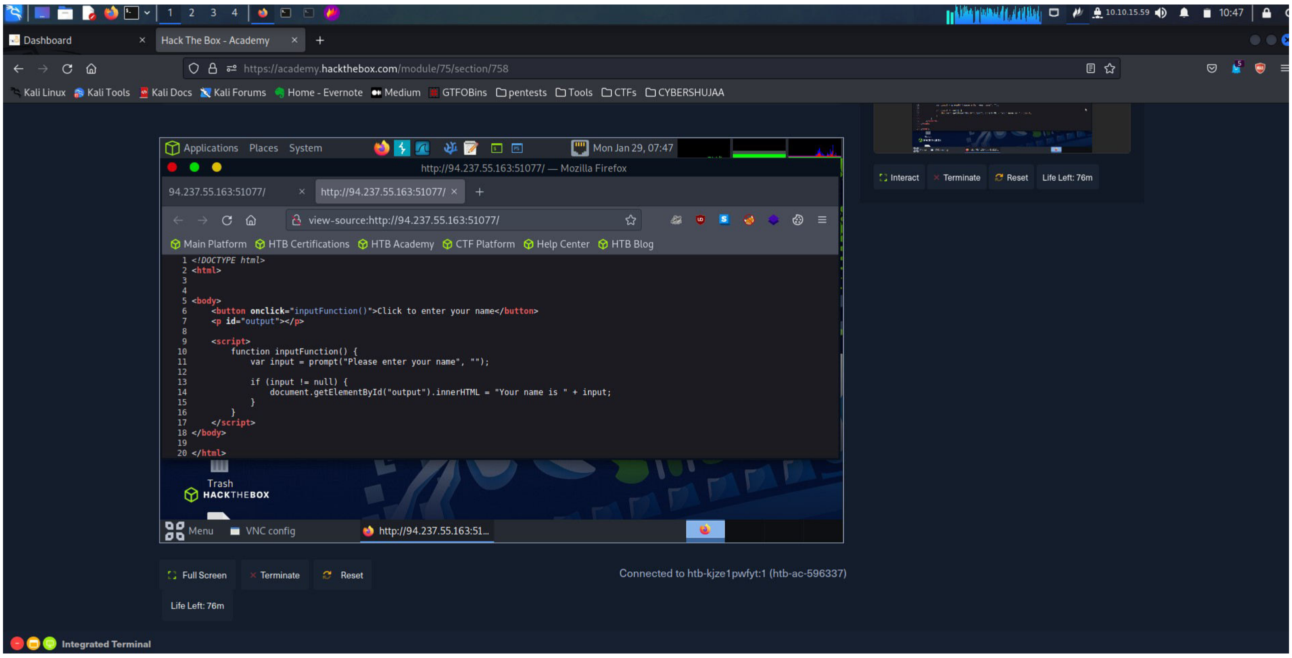
There are three main types of XSS: Reflecte XSS, Stored XSS and Dom XSS.

Using this " <img src=/ onerror=alert(document.cookie)> " javascript code, we can get an alert window pops up with the cookie value in it:

Question.

Try to use XSS to get the cookie value in the above page ANS: **XSSisFun**

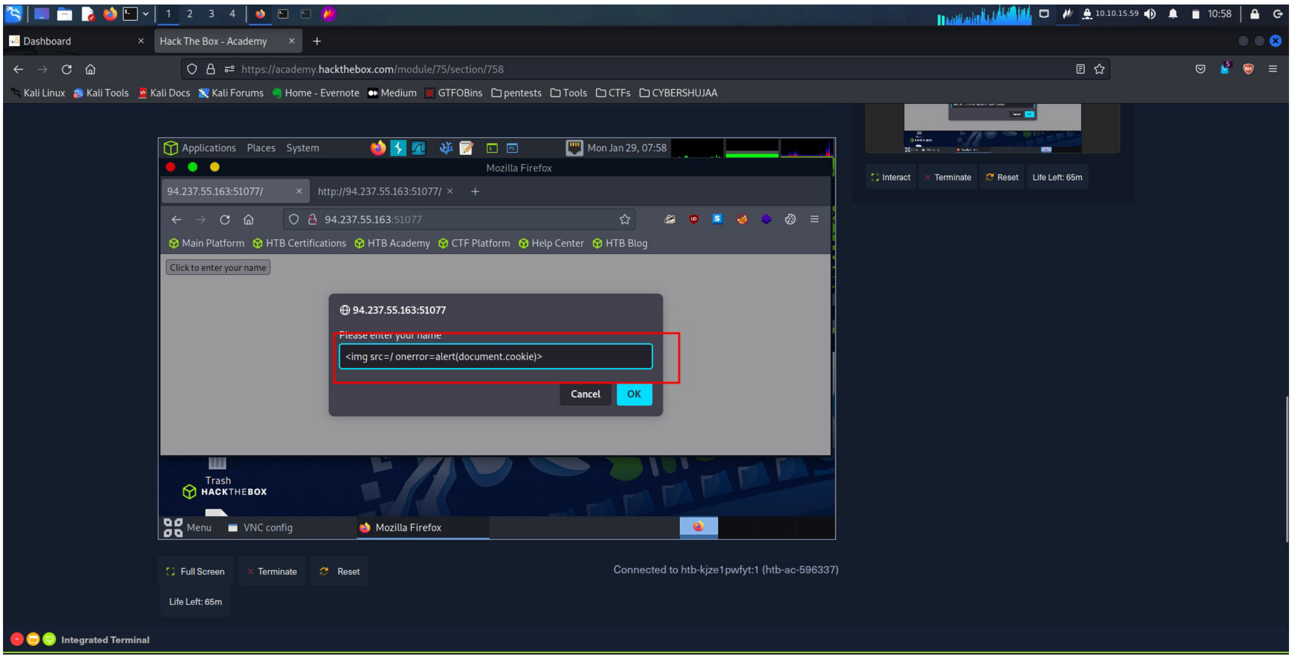
Here is the source code for the targeted machine.



```html
<!DOCTYPE html>
<html>


<body>
    <button onclick="inputFunction()">Click to enter your name</button>
    <p id="output"></p>

    <script>
        function inputFunction() {
            var input = prompt("Please enter your name", "");

            if (input != null) {
                document.getElementById("output").innerHTML = "Your name is " + input;
            }
        }
    </script>
</body>

</html>
```

