

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



Tryhackme:How websites work



jagadeesh · [Follow](#)

7 min read · Mar 31, 2021



Listen

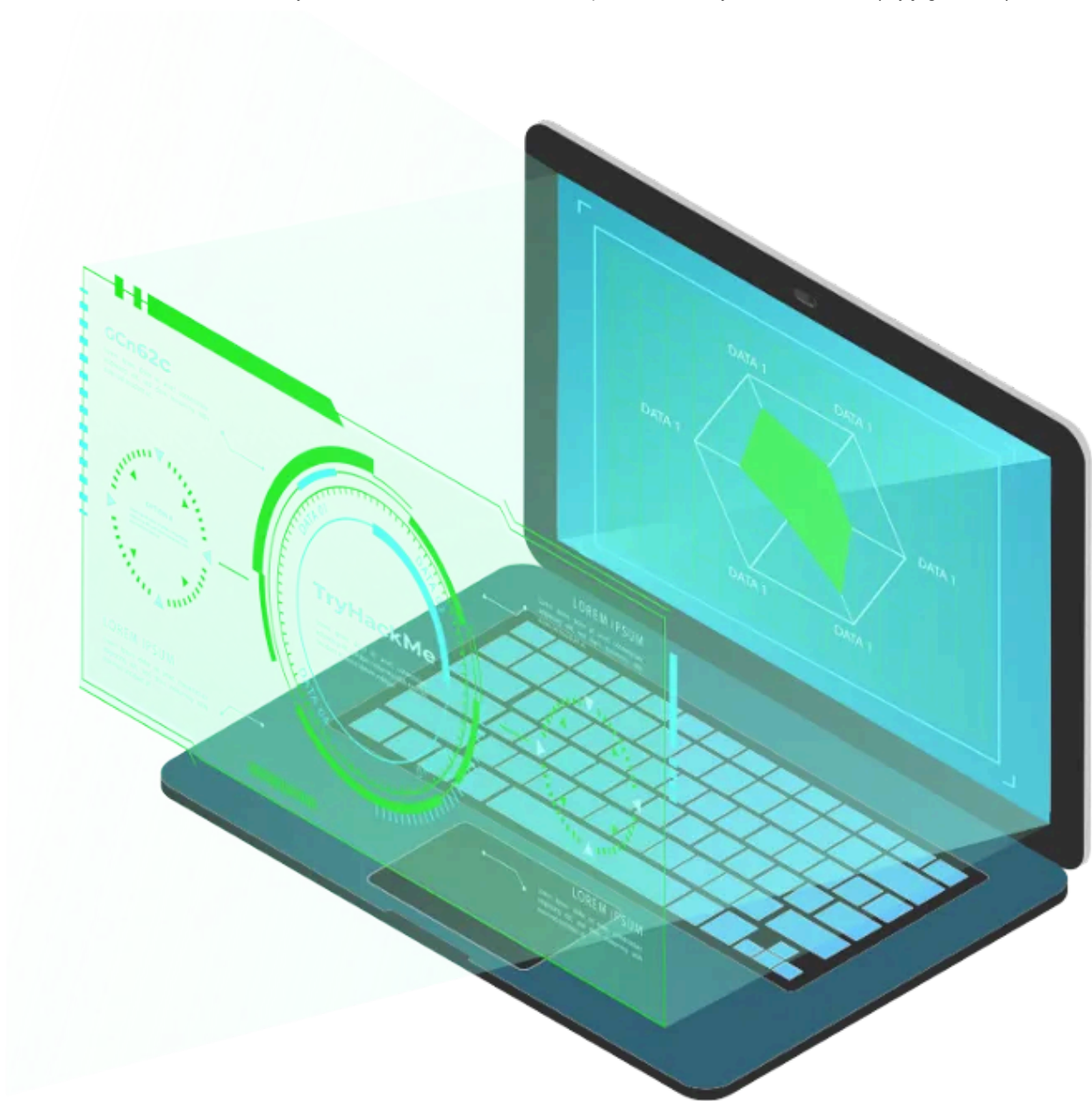


Share



More

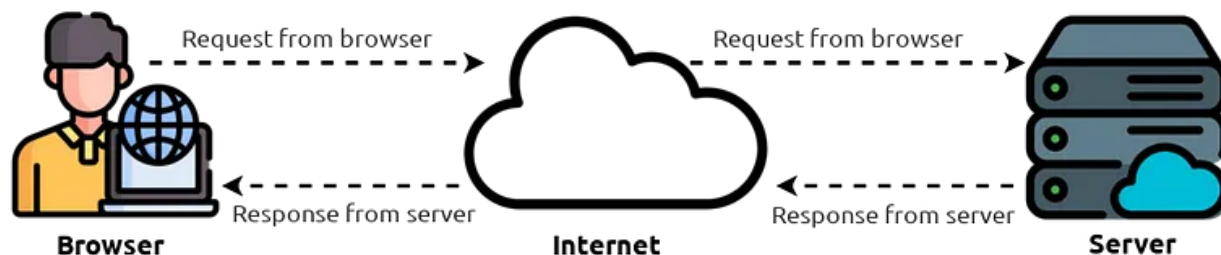
To exploit a website, you first need to know how they are created.



How websites work:

By the end of this room, you'll know how websites are created and will be introduced to some basic security issues.

When you visit a website your browser (*like Safari or Google Chrome*) makes a request to a web server asking for information about the page you're visiting and will respond with data that your browser uses to show you the page; a web server is just a dedicated computer somewhere else in the world that handles your requests.



There are two major components that make up a website:

1. Front End (Client-Side) — the way your browser renders a website.
2. Back End (Server-Side) — a server that processes your request and returns a response.

