Ray Franklin

Cross Site Scripting (XSS)

Jul 12, 2021

Attack Description

Cross site scripting is a common vulnerability where the attacker injects script based code into a website. The browser interprets the code and executes the attacker's script. There are two basic styles of XSS and one lesser common version.

- Stored XSS Attacks: This occurs when the script is stored on the server and the script is run each time the data is retrieved from that location..
- Blind Cross-Site Scripting: Similar to a stored attack, but the stored information is only triggered from the backend application, usually when accessed by an administrator.
- Reflected XSS Attacks: This is an attack that is non-persistent. It is not stored in the server, but rather it is sent to the server then executed once the code returns to the client. Often the attacker will use a malicious link via email, the goal is to trick the user into clicking this link, which will send the script to the server and execute when reflected back to the user.

Attacker perspective

The attacker will look for a field that accepts an input text from the user. Typically this will be a search field, a user login page, or perhaps a blog post. Regardless of whether the target field is linked to a database, the attacker can still create a script that breaches security.

Vulnerability: Usually the attacker will input test html text to check whether the site has vulnerable fields in it. This step can be easily automated to check many sites with minimal effort. Once a vulnerability has been identified, the attack can begin.

Implementation: The attacker will typically have two goals in mind with an attack. The first goal uses a script to steal the user's session information, typically stored in cookies. This is referred to as a session hijacking attack. The attacker sniffs the site to find the current user's session cookies, then using a script command they store that information. Once the attacker has the information, they can navigate to the site with the user's cookies now in the attackers cache allowing them to pose as the original user.

The second attack goal is to completely take over the server or host machine. This can occur when the script is executed on the server side. The attacker enters their malicious code, and upon submitting, the server carries out the code. The first step is to create a reverse shell on the server and then to escalate permissions until the attacker has administrator privileges.

Web application perspective

Detection: Any user input from a web application will be subject to XSS attacks unless proper actions are taken to secure the input. This means the vast majority of inputs will be vulnerable by default, and preventative measures will need to be taken. You can perform some simple tests by sending one of the many filter evasion methods found on OWASP's website. It will provide a large list of possible scripts to try and see if your site is vulnerable. Effectively, if the script successfully runs, your site is not secure.

Solution: The best method to prevent XSS is by sanitizing inputs. This is performed by utilizing various functions built into the coding languages themselves. Using these security encoding

libraries will help filter out any special characters that allow the attacker to execute scripts. If you must allow some level of special characters in your inputs, it can be done such that you disallow all of them by default and add in only the ones you need to allow. This does open your site to a potential attack, but the likelihood decreases significantly by using this method.

Further prevention: To further prevent threats, the developer should have the mindset that all user data is not to be trusted. If you assume every input form a user is malicious, you may seem a bit paranoid, but the reality is that you will be one of the few who are building secure sites. This does mean every bit of information input into the application will need to be carefully reviewed and controlled. Through completely managing the user inputs before performing any actions with the information you can build a secure application.

References for sources used

Cross Site Scripting Prevention Cheat Sheet¶. Cross Site Scripting Prevention - OWASP Cheat Sheet Series. (n.d.). https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html#Why_Can.27t_I_Just_HTML_Entity_Encode_Untrusted_Data.3F.

DrapsTV. (2015, January 22). *XSS Tutorial #1 - What is Cross Site Scripting?* YouTube. https://www.youtube.com/watch?v=M_nIIcKTxGk.

Wikimedia Foundation. (2021, July 10). *Cross-site scripting*. Wikipedia. https://en.wikipedia.org/wiki/Cross-site_scripting.

XSS Filter Evasion Cheat Sheet. OWASP. (n.d.). https://owasp.org/www-community/xss-filter-evasion-cheatsheet.