HTML INJECTION

What is HTML?

HTML stands for Hypertext Markup Language. It is a standard markup language for web pages. Collection of web pages makes a website. HTML elements are represented by <> tags. Where each tag has a different working.

Let's understand with an example:

Below is code of a simple HTML page.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Lets understand each tag one by one:

- The **<!DOCTYPE html>** declaration defines that this document is an HTML5 document.
- The **<html>** element is the root element of an HTML page.
- The **<head>** element contains meta information about the HTML page.
- The **<title>** element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab).

- The **<body>** element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The **<h1>** element defines a large heading.
- The **<p>** element defines a paragraph.

What is HTML Injection Attack?

HTML Injection is a vulnerability which occurs in web applications that allows users to insert HTML code via a specific parameter or an entry point.

HTML Injection is an attack that is similar to Cross-site Scripting (XSS). While in the XSS vulnerability the attacker can inject and execute JavaScript code, the HTML injection attack only allows the injection of certain HTML tags. When an application does not properly handle user supplied data, an attacker can supply valid HTML code, typically via a parameter value, and inject their own content into the page.

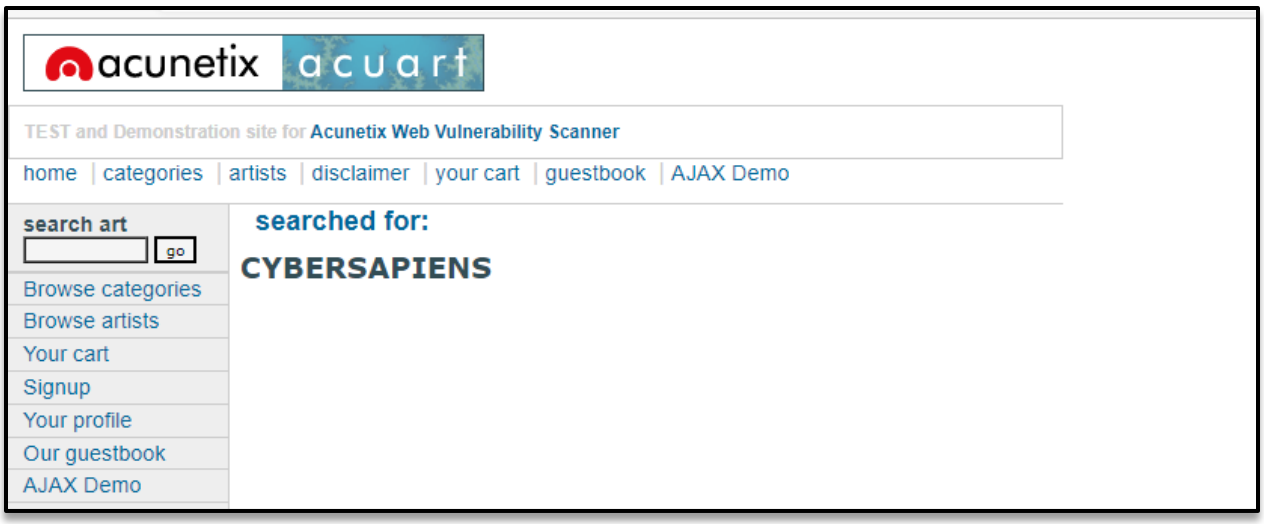
It is generally exploited using social engineering in order to trick valid users of the application to open malicious websites or to insert the credentials in a fake login form that will redirect the users to a page that captures cookies or credentials

Let's understand using an example

So currently I am on a vulnerable website which is: http://testphp.vulnweb.com

Notice the search box. Let's try to search something using it.

Alright! So, whatever I searched for is getting reflected on the webpage. Let's try to enter a simple HTML Injection Code which is: **<h1>CYBERSAPIENS</h1>** and check out if our payload is being executed or not.



Perfect! Notice the word **CYBERSAPIENS** now is executed with h1 tag which states that this web application is vulnerable to HTML Injection.

An attacker can also add some interesting offers using this html injection and redirect the user to its own desire website.