There are many other processes involved in your browser making a request to a web server, but for now you just need to understand that you make a request to a server and it responds with data your browser uses to render information to you.

1. What term describes the way your browser renders a website?

A:front end

HTML:

Websites are primarily created using:

- HTML, to build websites and define their structure
- CSS, to make websites look pretty by adding styling options
- JavaScript, implement complex features on pages using interactivity

HyperText Markup Language (HTML) is the language websites are written in. Elements (also known as tags) are the building blocks of HTML pages and tells the browser how to display content. The code snippet below shows a simple HTML document, the structure of which is the same for every website:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Example Heading</h1>
    <p>Example paragraph..</p>
  </body>
</html>
```

The HTML structure (as shown in the screenshot) has the following components:

- The `<!DOCTYPE html>` defines that the page is a HTML5 document. This helps with standardisation across different browsers and tells the browser to use HTML5 to interpret the page.
- The `<html>` element is the root element of the HTML page - all other elements come after this element.
- The `<head>` element contains information about the page (such as the page title)
- The `<body>` element defines the HTML document's body, only content inside of the body is shown in the browser.
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph
- There are many other elements (tags) used for different purposes. For example, there are tags for: buttons (`<button>`), images (``), lists, and much more.

Tags can contain attributes such as the class attribute which can be used to style an element (e.g. make the tag a different color) `<p class="bold-text">` , or the `src` attribute which is used on images to specify the location of an image: ``. An element can have multiple attributes each with its own unique purpose e.g. `<p attribute1="value1" attribute2="value2">`

Elements can also have an id attribute (`<p id="example">`), which is unique to the element. Unlike the class attribute where multiple elements can use the same class, an element must have different id's to uniquely identify them. Element id's are used for styling and to identify it by JavaScript.

You can view the HTML of any website by right clicking, and selecting “View Page Source” (Chrome) / “Show Page Source” (Safari).

1.Let’s play with some HTML! On the right-hand side, you should see a box that renders HTML — If you enter some HTML into the box, and click the green “Render HTML Code” button it will render your HTML on the page; you should see an image of some cats.

A:no answer need

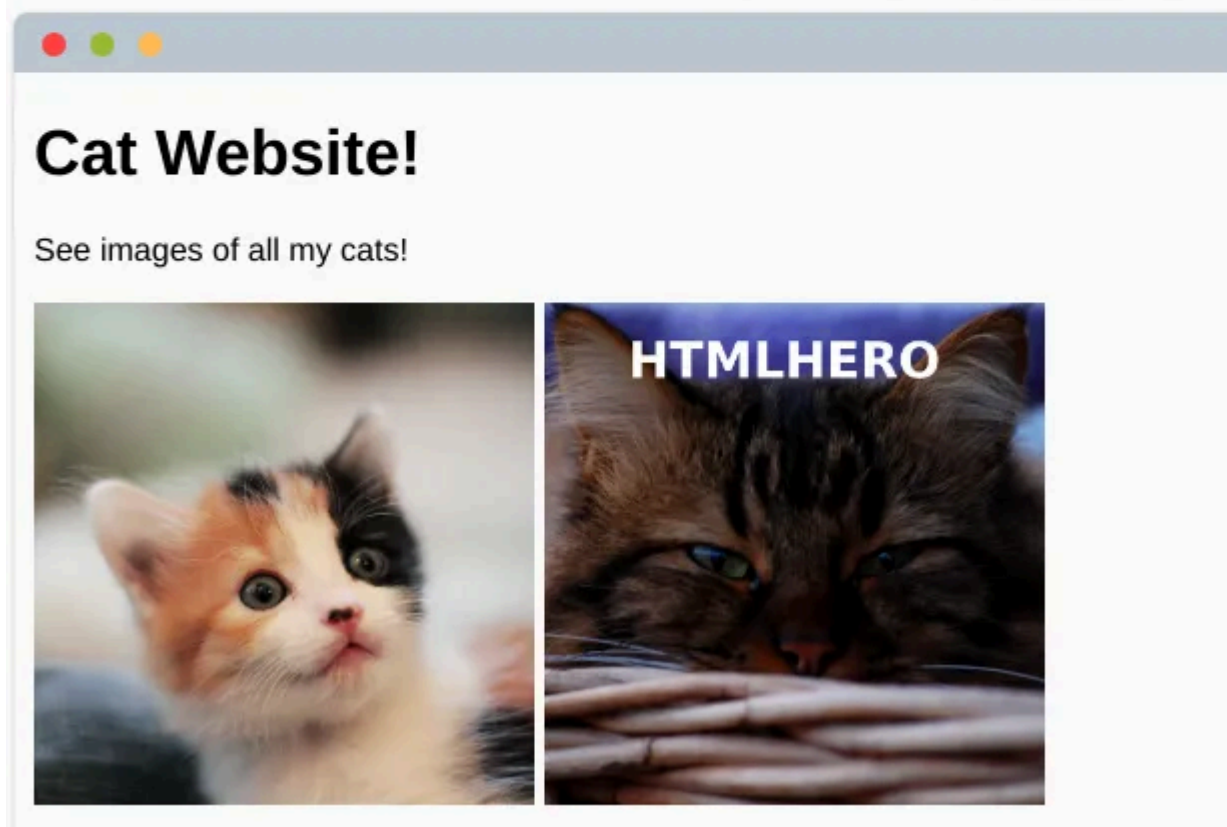
2.One of the images on the cat website is broken — fix it and the image will reveal the hidden text answer!

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>TryHackMe HTML Editor</title>
5   </head>
6   <body>
7     <h1>Cat Website!</h1>
8     <p>See images of all my cats!</p>
9     <img src='img/cat-1.jpg'>
10    <img src='img/cat-2.jpg'>
11    <!-- Add dog image here -->
12  </body>
13 </html>
```

