A great resource I highly recommend you check out is https://github.com/masatokinugawa/filterbypass/wiki/Browser's-XSS-Filter-Bypass-Cheat-Sheet

Cross Site Request Forgery (CSRF)

CSRF is being able to force the user to do a specific action on the target website from your website, usually via an HTML form (<form action="/login" method="POST">) and is rather straightforward to find. An example of a CSRF bug is forcing the user to change their account email to one controlled by you, which would lead to account takeover. Developers can introduce CSRF protection **very easily** but still some developers opt to create custom code instead. When first hunting for CSRF bugs I look for areas on the website which **should** contain protection around them, such as updating your account information. I know this sounds a bit silly but from actually proving a certain feature **does** have security can again give you a **clear indication to the security throughout the site**. What behavior do you see when sending a blank CSRF value, did it reveal any framework information from an error? Did it reflect your changes but with a CSRF error? Have you seen this parameter name used on other websites? Perhaps there isn't even any protection! Test their most secure features (*account functions usually as mentioned above*) and work your way backwards. As you continue testing the website you may discover that some features have **different** CSRF protection. Now consider, why? Different team? Old codebase? Perhaps a different parameter name is used and now you can hunt specifically for this parameter knowing it's vulnerable.

One common approach developers take is checking the referer header value and if it isn't their website, then drop the request. However this backfires because sometimes the checks are **only** executed **if** the referer header is actually found, and if it **isn't**, **no** checks done. You can get a blank referrer from the following: