CSRF attacks

Definition

SRF (Cross-Site Request Forgery) attacks involve tricking authenticated users into unknowingly performing actions on a web application. The attacker exploits the trust between the victim's browser and the application to execute unauthorized actions.

The classic example is a victim receiving an email containing an image tag that secretly initiates a fund transfer request to an attacker's website. When the victim's browser loads the image, it sends a request to the attacker's site with the victim's session cookie. As a result, funds are transferred from the victim's account to the attacker's account without the victim's knowledge or consent. This demonstrates how CSRF attacks exploit trust to perform unauthorized actions on web applications.

Security risks

The security risks related to CSRF attacks can be significant:

1. **Unauthorized actions**: CSRF attacks allow attackers to perform actions on behalf of the victim, potentially leading to unauthorized changes, data breaches, or financial loss.
2. **Bypassing authentication**: As CSRF attacks use the victim's authenticated session, they can bypass any authentication checks implemented by the web application, making it difficult to detect and prevent such attacks.
3. **Trust exploitation**: The attack leverages the trust relationship between the victim's browser and the web application, taking advantage of the fact that the web application treats the victim's requests as legitimate.
4. **Social engineering**: CSRF attacks often rely on social engineering techniques to deceive users into visiting malicious webpages or clicking on malicious links, making them more susceptible to exploitation.

How to prevent it

There is 2 solutions that can be used altogether to prevent CSRF attacks :

- **Implement anti-CSRF tokens**: Include a unique and random token in each HTML form or request that modifies state on the server. The token should be validated before processing the request, ensuring that it originated from the correct page and not an attacker.

> Anti-CSRF token are implemented in most of frameworks.
> But you often need to **enable them and enable the CSRF check**.

- **Define cookie's samesite attributes as** Lax **or** Strict
  - With the SameSite attribute set to Lax cookies are allowed to be sent with cross-site requests that are initiated by top-level navigation (for example, when a user clicks on a link). However, they are not sent for cross-site requests triggered by other means, such as through the use of an image tag or a form submission from another site.
  - When the SameSite attribute is set to Strict, cookies are not sent with any cross-site requests, regardless of how the request is initiated. They are only sent for requests originating from the same site or domain as the web application.

> **Do not use** `None` attribute as it enables the use of the cookie in cross site request, no matter the origin.

Revision #3
Created 12 April 2024 03:40:32 by Seaweedbrain
Updated 12 April 2024 03:53:31 by Seaweedbrain