Type HTML into the box above, then click the "Render HTML" button



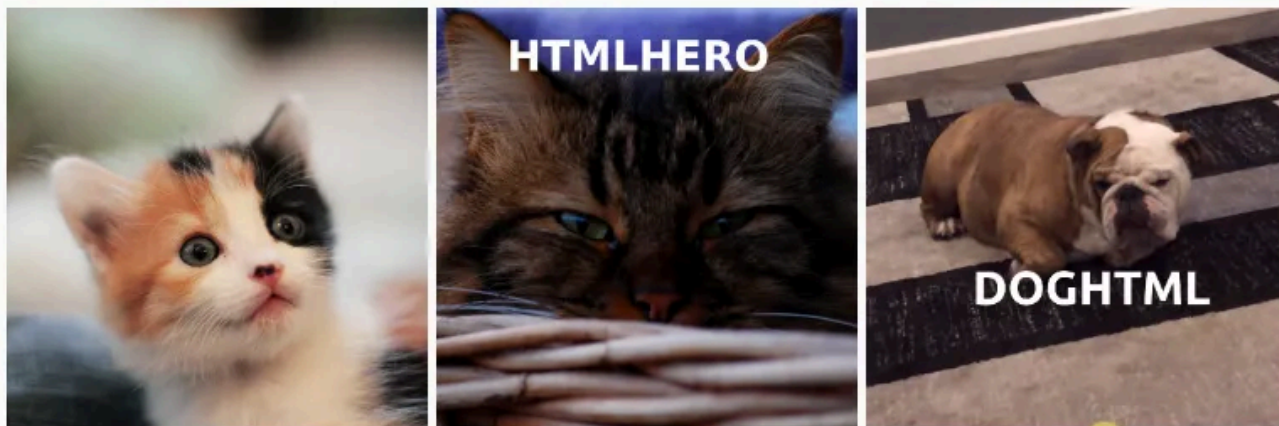
Rendered HTML Code



A:htmlhero

3.Add a dog image to the page by adding another img tag () on line 11. The dog image location is img/dog-1.png

See images of all my cats!



A:doghtml

JavaScript:

JavaScript (JS) is one of the most popular coding languages in the world and allows pages to become interactive. HTML is used to create the website structure and content, while JavaScript is used to control the functionality of webpages — without JavaScript a page would not have interactive elements, and would always be static. JS can dynamically update the page in real-time, giving functionality to change the style of a button when a particular event on the page occurs (such as when a user clicks a button), or to display moving animations.

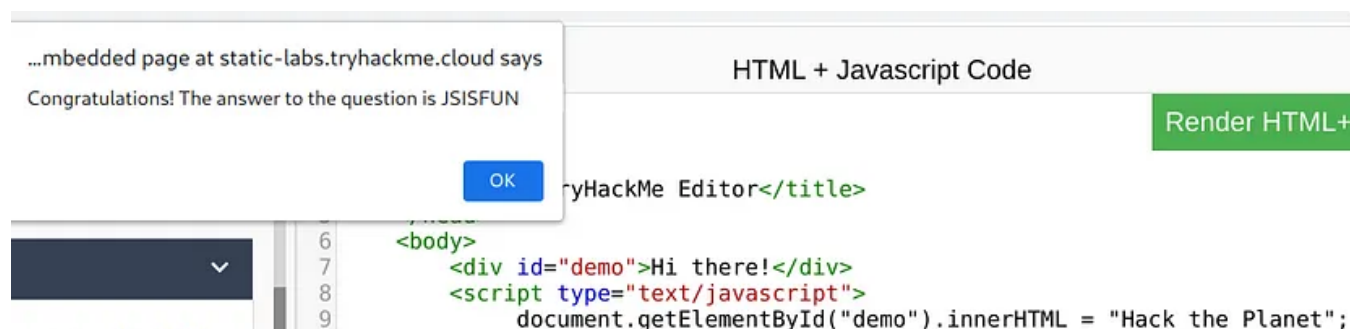
JavaScript is added within the page source code and can be either loaded within `<script>` tags or can be included remotely with the `src` attribute: `<script src="/location/of/javascript_file.js"></script>`

The following JavaScript code finds a HTML element on the page with the id of “demo” and changes the element’s contents to “Hack the Planet” :

```
document.getElementById("demo").innerHTML = "Hack the Planet";
```

HTML elements can also have events, such as “onclick” or “onhover” that execute JavaScript when the event occurs. The following code changes the text of the element with the demo ID to Button Clicked: `<button onclick='document.getElementById("demo").innerHTML = "Button Clicked";'>Click Me! </button>` - onclick events can also be defined inside the JavaScript script tags, and not on elements directly.

1.Click the “View Site” button on this task. On the right-hand side, add JavaScript that changes the demo element’s content to “Hack the Planet”



A:JSISFUN

2.Add the button HTML from this task that changes the element’s text to “Button Clicked” on the editor on the right, update the code by clicking the “Render HTML+JS Code” button and then click the button.

A:no answer need

Sensitive Data Exposure:

Sensitive Data Exposure is when a website doesn’t properly protect (or remove) sensitive clear-text information to the end-user; usually found in the frontend source code of sites.

We now know that websites are built using many HTML elements (tags), all of which we can see by simply “viewing the page source”, a website developer may have forgotten to remove login credentials, hidden links to private parts of the website or other sensitive data shown in HTML or JavaScript.