I will then inject the XSS to get the cookie value in the above page.

XSS Code = <img src=/ onerror=alert(document.cookie)>

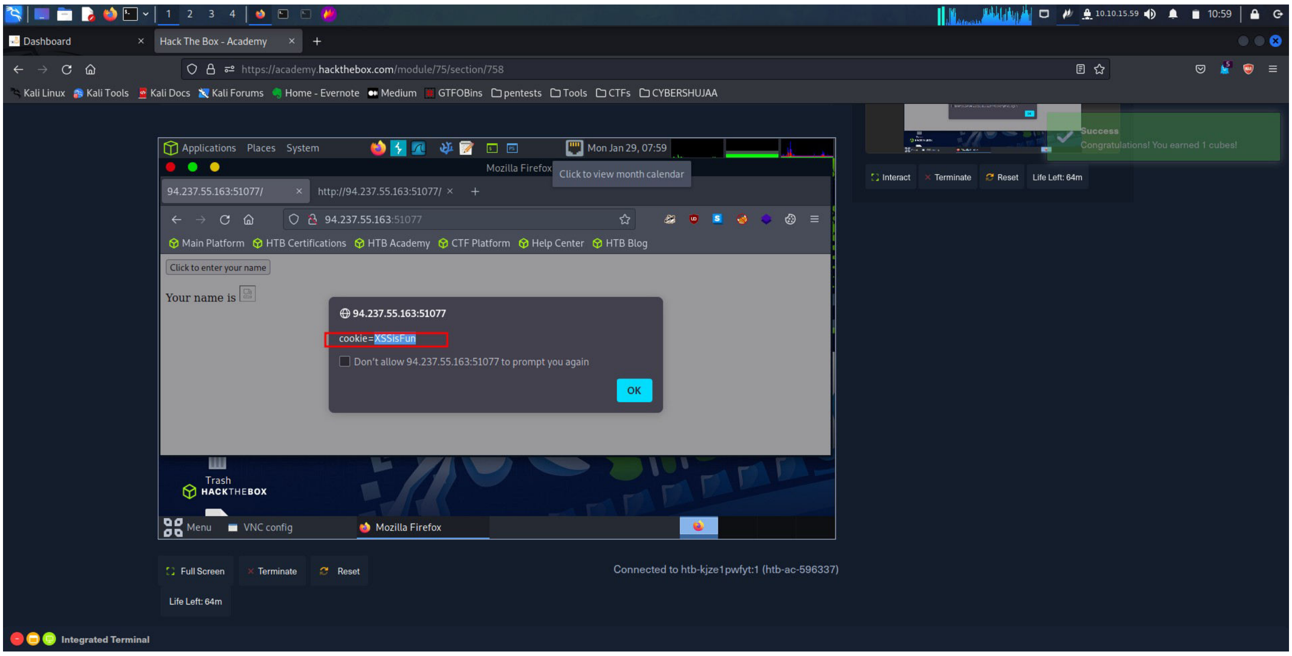


94.237.55.163:51077

Please enter your name

`<img src=/ onerror=alert(document.cookie)>`

Connected to htb-kjze1pwfyt:1 (htb-ac-596337)



