HTML Injection

- HTML is the language that determines how application data (like a products' catalog) gets presented to users in their web browser. This language contains visualization commands, like the color of the page's background and the size of embedded pictures. It also contains links to other web pages, and additional commands intended for the user's browser. Furthermore, automated tools that collect useful information from the web on behalf of users often do so by systematically accessing and parsing the relevant information in the application's HTML pages.
- Hypertext Markup Language (HTML) injection is a technique used to take advantage of non-validated input to modify a web page presented by a web application to its users. Attackers take advantage of the fact that the content of a web page is often related to a previous interaction with users. When applications fail to validate user data, an attacker can send HTML-fomatted text to modify site content that gets presented to other users. A specifically crafted query can lead to inclusion in the web page of attacker-controlled HTML elements which change the way the application content gets exposed to the web.

Impact of HTML Injection:

- It can allow an attacker to modify the page.
- To steal another person's identity.
- The attacker discovers injection vulnerability and decides to use an HTML injection attack.

- Attacker crafts malicious links, 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.

Types of HTML Injection

This attack does not seem to be very difficult to understand or to perform, as HTML is considered as a quite simple language. However, there are different ways to perform this type of attack. We can also distinguish different types of this injection.

However, the main types are:

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.

Reflected HTML Injection

However, in the reflected injection attack case, malicious HTML code is not being permanently stored on the webserver. Reflected Injection occurs when the website immediately responds to the malicious input.

Reflected Injection attack can be performed differently according to the HTTP methods i.e, GET and POST. I would remind, that with POST method data is being sent and with GET method data is being requested.

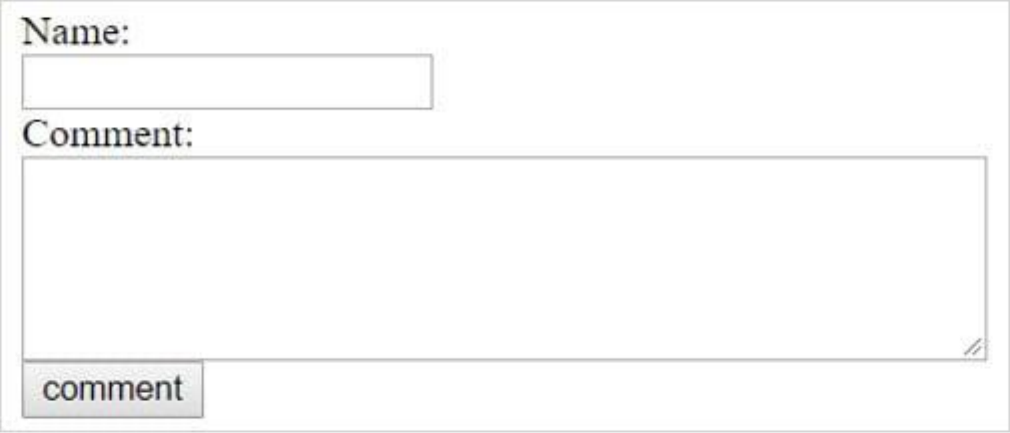
Reflected GET Injection occurs, when our input is being displayed (reflected) on the website. Suppose, we have a simple page with a search form, which is vulnerable to this attack. Then if we would type any HTML code, it will appear on our website and at the same time, it will be injected into the HTML document.

Reflected POST HTML Injection is a little bit more difficult. It occurs when a malicious HTML code is being sent instead of correct POST method parameters.

How is HTML Injection Performed?

- In order to perform this type of injection, firstly, the malicious user should find vulnerable parts of the website. Vulnerable parts of the website may be data input fields and website's link.
- Malicious HTML code can get into the source code by innerHTML. InnerHTML is the property of DOM document and with innerHTML, we can write dynamic HTML code. It is used mostly for data input fields like comment fields, questionnaire forms, registration forms, etc. Therefore, those elements are most vulnerable to HTML attack.

Suppose, if we have a comment form, then that is vulnerable to the HTML attack.



In the form, the user types his name and comment's text. All saved comments are listed in the page and loaded on the page load. Therefore, if malicious code was typed and saved, it also will be loaded and displayed on the website.

How to Test Against HTML Injection?

- When starting to test against possible injection attack, a tester should firstly list out all the potentially vulnerable parts of the website that is all data input fields, website’s link.
- When testing manually if an HTML Injection is possible, then simple HTML code could be entered, to check if the text would be displayed. There is no point to test with a very complicated HTML code, simple code may be enough to check if it is being displayed.
- If an HTML code being saved somewhere is displayed, then the tester can be sure, that this injection attack is possible. Then a more complicated code may be tried, to display the fake login form.
- Another solution is HTML Injection scanner. Scanning automatically against this attack may save a lot of your time. I would like to notify, that there are not many tools for HTML Injection testing in comparison with other attacks.
- However, one possible solution is WAS application. WAS can be named as a quite strong vulnerabilities scanner, as it tests with the different inputs and not just stops with the first failed.
- It is helpful for testing, maybe as mentioned in the above browser plugin “Tamper Data”, it gets sent data, allows the tester to change it and sends to the browser.
- We can also find some online scanning tools, where you only have to provide the website’s link and scanning against HTML attack will be performed. When testing is completed, the summary will be displayed.
- I would like to comment, that when selecting a scanning tool, we have to pay attention on how it analyzes the results and is it accurate enough or not.
- However, it should be kept in mind, that testing manually should not be forgotten. This way we can be sure what exact inputs are tried and what exact results we are getting. Also this way it is easier to analyze the results as well.

Prevention

- There is no doubt that the attack which occurred was mainly due to the developer's negligence and lack of knowledge. This type of injection attack occurred due to the non-validation of the input and output. It is therefore essential to have appropriate data validation in place to prevent such attacks.
- 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.imperva.com/learn/application-security/html-injection/

softwaretestinghelp.com/html-injection-tutorial/

https://www.vistainfosec.com/blog/comprehensive-guide-on-html-injection/