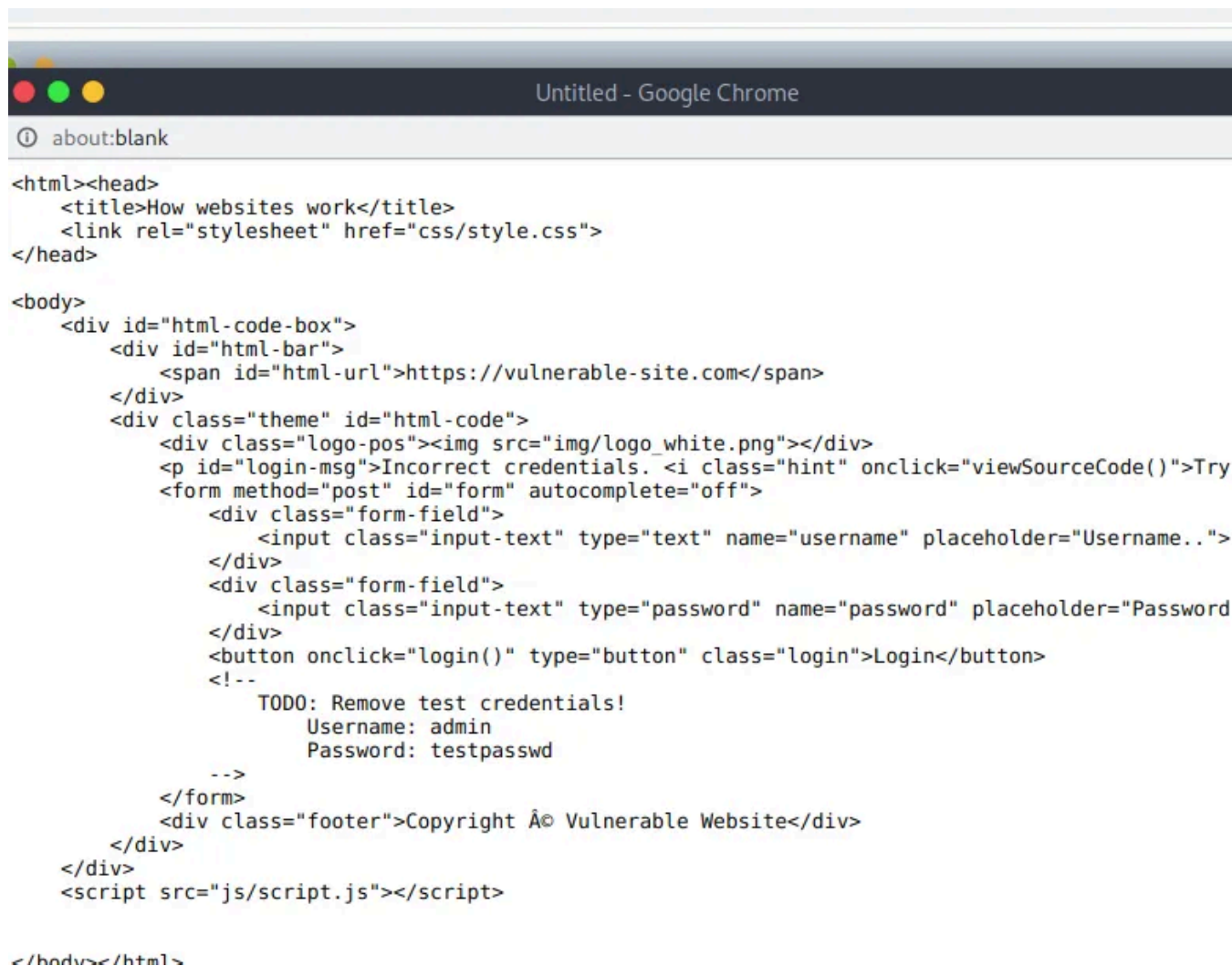

```
<!DOCTYPE html>
<html>
  <head>
    <title>Fake Website</title>
  </head>
  <body>
    <form>
      <input type='text' name='username'>
      <input type='password' name='password'>
      <button>Login</button>
      <!-- TODO: remove test credentials admin:password123 -->
    </form>
  </body>
</html>
```

[Open in app](#) ↗**Medium** Search

Sensitive information can be potentially leveraged to further an attacker's access within different parts of a web application. For example, there could be HTML comments with temporary login credentials, and if you viewed the page's source code and found this, you could use these credentials to login elsewhere on the application (or worse, used to access other backend components of the site).

Whenever you're assessing a web application for security issues, one of the first things you should do is review the page source code to see if you can find any exposed login credentials or hidden links.

1. View the website on this task. What is the password hidden in the source code?



```

<html><head>
  <title>How websites work</title>
  <link rel="stylesheet" href="css/style.css">
</head>

<body>
  <div id="html-code-box">
    <div id="html-bar">
      <span id="html-url">https://vulnerable-site.com</span>
    </div>
    <div class="theme" id="html-code">
      <div class="logo-pos"></div>
      <p id="login-msg">Incorrect credentials. <i class="hint" onclick="viewSourceCode()">Try
      <form method="post" id="form" autocomplete="off">
        <div class="form-field">
          <input class="input-text" type="text" name="username" placeholder="Username..">
        </div>
        <div class="form-field">
          <input class="input-text" type="password" name="password" placeholder="Password">
        </div>
        <button onclick="login()" type="button" class="login">Login</button>
      <!--
        TODO: Remove test credentials!
        Username: admin
        Password: testpasswd
      -->
      </form>
      <div class="footer">Copyright Â© Vulnerable Website</div>
    </div>
  </div>
  <script src="js/script.js"></script>

</body></html>

