Cross-Site Request Forgery, or CSRF is an attack that forcefully makes an authenticated user to submit a malicious request against a Web application they are authenticated with. This attack intends to exploit the trust of a Web application on an authenticated user. The aim of the attacker behind conducting a CSRF attack is to make users submit a request for changing the state.

For example:

* Submitting a transaction
* Changing a password
* Deleting or submitting a record
* Sending a message
* Purchasing a product

Attackers often launch CSRF attacks by using social engineering methods. They trick a victim by making them click a malicious URL that sends an unauthorized request for a web app. The browser of user sends this malicious request to a particular web application. It also includes website-related credentials such as session cookies. When a user has an active connection with the targeted application, it treats the new request like an authorized request by user. This makes the attack successful.

# How does Cross-Site Request Forgery Work?

Cross-site Request Forgery attack targets those web applications that cannot distinguish between valid and forged requests. An attacker can use many methods to exploit a vulnerability in an application to conduct CSRF.

Let’s consider an example. Brian has an online account and visits his bank site regularly to carry out transactions with his brother David. He is not aware that his bank’s website is vulnerable to CSRF attacks. A hacker plans to send $10,000 dollars from Brian’s account by exploiting the vulnerability. To launch an attack successfully,

* The attacker will create an exploit URL
* They will trick Brian by making him click the exploit URL
* Brian must be in an active session with the website when the attacker launches the attack.

By using different attack methods through social engineering, the attacker tricks Brian into loading the infected URL. They can do this by putting a malicious URL on pages the user often accesses while logged in, including malicious HTML images into a form, or by simply sending an email with a malicious URL.

# The Limitations

However, there are certain limitations for carrying out a successful CSRF attack. A CSRF attack’s success mainly depends upon a user’s active session with the vulnerable application. If the user is not in an active session, the attack cannot be successful. Moreover, the attacker needs to find a valid URL for crafting it maliciously. This URL must be able to change the state of the target application. They must also find the correct URL parameter values, or the target application may possibly not accept the malicious request.

To find out more about how to protect your website from CSRF attacks, get in touch with Aardwolf Security and talk to our experts.