**Cross-site request forgery**

Cross-site request forgery (also known as CSRF) ranks at 8th position according to 2013 OWASP.

When we dont log out from the browser saves our session cookie as a part of remember me.That session cookie authenticates you to the website. The browser that you are using receives the session cookie, stores and sends it along with every request.So the attack happens when the attacker sends the malicious link to the victim and the victim's credentials get used for the attack as they were saved in browser's cookies.Attacker take some actions in the victim's profile which were not meant to be performes by him. It allows an attacker to partly circumvent the same origin policy, which is designed to prevent different websites from interfering with each other.

METHOD OF CREATING CSRF POC USING BURPSUITE:

Select a request anywhere in Burp Suite Professional that you want to test or exploit.

From the right-click context menu, select Engagement tools / Generate CSRF PoC.

Burp Suite will generate some HTML that will trigger the selected request (minus cookies, which will be added automatically by the victim's browser).

You can tweak various options in the CSRF PoC generator to fine-tune aspects of the attack. You might need to do this in some unusual situations to deal with quirky features of requests.

Copy the generated HTML into a web page, view it in a browser that is logged in to the vulnerable web site, and test whether the intended request is issued successfully and the desired action occurs.

PRECAUTIONS:

1.Log out of sites after using: CSRF attacks rely on the victim having a valid session for a particular site. When you log out, you invalidate the session on that site, leaving no sessions for the hacker to exploit. (The cookies stored in your browser will no longer work.)

2.Use a different browser for sensitive sites: CSRF attacks also rely on the victim’s browser storing the credentials of the vulnerable website. By using separate browsers for sensitive activities (such as banking sites) and casual browsing, you can make sure that if you do stumble upon a CSRF exploit page during casual browsing, the browser that you are using will not have the credentials to sensitive sites.

3.Use a Javascript blocker: JavaScript is often used to automatically submit the forms used to exploit a CSRF vulnerability. By disabling Javascript on your browser or email client, you essentially render auto-submit forms useless. This way, CSRF attacks would become harder since attackers would have to trick you into manually submitting the form instead.

4.Requiring the user to reauthenticate, or prove they are a user (e.g., via a CAPTCHA) can also protect against CSRF.

POCs:

1.On creating the poc of logging out it can be use to log out any user when other logged in user click on that generated poc.

https://www.youtube.com/watch?v=-zqYRGfomf8&t=31s

2.With every user login a cookie is generated.Nowlogot of the profile and then again login (with new cookie). When some changes were made on the profile and the old cookie was pasted the results were reflected ie old information of the profile were displayed even when the changes were made.

https://www.youtube.com/watch?v=-UMcCwfvEEY

3.The attacker first intercept the request of changing the email address of the user. Then using burp a poc was made in which the values of user (such as email add, name etc) were manipulated and when the request was opened on the browser those changes were reflected on the account.

https://www.youtube.com/watch?v=U1qdxHsLJHE

4.Stored XSS via CSRF in \*.sony.co.in. In this attack 2 accounts were made in which the attacker used xss payload in address field which was stored xss vuln. On making the poc of the request and using that on diff browser of the victim it was giving the cookie of the victim which was the payload of the attacker.

https://www.youtube.com/watch?v=dBTuWzX8hd0

5.On changing the value of the generated poc it was giving the full account takeover i.e the change password option was aslo vulnerable to csrf.

https://www.youtube.com/watch?v=RwrKJnccXvM

https://owasp.org/www-pdf-archive/OWASP\_Top\_10\_-\_2013.pdf

https://portswigger.net/web-security/csrf

https://medium.com/swlh/intro-to-csrf-cross-site-request-forgery-9de669df03de