```

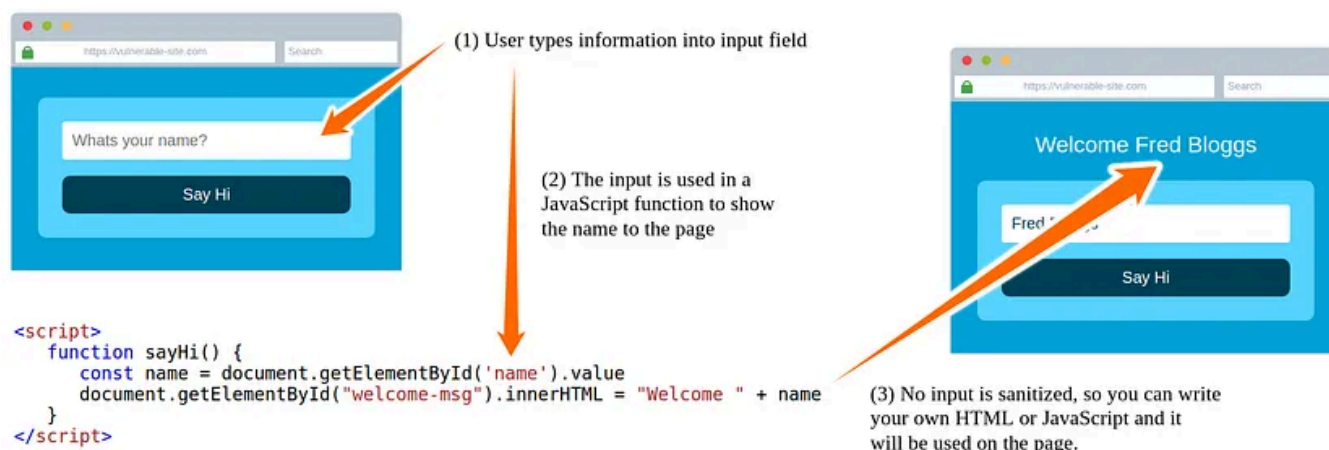
A:testpasswd

HTML Injection:

HTML Injection is a vulnerability that occurs when unfiltered user input is displayed on the page. If a website fails to sanitize user input (filter any “malicious” text that a user inputs into a website), and that input is used on the page, an attacker can inject HTML code into a vulnerable website.

Input sanitization is very important in keeping a website secure, as information a user inputs into a website is often used in other frontend and backend functionality — a vulnerability you’ll explore in another lab is database injection, where you can manipulate a database lookup query to login as another user by controlling the input that’s directly used in the query — but for now, let’s focus on HTML injection (which is client-side).

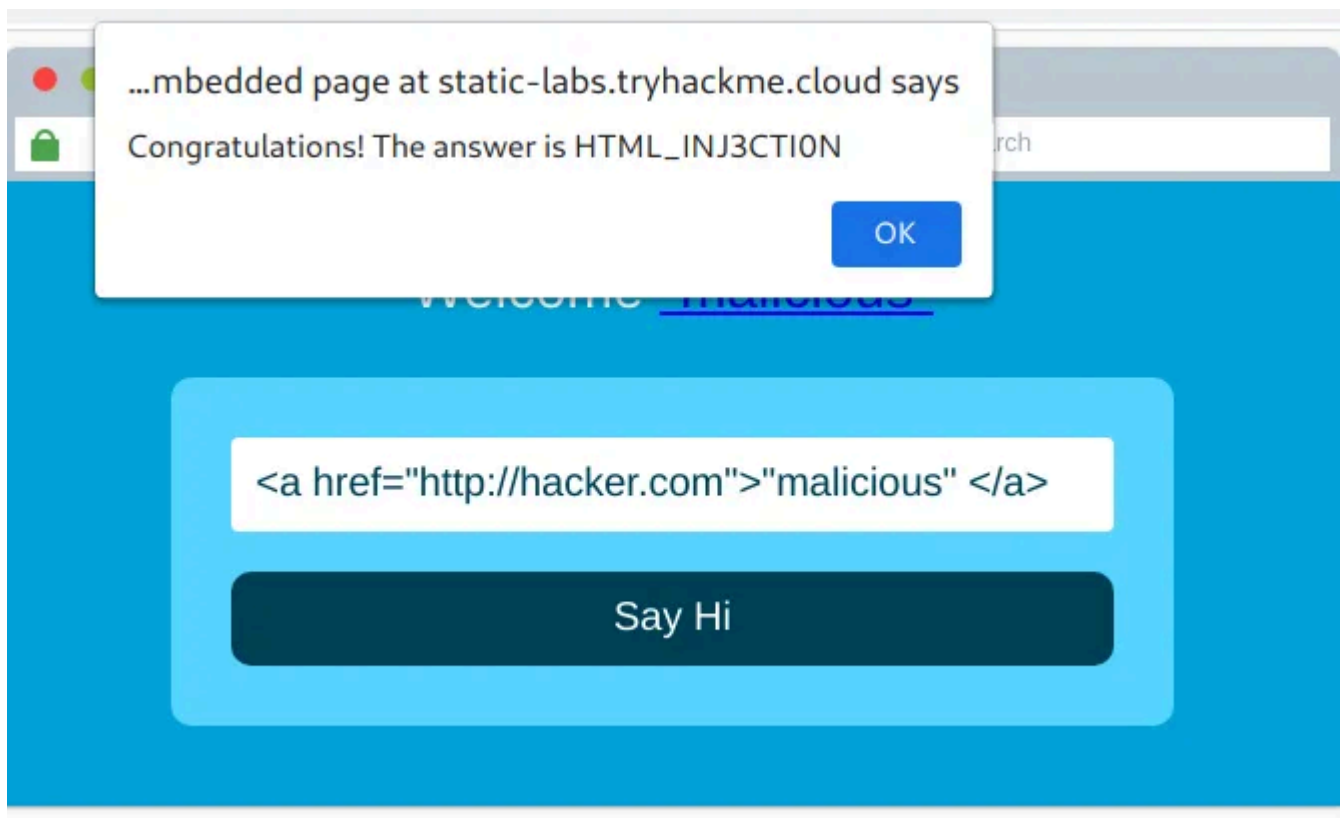
When a user has control of how their input is displayed, they can submit HTML (or JavaScript) code and the browser will use it on the page, allowing the user to control the page’s appearance and functionality.