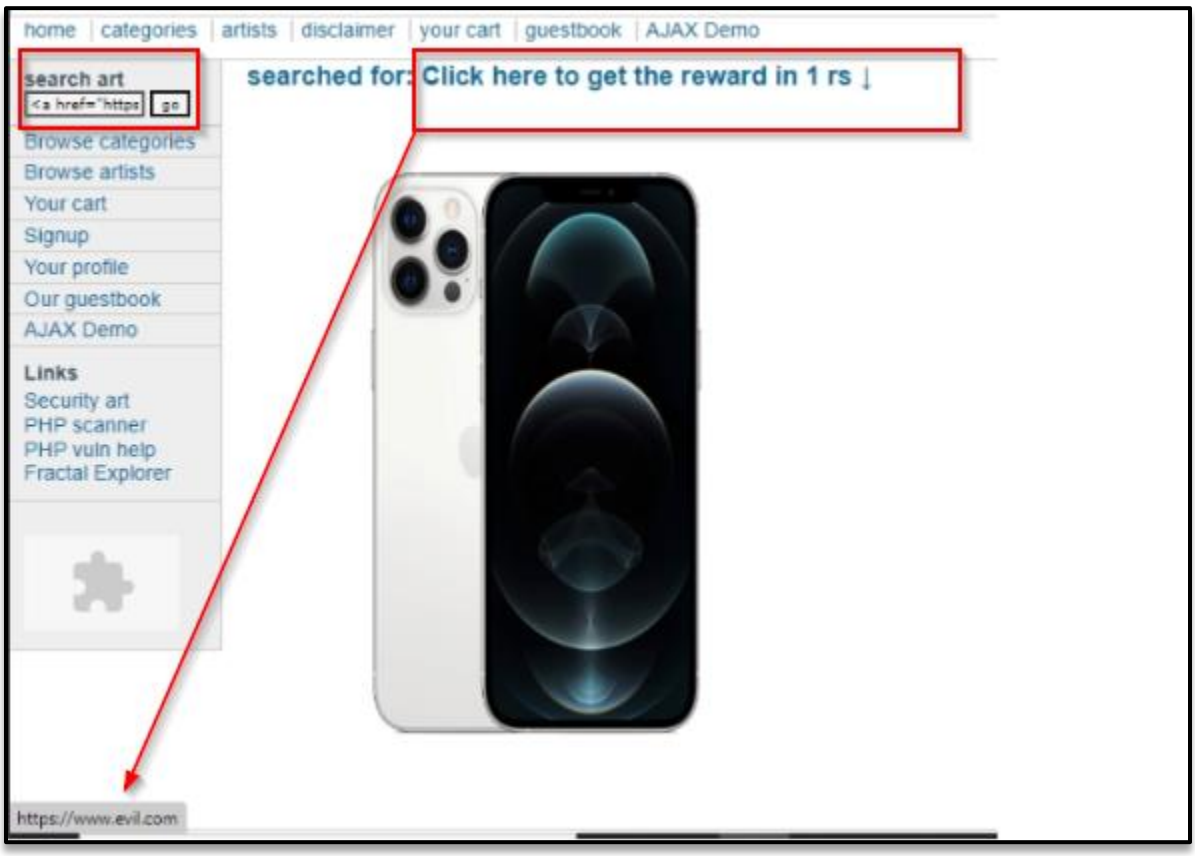
Let's understand using an example

An attacker inserted the below mentioned payload.

Payload

 Click here to get the reward in 1 rs ↓ <img src="https://m.media-amazon.com/images/I/71umuN8XVeL._FMwebp__.jpg" width="400" height="500">

The payload has an attacker directed URL. As shown below .



Types of html injection

- Stored HTML Injection
- Reflected HTML Injection

Stored HTML Injection

Stored injection attack occurs when malicious HTML code is saved in the web server and is being executed every time when the user calls an appropriate functionality.

The most common example of Stored HTML is the “comment option” in the blogs, which allow any user to enter his feedback as in the form of comments for the administrator or other users.

Reflected HTML Injection

The reflected HTML also known as “non-Persistence”

It is termed “non-persistent” In the reflected injection attack case; malicious HTML code is not being permanently stored on the web server.

Reflected Injection occurs when the website immediately responds to the malicious input. thus, the attacker needs to send the malicious link through phishing to trap the user.

Reflected HTML vulnerability can be easily found in website’s search engines

Reflect HTML is basically of three types:

- Reflected HTML GET
- Reflected HTML POST
- Reflected HTML Current URL

Exploiting HTML Injection

HTML Injections are easy to exploit. You just need to find out all parameters=values and check out each one of it for reflection of your HTML Injection Payload.

An HTML Injection vulnerability can be chained with an account takeover vulnerability.

The steps would be as follows:

- Attacker discovers injection vulnerability and decides to use an HTML injection attack
- Attacker crafts malicious link, including his injected HTML content, and sends it to a user via email
- The user visits the page due to the page being located within a trusted domain
- The attacker's injected HTML is rendered and presented to the user asking for a username and password
- The user enters a username and password, which are both sent to the attacker's server

Severity

The severity of HTML Injection can be categorized as P4 bug with a CVSS score of 0.1-3.9 which is Low. In case of an account takeover, it can be categorized as P3.

Impact of HTML Injection

Attacker can perform any action on the web page and can also create it as a phishing page to divert all users to other attacker-controlled web page.

Prevention of HTML Injection

- Every input should be checked if it contains any script code or any HTML code. One should check, if the code contains any special script or HTML brackets – <script></script>, <html></html>.

- There are many functions for checking if the code contains any special brackets. The selection of the checking function depends on the programming language that you are using.

References

- https://www.acunetix.com/vulnerabilities/web/htmlinjection/

- https://owasp.org/www-project-web-security-testingguide/latest/4-Web_Application_Security_Testing/11-Client-side_Testing/03-Testing_for_HTML_Injection

- https://www.imperva.com/learn/application-security/htmlinjection