LU10a - Cross-Site Scripting (XSS) Attacks

Cross-site scripting (XSS) is a very common type of web application attack vector in which malicious code is injected into a vulnerable web application. Unlike other attacks that target the application's server or database directly, XSS targets the **users of the web application** — because the injected code runs in their browsers in the context of the legitimate website.

A successful XSS attack can cause serious damage — including compromised user accounts, activation of Trojan code, manipulation of page content to trick users into sharing sensitive data, or exposure of session cookies that allow attackers to impersonate valid users.

Types of XSS Attacks

There are **two major types** of cross-site scripting attacks commonly discussed:

- **Stored XSS (Persistent)** – A malicious script is permanently injected into an application and served to all users.
- **Reflected XSS (Non-Persistent)** – A malicious script is embedded in a URL or request, then reflected off the web server back to the user's browser when the link is visited.

What is Stored Cross-Site Scripting

To execute a **stored XSS attack**, the attacker must find a vulnerability in a web application where user input is stored without proper validation or escaping. A common example is when a comment field or form accepts HTML input and embeds it directly into pages viewed by other users.

Example: An attacker enters a comment containing JavaScript that steals session cookies. Every visitor who views the comment will run the malicious script unknowingly.

Stored XSS Attack — Step by Step

1. The attacker discovers a page with an input field that allows HTML. 2. They insert malicious JavaScript code into that field. 3. The application stores and later serves that code as part of normal content. 4. When other users visit the page, their browser executes the attack script.

How Stored XSS Endangers Users

Stored XSS is especially dangerous because:

* It can impact **all users** who view the infected page. * Attacker-controlled scripts can steal session credentials or redirect users to phishing sites. * Malicious payloads can embed external JavaScript

that reports user data back to the attacker.

Related Topics

[1]: https://www.imperva.com/learn/application-security/cross-site-scripting-xss-attacks/?utm_source=chatgpt.com „What is XSS | Stored Cross Site Scripting Example - Imperva“ [2]: https://www.imperva.com/learn/application-security/reflected-xss-attacks/?utm_source=chatgpt.com „Reflected XSS | How to Prevent a Non-Persistent Attack - Imperva“ [3]: https://developer.mozilla.org/docs/Web/Security/Attacks/XSS?utm_source=chatgpt.com „Cross-site scripting (XSS) - Security | MDN“