How Websites Work

Learned:

- Visiting a Website starts the DNS request which gets the servers IP Address, then the browser connects to the IP Address on the Port for the protocol HTTPS or HTTP.

- It is important for a sites front end to be secure of any personal information in the code that could be easily accessed by hackers.

- Attackers can input HTML code into a website if data is not erased after a user inputs sensitive information.

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. It will respond with data that your browser uses to show you the page. A web server is just a dedicated computer somewhere 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.

Sensitive Data Exposure

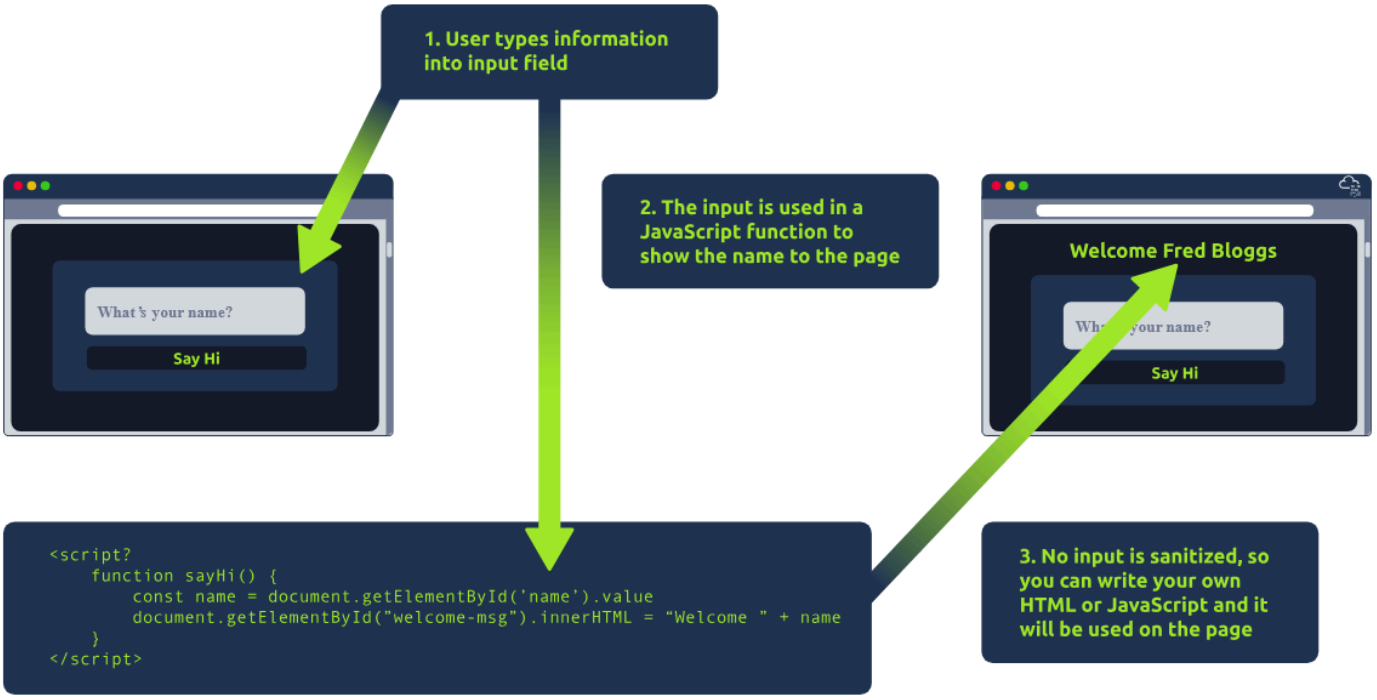
- Sensitive Data Exposure occurs when a website doesn't properly protect (or remove) sensitive text information to the end-user which is usually found in a site's frontend source code.
- Sensitive information can be used to further allow an attacker to access different parts of a web application. For example, there could be HTML codes with login information, and if you viewed the page's source code and found this, you could log in 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.

HTML Injection

HTML Injection is a vulnerability that occurs when a user input is displayed on the page. If a website fails to erase the data and that input is used on the page then the attacker can put HTML code into a vulnerable website.

- Input sanitization is important for website security because user input is often used by both the frontend and backend.

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 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 say Hi 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.