94.237.55.163:51077

cookie=XSSisFun

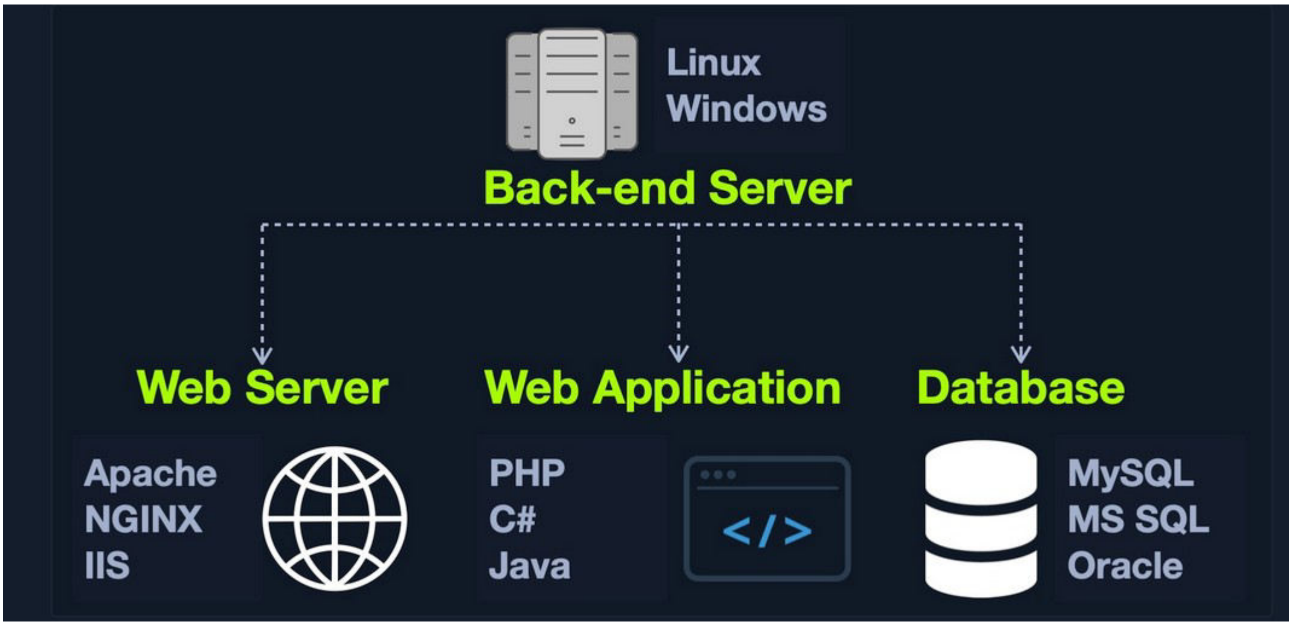
□ Don't allow 94.237.55.163:51077 to prompt you again

Connected to htb-kjze1pwfyt:1 (htb-ac-596337)

Cross-Site Request Forgery (CSRF)

This is a third party front end vulnerability caused by unfiltered user input. CSRF attacks may utilize XSS vulnerabilities to perform certain queries , and API calls on a web application that the victim is currently authenticated to. I understood that CSRF can also be leveraged to attack admins and gain access to their accounts. Admins usually have access to sensitive functions, which can sometimes be used to attack and gain control over the back-end server although this depends on the functionality provided to admins within a given web application.

Back End Servers



This is the hardware and operating system on the back end that hosts all the applications necessary to run the web application. This is the real system that carries out all processes and tasks making up the entire web application.

Back end has 3 back components:-

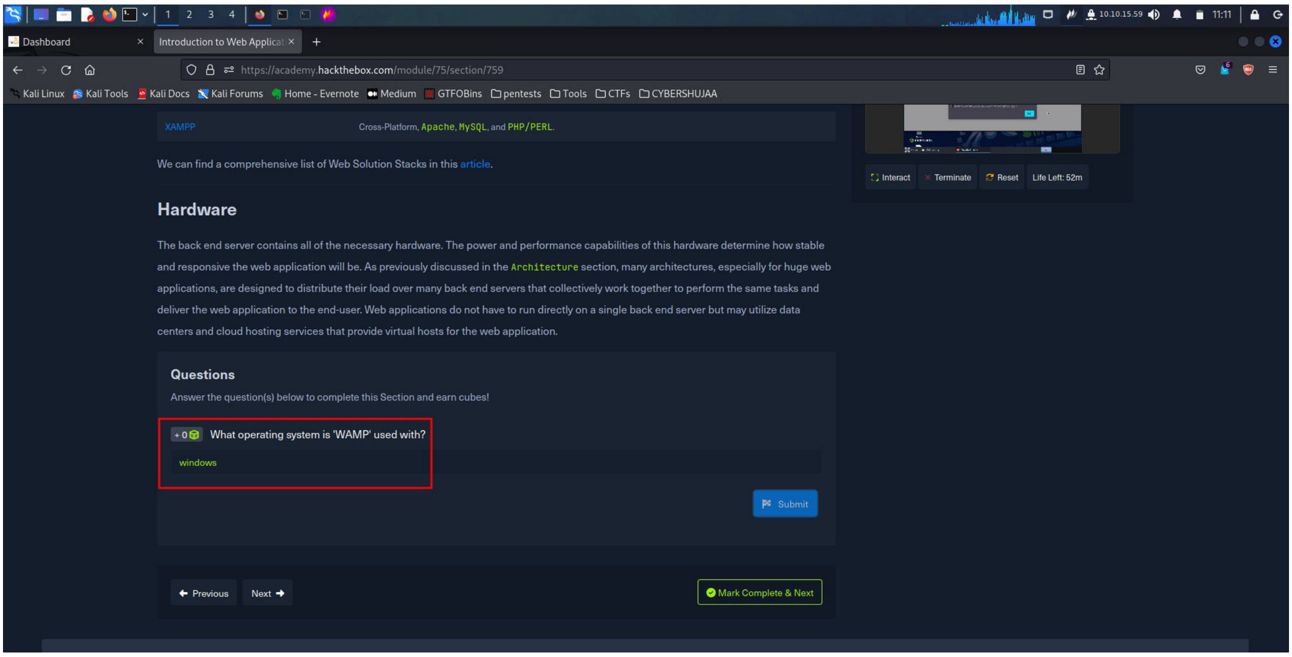
1. Web Server

2. Database

3. Development Framework

Question

What operating system is 'WAMP' used with? ANS: **windows**



XAMPP	Cross-Platform, Apache, MySQL, and PHP/PERL.

We can find a comprehensive list of Web Solution Stacks in this article.