The image above shows how a form outputs text to the page. Whatever the user inputs into the “What’s your name” field is passed to a JavaScript function and output to the page, which means if the user adds their own HTML or JavaScript in the field it’s used in the sayHi function and is added to the page — this means you can add your own HTML (such as a `<h1>` tag) and it will output your input as pure HTML.

The general rule is never to trust user input — to prevent malicious input the website developer should sanitize everything the user enters before using it in the JavaScript function; in this case, the developer could remove any HTML tags.

1. View the website on this task and inject HTML so that a malicious link to <http://hacker.com> is shown.



A:HTML_INJ3CTION

please everyone join my telegram channel :<https://t.me/hackerwheel>

please everyone join my youtube channel

:<https://www.youtube.com/channel/UC110XUIb7Ka6fsq1Pl7m0Hg>

Hackerwheel

Change the world

<https://t.me/hackerwheel>



Follow

Written by jagadeesh

35 Followers · 5 Following

CTF-PLAYER, security analyst, Pentesting, vapt, digital forensics

No responses yet



What are your thoughts?

Respond

More from jagadeesh



 jagadeesh

Tryhackme:Memory Forensics

Perform memory forensics to find the flags

Apr 9, 2021  53  1



 jagadeesh

Tryhackme:Introductory Researching

A brief introduction to research skills for pentesting.

Mar 27, 2021  8





 jagadeesh

Tryhackme:CC: Steganography

A crash course on the topic of steganography

Mar 29, 2021  100  2

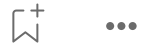


 jagadeesh

Tryhackme:Volatility

Learn how to perform memory forensics with Volatility!

Mar 18, 2021 17



See all from jagadeesh

Recommended from Medium



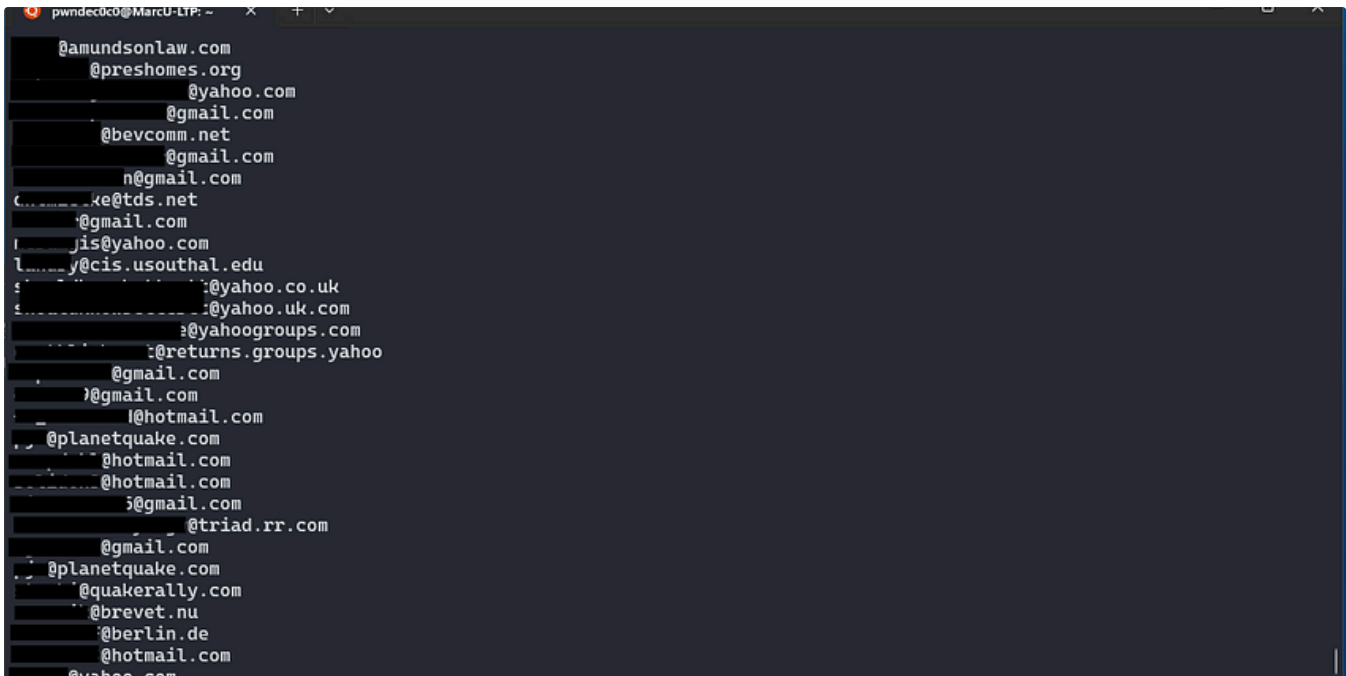
 In System Weakness by AbhirupKonwar


The best way to find private Bug-Hunting programs

 Recon process to find private programs

★ Dec 25, 2024 122 5





 In OSINT Team by PwnDec0c0

Exposing the Dark Side of Google Dorks: How I Extracted Millions of Emails.

