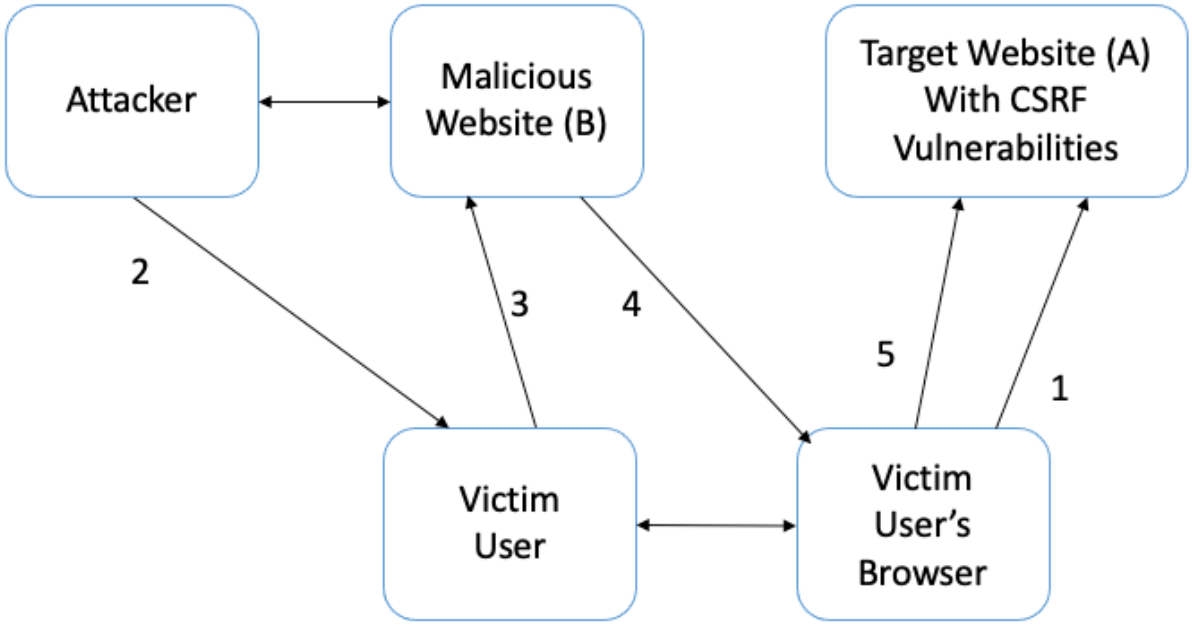
How Cross Site Request Forgery (CSRF) attack works:

CSRF attacks exploit "***the trust that a website has in a user's browser***".



1. Victim user has an active session (authenticated) with website A. Victim user's browser holds a valid session cookie for website A. The browser attaches the session cookie for both same-site and cross-site requests. Using the session cookie alone, the website A can't tell if a request from victim's browser is *same-site* or *cross-site*.
2. The attacker sends an inviting message/email with a link to a page on the malicious website B.
3. Victim user clicks on the link in the message/email which sends a request to a page on website B.
4. The malicious page from website B loads into victim user's browser.
5. The malicious page displays some interesting text but also sends a forged/unauthorized request to website A on behalf of the victim user without his/her knowledge. This request causes a state change for victim user on website A. The attack is a success. If the victim user is not in session on website A, the CSRF attack will fail.

Countermeasures for CSRF Attacks:

1. **`Referer` header**: The browser adds this header to the request that a server can use to check if the request originated from one its pages or not.
2. **`SameSite` attribute to cookies**: This attribute is set on a cookie by the server which tells a browser whether a cookie should be attached to a cross-site request or not.
3. **`Secret Token` approach**: The server embeds a secret random value(s) to all pages where an action can be initiated (like forms that request state changes). The secret random value(s) are included when the action/request is submitted to the server. The server checks these values to defend against the CSRF attacks. Pages from a different (like website B) origin will not have access to these secret values. The browser enforces the same origin policy.

In CSRF lab, the Elgg web application uses the secret token approach using two secret random values: **`__elgg_ts`** and **`__elgg_token`**.