Hardware

The back end server contains all of the necessary hardware. The power and performance capabilities of this hardware determine how stable and responsive the web application will be. As previously discussed in the `Architecture` section, many architectures, especially for huge web applications, are designed to distribute their load over many back end servers that collectively work together to perform the same tasks and deliver the web application to the end-user. Web applications do not have to run directly on a single back end server but may utilize data centers and cloud hosting services that provide virtual hosts for the web application.

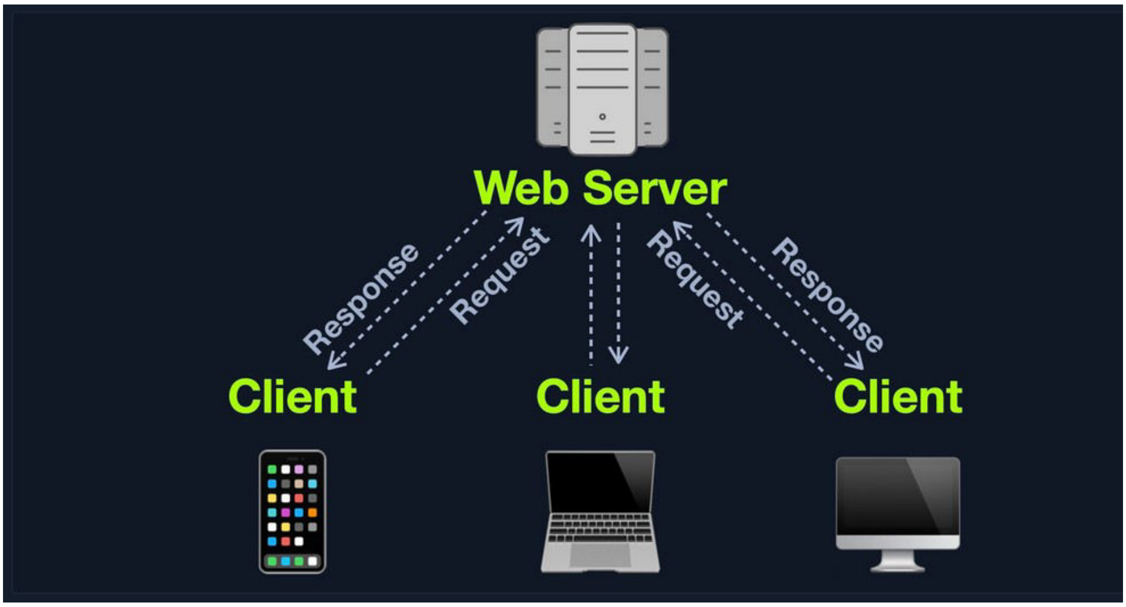
Questions

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

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

windows

Web Servers



I was able to understand on how typical web server will accept HTTP requests from the client-side, and responds with different HTTP responses and codes. Example:

- 200 OK – Responds to a successful request.

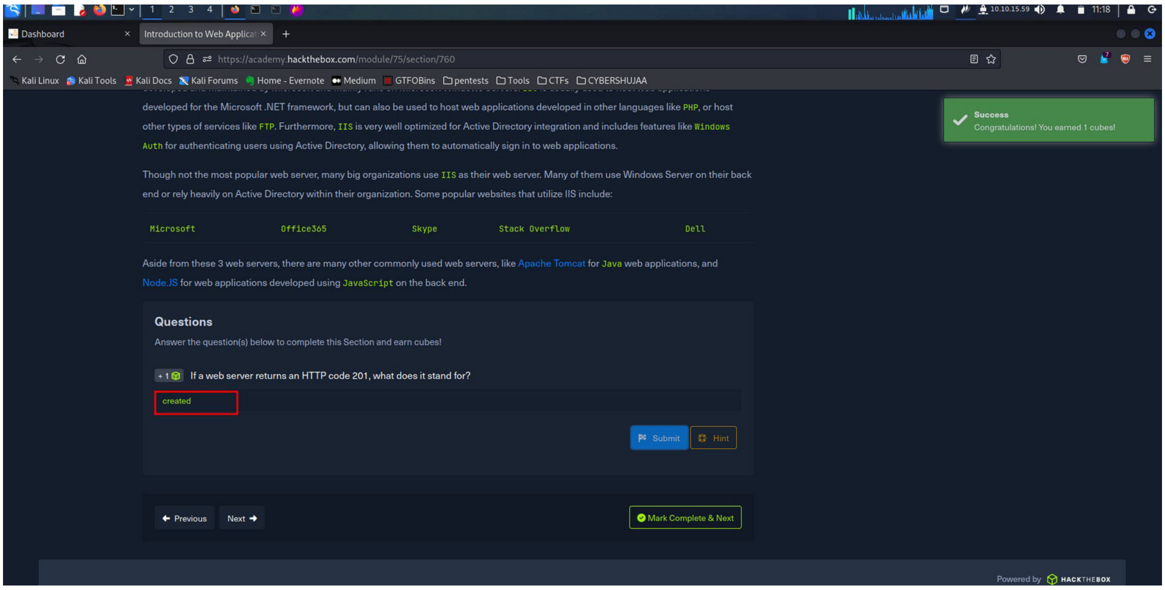
- 404 NOT FOUND – Responds to pages that do not exist.

- 403 FORBIDDEN – Responds to sites that are restricted to access.

Question

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

ANS: **Created.**



developed for the Microsoft .NET framework, but can also be used to host web applications developed in other languages like PHP, or host other types of services like FTP. Furthermore, IIS is very well optimized for Active Directory integration and includes features like Windows Auth for authenticating users using Active Directory, allowing them to automatically sign in to web applications.