How I was able to scrape large data of emails from the top mail domains(gmail,yahoo,hotmail etc.), .gov, .edu on of all countries and...

★ Nov 12, 2024 🖱 359 💬 2

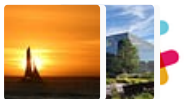


Lists



Staff picks

791 stories · 1541 saves



Stories to Help You Level-Up at Work

19 stories · 907 saves



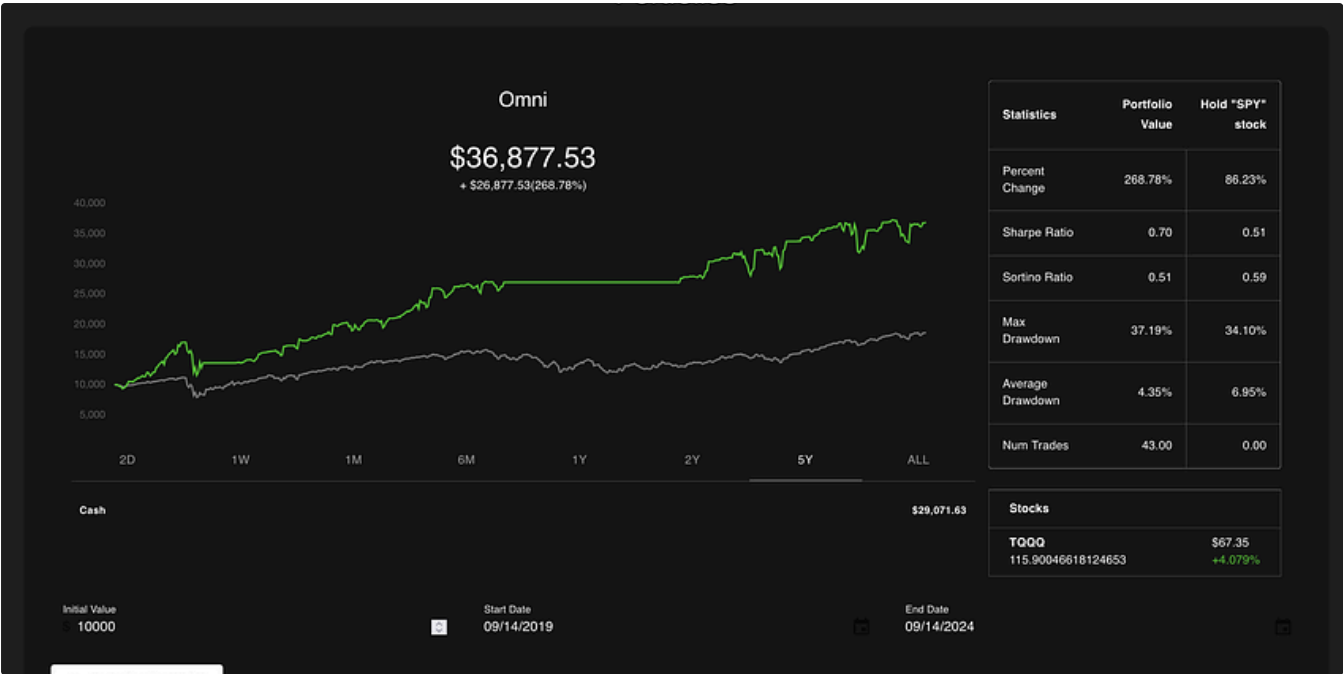
Self-Improvement 101


20 stories · 3177 saves



Productivity 101

20 stories · 2693 saves



 In DataDrivenInvestor by Austin Starks

I used OpenAI's o1 model to develop a trading strategy. It is DESTROYING the market

It literally took one try. I was shocked.


★ Sep 16, 2024 🖱️ 8.1K 💬 196



The screenshot shows the Capital One Shopping website interface. At the top, there's a search bar and navigation links. Below, a banner promotes personalized deals. Three main deal cards are visible:

- Walmart:** "Save at Walmart" with a "Get 1% Back" button.
- LEGO:** "Trending Product" - "LEGO - Bouquet of Roses" with a "Get up to 8% Back" button.
- VICTORIA'S SECRET:** "Price Drop" - "Everyday Fleece Polo Sweatshirt - Apparel - PINK" for \$12.99 (57% discount) with a "Get 6% Back" button.

