Clickjacking

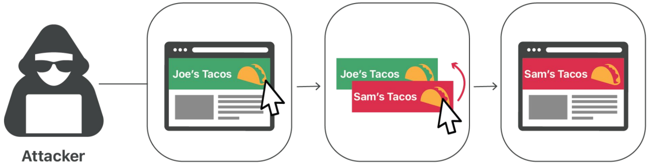
Jack Wolak, Hunter Sutton, Matteo Ciavaglia, Andrew Slayton

What is Clickjacking And Example

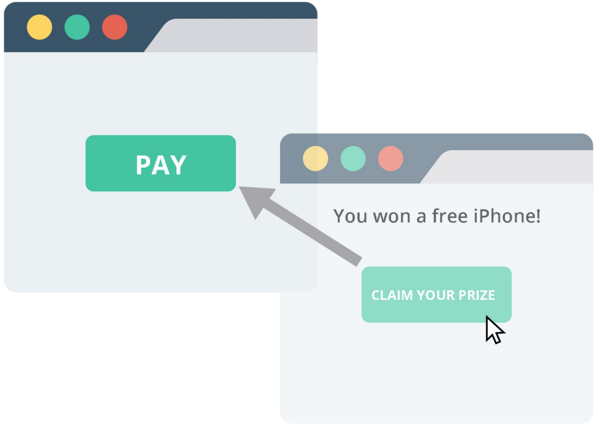
There can be two different versions of clickjacking.

- When a user clicks on an ad they see and it redirects them to a different website not related to the one they clicked. In short, the website they actually go to steals the click and the ad networks pay them instead of the seen ad.

- Another way of clickjacking is instead of both websites potentially being legit, the website you are redirected to is a malicious website that may steal your information. This can be done by someone putting an invisible frame over the existing button on the page.

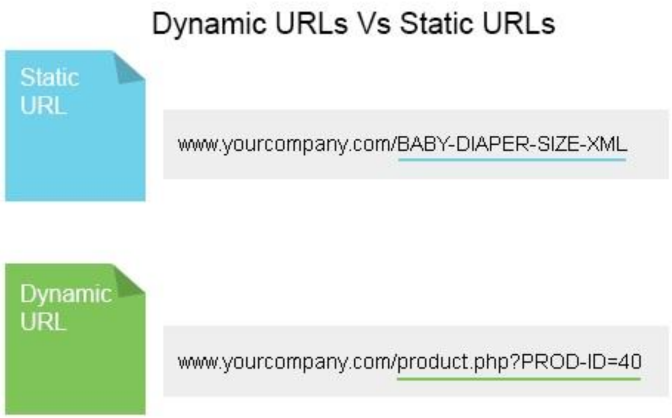


Example 2



In this example of clickjacking, the user has already put their bank information below and they believe they have won a new iPhone. When they click the “CLAIM YOUR PRIZE” button, they are actually clicking a “PAY” button that takes their bank information and pays the click-jacker.

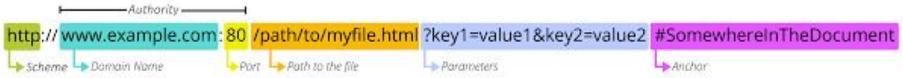
A Problem with Current Solutions


Dynamic URLs Vs Static URLs
Static URL
www.yourcompany.com/BABY-DIAPER-SIZE-XML
Dynamic URL
www.yourcompany.com/product.php?PROD-ID=40

A current solution to clickjacking is using whitelisting to only allow certain URLs to be used on the webpage.

Dynamic URLs make this more difficult because every person who clicks on the link will result in a new URL that won’t currently be in the whitelist.

Our Solution



Our solution to dynamic URLs consists of parsing the URL so that they can be used more easily. The sections of the URL we use are the scheme ex. (https) and domain ex. (www.gvsu.edu). This way it bypasses the dynamic part of the URL.

Explanation of Code

We use the urlparse library in python to parse the url into pieces we will need later. When a URL is passed to the code, the remove_dynamic_part function will parse the scheme and domain out of the entire URL. It is then sent to the is_whitelisted function to compare this smaller section of the URL to all the URLs in the whitelist. If a match is found, it is accepted.

Demo

github link: https://github.com/MatteoMCiavaglia/cis457-project2.git

Ideas for Future Work

Implementation of this solution into a website to further test it would be the next step. Then, taking data from users and comparing the values of our solution to the current solution to see its effectiveness. In the end, we would hope our solution improved the effectiveness of the whitelist solution for clickjacking. If the data shows otherwise, we would start over by coming up with a solution related to blacklisting instead and complete all the steps we did for our original solution.