Though not the most popular web server, many big organizations use IIS as their web server. Many of them use Windows Server on their back end or rely heavily on Active Directory within their organization. Some popular websites that utilize IIS include:

```
Microsoft        Office365        Skype        Stack Overflow        Dell
```

Aside from these 3 web servers, there are many other commonly used web servers, like Apache Tomcat for Java web applications, and Node.JS for web applications developed using JavaScript on the back end.

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

Success
Congratulations! You earned 1 cubes!

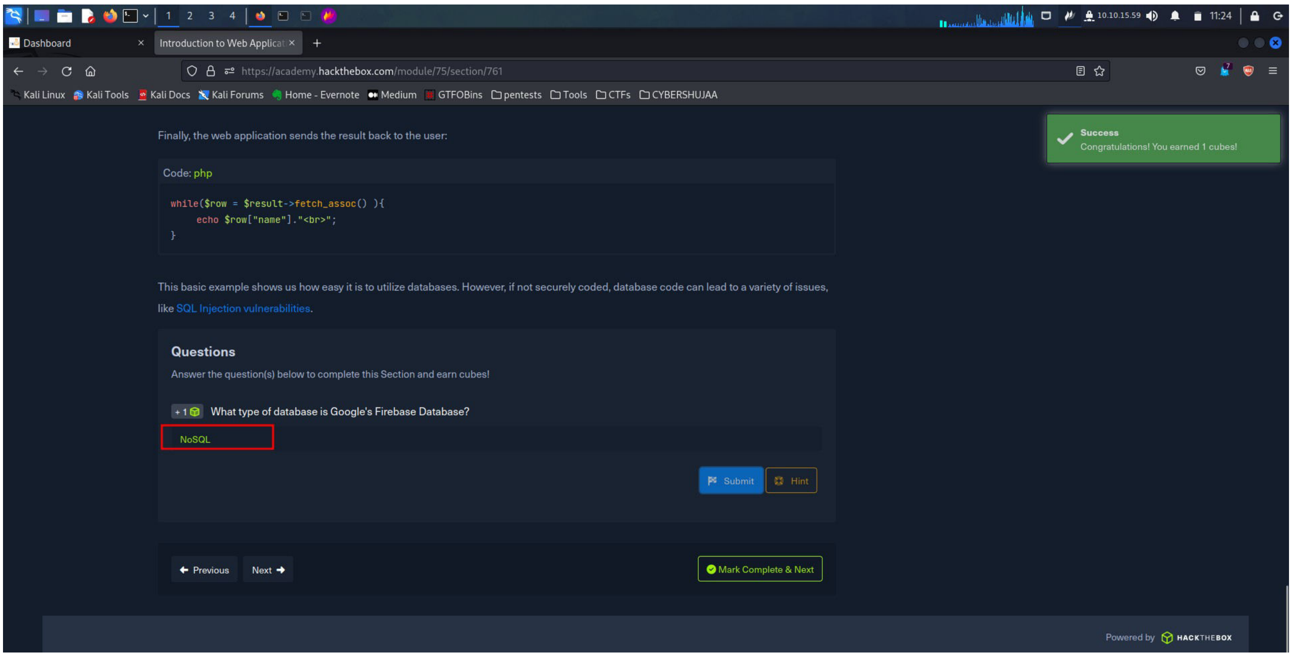
Databases.

Web applications will utilize back end databases to store various content and information related to the web application.

This content include:- Passwords, Usernames, Addresses, Contacts, Images and files etc.

Question

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



Success
Congratulations! You earned 1 cubes!

Finally, the web application sends the result back to the user:

Code: php

```
while($row = $result->fetch_assoc() ){
     echo $row["name"]."<br>";
}
```

This basic example shows us how easy it is to utilize databases. However, if not securely coded, database code can lead to a variety of issues, like SQL Injection vulnerabilities.