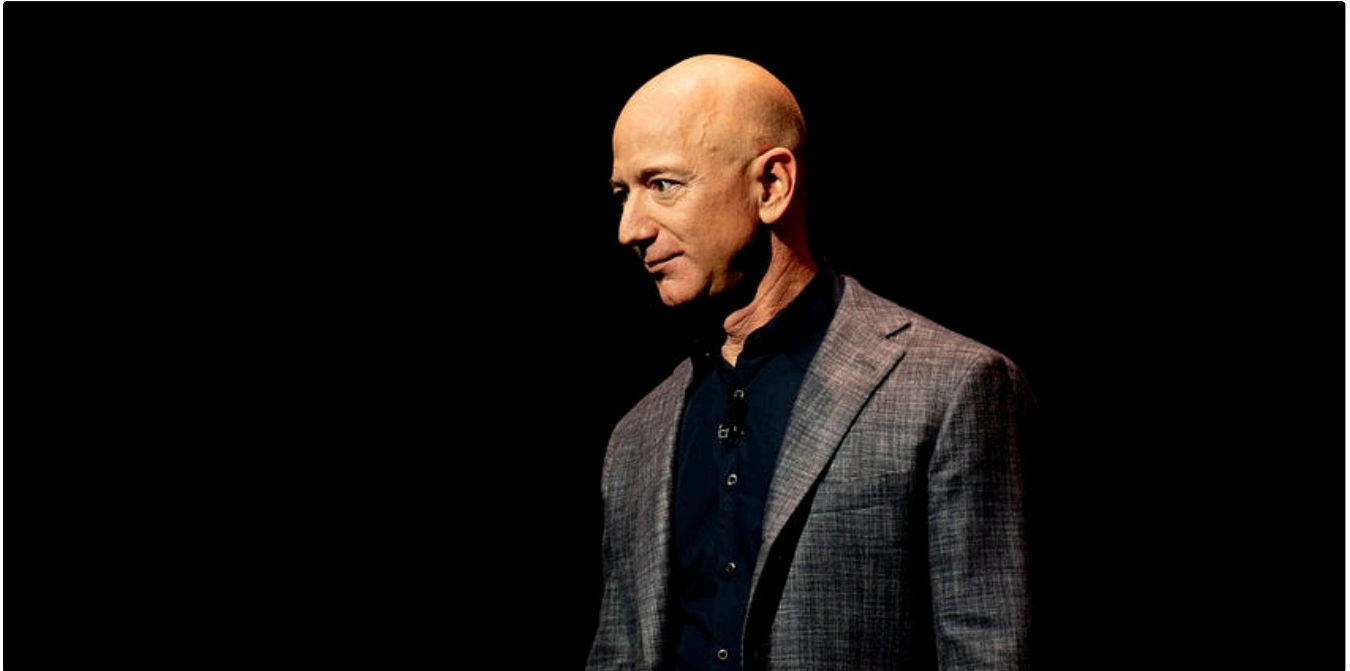
At the bottom, another "Trending Product" section is partially visible.

 Cosmos Kay

These 15 Websites Will Pay You DAILY Within 24 Hours (Easy Work At Home Jobs)

I found not five, not 10, but 15 websites that you can use right now to make money and get paid daily.

★ Dec 4, 2024 🖱 956 💬 44

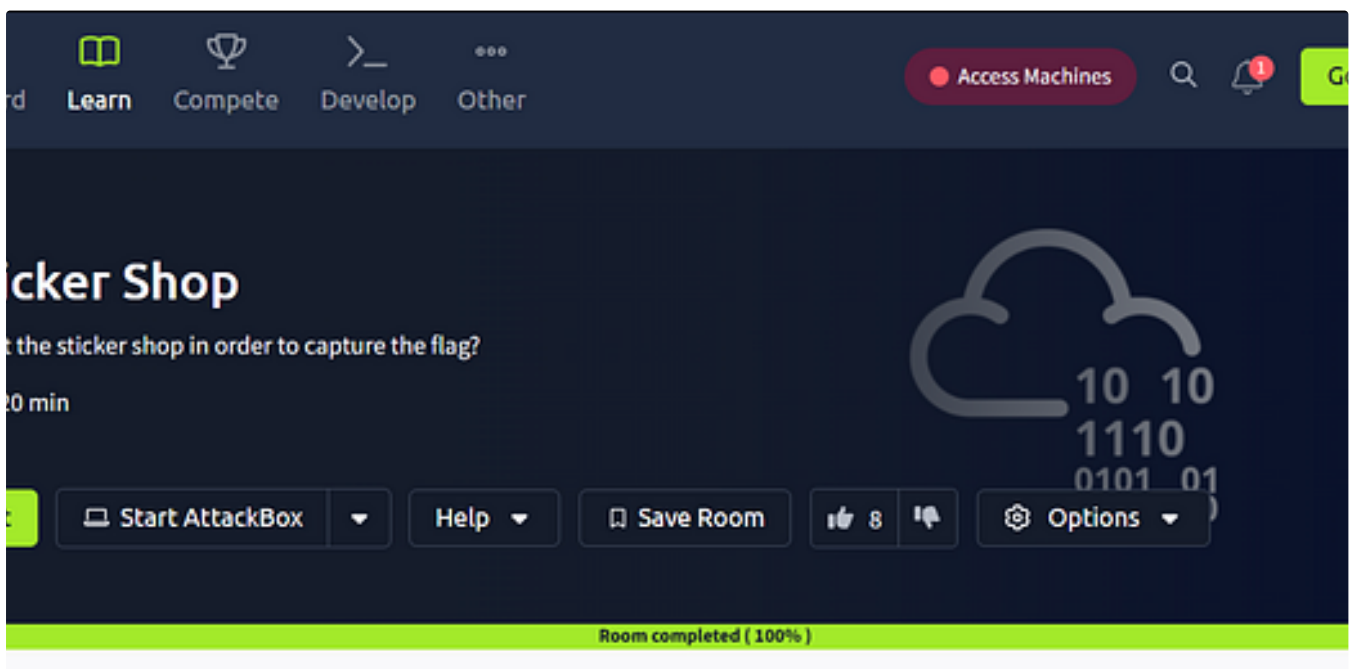


 Jessica Stillman

Jeff Bezos Says the 1-Hour Rule Makes Him Smarter. New Neuroscience Says He's Right

Jeff Bezos's morning routine has long included the one-hour rule. New neuroscience says yours probably should too.

★ Oct 30, 2024 🖱 18.5K 💬 486





Shakhawat Hossain - OxShakhawat

The Sticker Shop | TryHackMe | Walkthrough

How I Solved The Sticker Shop CTF: Exploiting Blind XSS to Capture the Flag. This writeup walks you through the steps of exploiting a Blind...

Nov 30, 2024



112



4



See more recommendations