Cross Site Scripting (XSS)

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

Description

Cross-Site Scripting (XSS) attacks occur when:

1. Data enters a Web application through an untrusted source, most frequently a web request.
2. The data is included in dynamic content that is sent to a web user without being validated for malicious content.

The malicious content sent to the web browser often takes the form of a segment of JavaScript, but may also include HTML, Flash, or any other type of code that the browser may execute. The variety of attacks based on XSS is almost limitless, but they commonly include transmitting private data, like cookies or other session information, to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user's machine under the guise of the vulnerable site.

Type of xss

These 3 types of XSS are defined as follows:

1.Reflected XSS (AKA Non-Persistent or Type I)

2.Stored XSS (AKA Persistent or Type II)

3.DOM Based XSS (AKA Type-0)

Also Blind cross-site scripting

Reflected XSS Attacks:

Reflected attacks are those where the injected script is reflected off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. Reflected attacks are delivered to victims via another route, such as in an e-mail message, or on some other website. When a user is tricked into clicking on a malicious link, submitting a specially crafted form, or even just browsing to a malicious site, the injected code travels to the vulnerable web site, which reflects the attack back to the user's browser. The browser then executes the code because it came from a "trusted" server. Reflected XSS is also sometimes referred to as Non-Persistent or Type-I XSS (the attack is carried out through a single request / response cycle).

Resources and lab: https://portswigger.net/web-security/cross-site-scripting/reflected

Stored XSS Attacks:

Stored attacks are those where the injected script is permanently stored on the target servers, such as in a database, in a message forum, visitor log, comment field, etc. The victim then retrieves the malicious script from the server when it requests the stored information. Stored XSS is also sometimes referred to as Persistent or Type-II XSS.

Resources and lab: https://portswigger.net/web-security/cross-site-scripting/stored

DOM Based XSS Attacks:

DOM Based XSS (or as it is called in some texts, “type-0 XSS”) is an XSS attack wherein the attack payload is executed as a result of modifying the DOM “environment” in the victim's browser used by the original client side script, so that the client side code runs in an “unexpected” manner.

Resources and lab: https://portswigger.net/web-security/cross-site-scripting/dom-based

Blind cross-site scripting:

Blind cross-site scripting is a sub-type of stored/persistent cross-site scripting where the web application stores the payload sent by an attacker and only executes it later – at a different time or in a different place, possibly even in another web application.

Resources and lab:

https://www.acunetix.com/websitesecurity/detecting-blind-xss-vulnerabilities/

More Information:

1. https://owasp.org/www-community/attacks/xss/
2. https://www.acunetix.com/blog/web-security-zone/test-xss-skills-vulnerable-sites/
3. https://xss-game.appspot.com/

4. https://pentesterlab.com/exercises/xss_and_mysql_file/course
5. https://portswigger.net/web-security/cross-site-scripting