```
<meta name="referrer" content="no-referrer" />
<iframe src=”data:text/html;base64,form_code_here”>
```

As well as this sometimes they'll only check if their domain is found in the referer, so creating a directory on your site & visiting https://www.yoursite.com/https://www.theirsite.com/ may bypass the checks. Or what about https://www.theirsite.computer/ ? Again, **to begin with I am focused purely** on finding areas that should contain CSRF protection (sensitive areas!), and then checking if they have created custom filtering. Where there’s a filter there is usually a bypass!

When hunting for CSRF there isn’t really a list of “common” areas to hunt for as every website contains different features, but typically all sensitive features should be protected from CSRF, **so find them and test there**. For example if the website allows you to checkout, can you force the user to checkout thus forcing their card to be charged?

Open url redirects

My favorite bug to find because I usually have a 100% success rate of using a “harmless” redirect in a chain if the target has some type of Oauth flow which handles a token along with a redirect. Open URL redirects are simply urls such as https://www.google.com/redirect?goto=https://www.bing.com/ which when visited will redirect to the URL provided in the parameter. A lot of developers fail to create any type of filtering/restriction on these so they are **very very** easy to find. However with that said, filters sometimes can exist to stop you in your tracks. Below are some of my payloads I use to bypass filters but more importantly used to determine how their filter is working.

```
\/yoururl.com
\/\/yoururl.com
\\yoururl.com
//yoururl.com
```