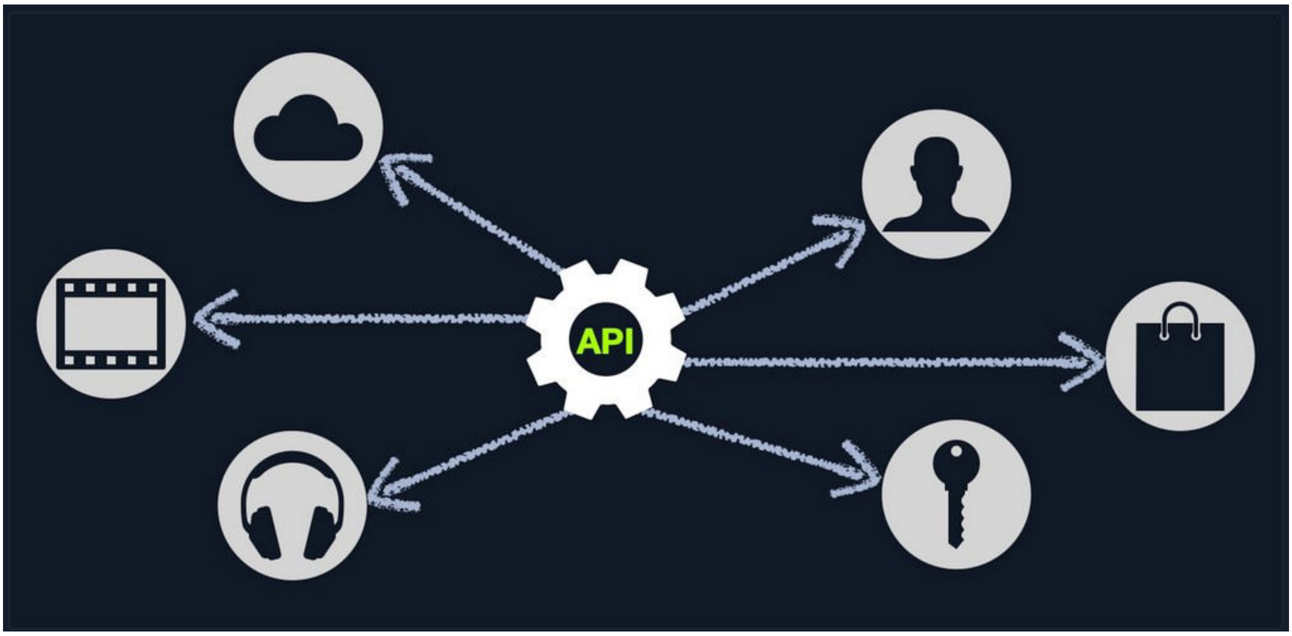
Questions

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

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

NoSQL

Development Frameworks & APIs



From this section I learnt that instead of developing a web application from scratch, we can utilize web frameworks.

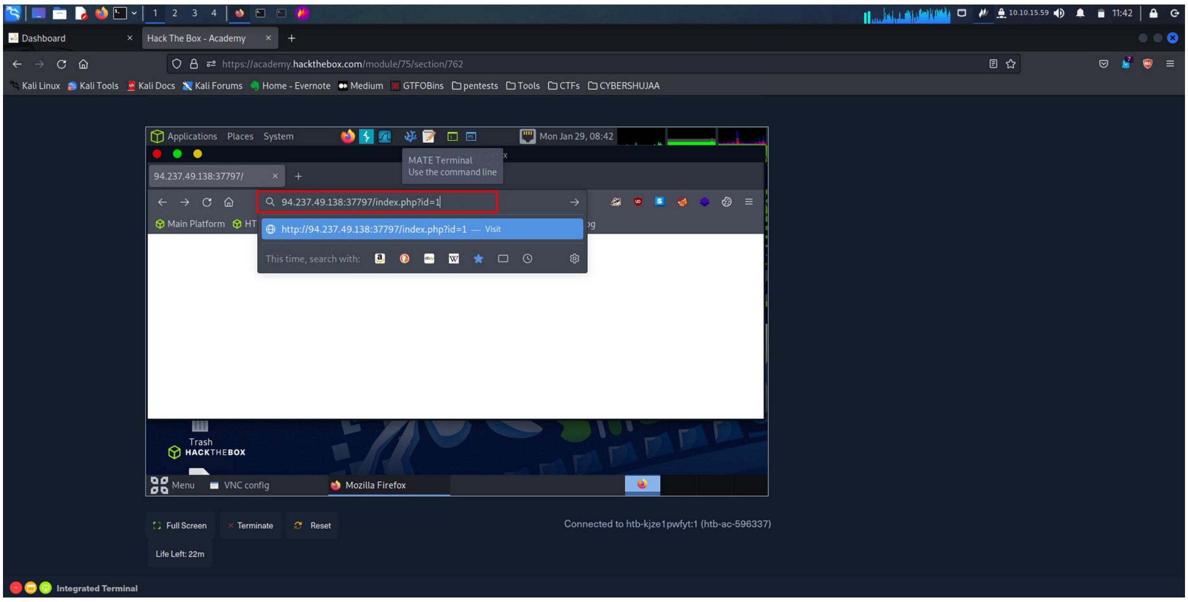
I have also learnt that from the fact that most web applications share common functionalities, for example user registration, web development frameworks makes it easy to quickly implement functionality and link them to the front end components and therefore by this making a fully functional web application.

Question

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

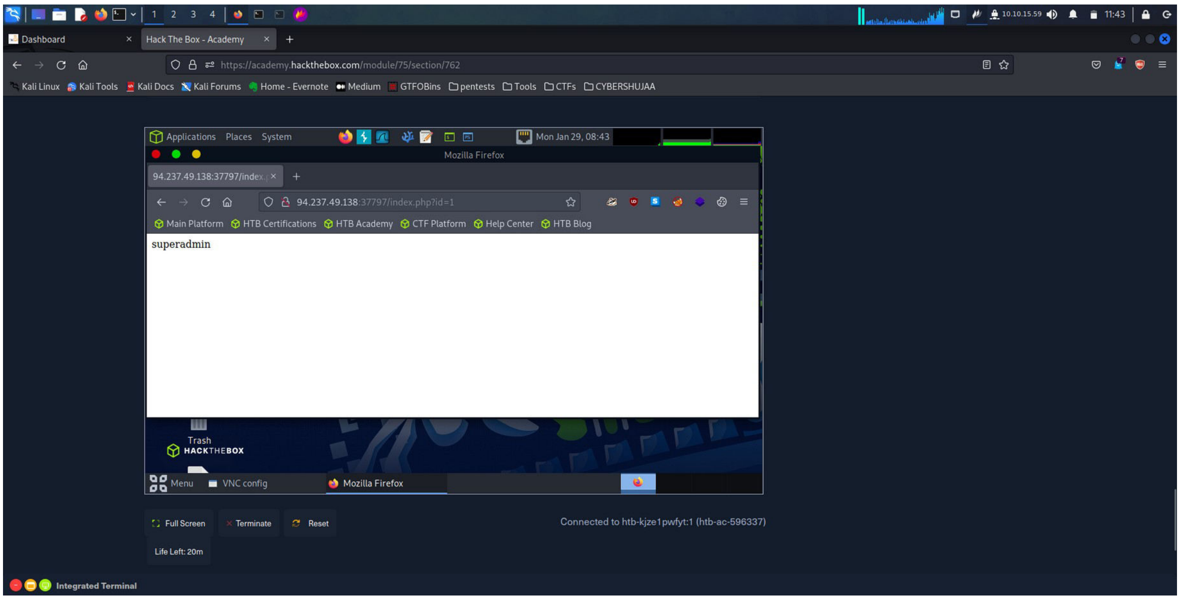
My target IP address is: 94.237.49.138:37797

Therefore the GET request will be: 94.237.49.138:37797/index.php?id=1



```
94.237.49.138:37797/index.php?id=1
http://94.237.49.138:37797/index.php?id=1 — Visit
```

Output



```
94.237.49.138:37797/index.php?id=1
superadmin
```

Common Web Vulnerabilities

Examples of common web vulnerabilities include:-

- Broken Authentication/Access Control.

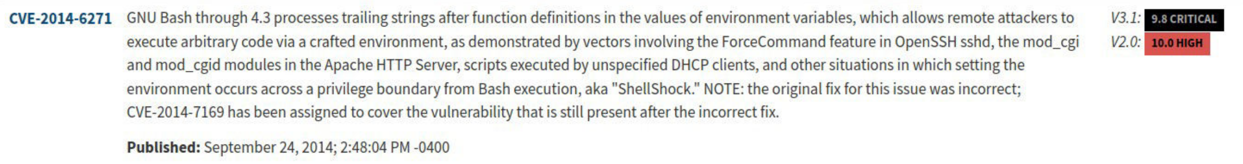
- Malicious File Upload.

- SQLi (SQL Injection).

Question

To which of the above categories does public vulnerability 'CVE-2014-6271' belongs to? ANS: **Command Injection**

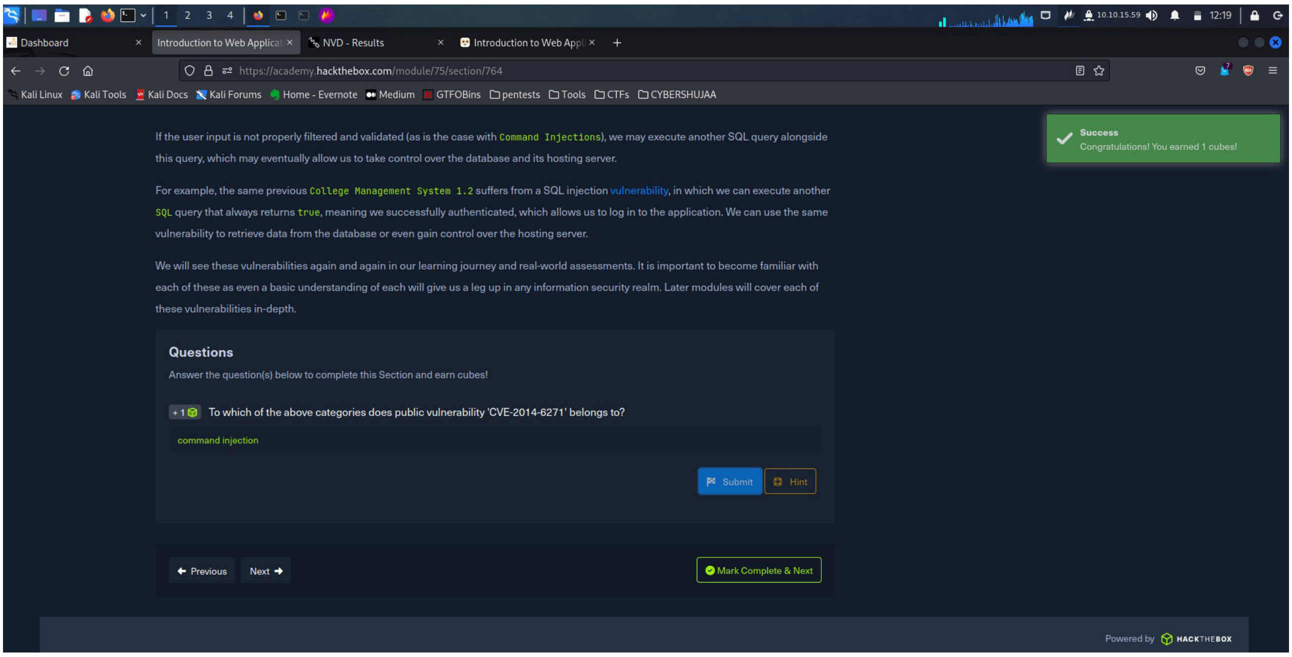
In this question I had to go a few blogs and posts online to the point at wich I landed on a website called, National Vulnerability Database. Here is an extended information about the CVE.



CVE-2014-6271 GNU Bash through 4.3 processes trailing strings after function definitions in the values of environment variables, which allows remote attackers to execute arbitrary code via a crafted environment, as demonstrated by vectors involving the ForceCommand feature in OpenSSH sshd, the mod_cgi and mod_cgid modules in the Apache HTTP Server, scripts executed by unspecified DHCP clients, and other situations in which setting the environment occurs across a privilege boundary from Bash execution, aka "ShellShock." NOTE: the original fix for this issue was incorrect; CVE-2014-7169 has been assigned to cover the vulnerability that is still present after the incorrect fix.

V3.1: 9.8 CRITICAL
V2.0: 10.0 HIGH

Published: September 24, 2014; 2:48:04 PM -0400



If the user input is not properly filtered and validated (as is the case with `Command Injections`), we may execute another SQL query alongside this query, which may eventually allow us to take control over the database and its hosting server.

For example, the same previous `College Management System 1.2` suffers from a SQL injection vulnerability, in which we can execute another `SQL` query that always returns `true`, meaning we successfully authenticated, which allows us to log in to the application. We can use the same vulnerability to retrieve data from the database or even gain control over the hosting server.

We will see these vulnerabilities again and again in our learning journey and real-world assessments. It is important to become familiar with each of these as even a basic understanding of each will give us a leg up in any information security realm. Later modules will cover each of these vulnerabilities in-depth.

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

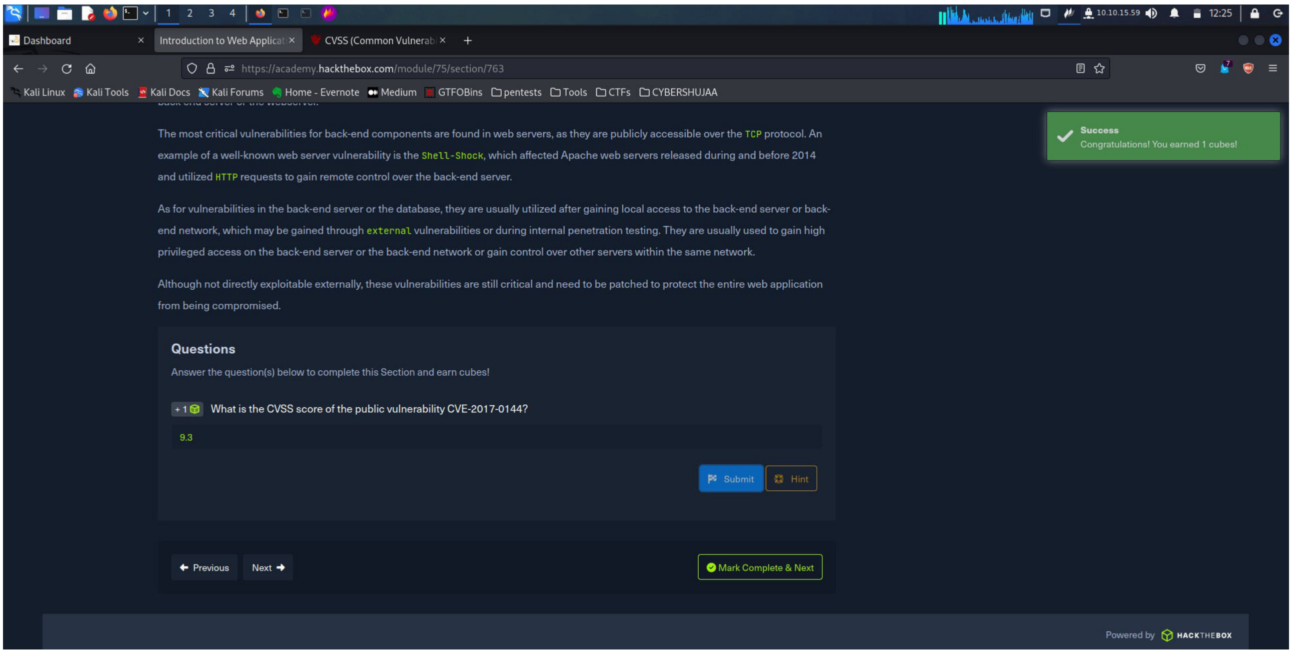
Success
Congratulations! You earned 1 cubes!

Public Vulnerabilities

In this section I learn of new tools to check the CVE scores from NVD that is CVSS v2 calculator and a CVSS v3 calculator that organizations can use to factor additional risk from Temporal and Environmental data unique to them.

Question

What is the CVSS score of the public vulnerability CVE-2017-0144? ANS: **9.3**



The most critical vulnerabilities for back-end components are found in web servers, as they are publicly accessible over the TCP protocol. An example of a well-known web server vulnerability is the Shell-Shock, which affected Apache web servers released during and before 2014 and utilized HTTP requests to gain remote control over the back-end server.

As for vulnerabilities in the back-end server or the database, they are usually utilized after gaining local access to the back-end server or back-end network, which may be gained through external vulnerabilities or during internal penetration testing. They are usually used to gain high privileged access on the back-end server or the back-end network or gain control over other servers within the same network.

Although not directly exploitable externally, these vulnerabilities are still critical and need to be patched to protect the entire web application from being compromised.

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

Success
Congratulations! You earned 1 cubes!

Conclusion.

In the completion of the "Introduction to Web Applications" module at htb-academy has enabled me to gain foundational knowledge and skills in web development. Throughout this module, I have explored key concepts such as front end (HTML, CSS, JavaScript) and back end. I have also gained an understanding of the basics of web architecture, client-server communication and security considerations.

With this knowledge, I am better equipped with the basics to design, develop and troubleshoot web applications.

Thank you.