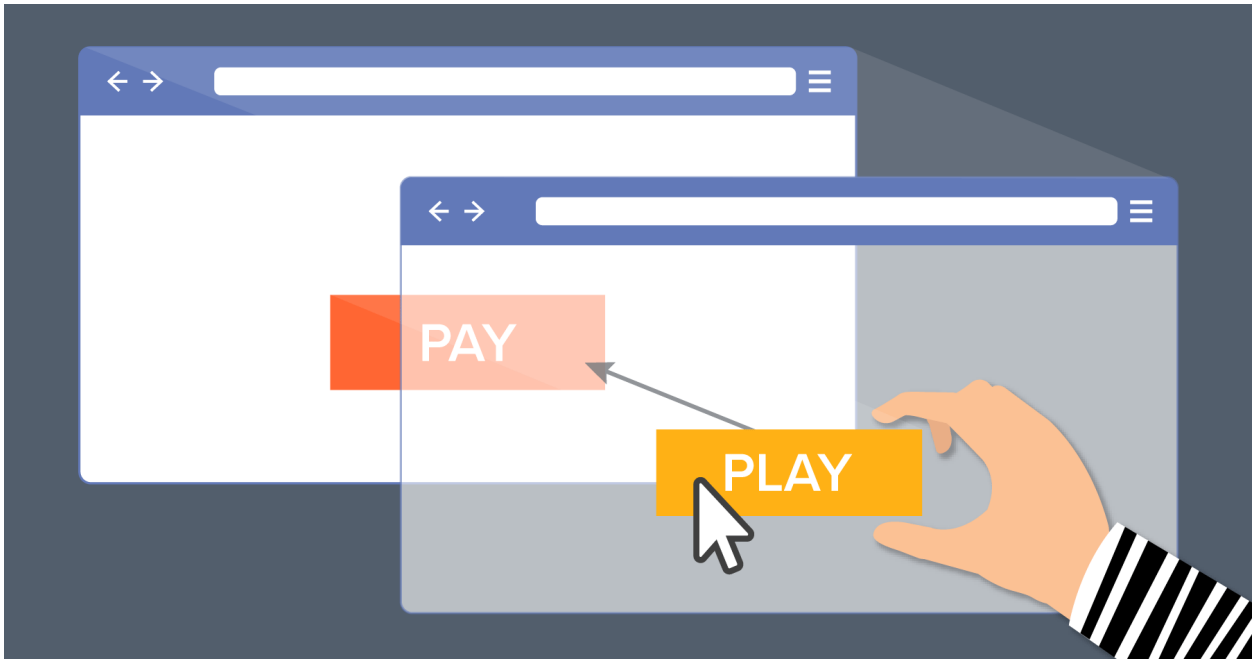
Click Jacking

What is clickjacking?

Clickjacking is an interface-based attack in which a user is tricked into clicking on actionable content on a hidden website by clicking on some other content in a decoy website. A web user accesses a decoy website (perhaps this is a link provided by an email) and clicks on a button to win a prize. Unknowingly, they have been deceived by an attacker into pressing an alternative hidden button and this results in the payment of an account on another site. This is an example of a clickjacking attack. The technique depends upon the incorporation of an invisible, actionable web page (or multiple pages) containing a button or hidden link, say, within an iframe. The iframe is overlaid on top of the user's anticipated decoy web page content. This attack differs from a CSRF attack in that the user is required to perform an action such as a button click whereas a CSRF attack depends upon forging an entire request without the user's knowledge or input.

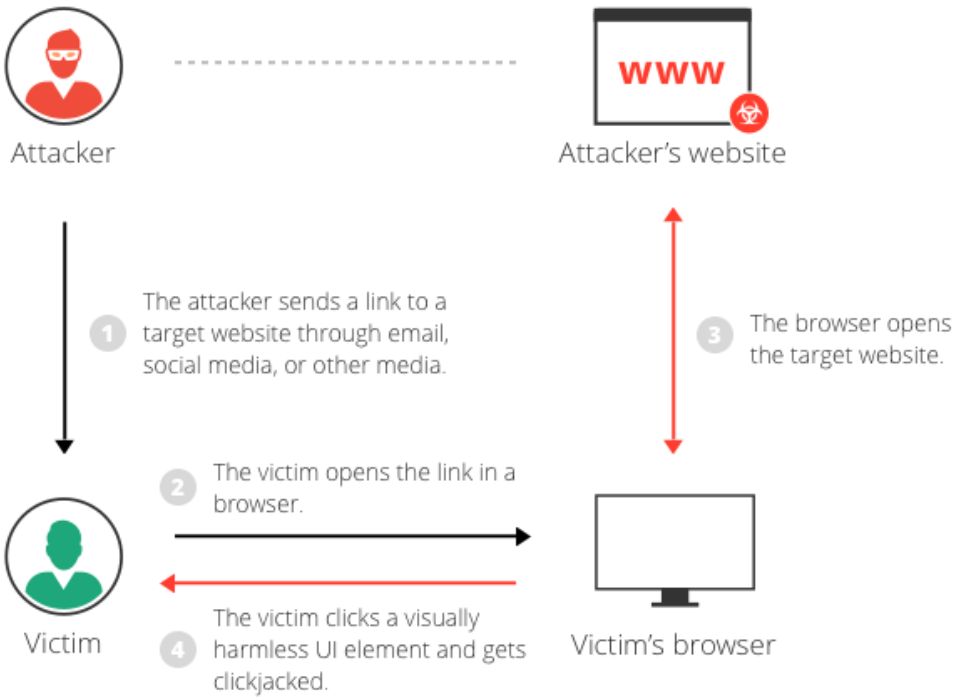


Protection against CSRF attacks is often provided by the use of a CSRF token: a session-specific, single-use number or nonce. Clickjacking attacks are not mitigated by the CSRF token as a target

session is established with content loaded from an authentic website and with all requests happening on-domain. CSRF tokens are placed into requests and passed to the server as part of a normally behaved session. The difference compared to a normal user session is that the process occurs within a hidden iframe

Clickjacking attack example

1. The attacker creates an attractive page which promises to give the user a free trip to Tahiti.
2. In the background the attacker checks if the user is logged into his banking site and if so, loads the screen that enables transfer of funds, using query parameters to insert the attacker's bank details into the form.
3. The bank transfer page is displayed in an invisible iframe above the free gift page, with the "Confirm Transfer" button exactly aligned over the "Receive Gift" button visible to the user.
4. The user visits the page and clicks the "Book My Free Trip" button.
5. In reality the user is clicking on the invisible iframe, and has clicked the "Confirm Transfer" button. Funds are transferred to the attacker.
6. The user is redirected to a page with information about the free gift (not knowing what happened in the background).



This example illustrates that, in a clickjacking attack, the malicious action (on the bank website,

in this case) cannot be traced back to the attacker because the user performed it while being legitimately signed into their own account.

How to prevent clickjacking attacks

We have discussed a commonly encountered browser-side prevention mechanism, namely frame busting scripts. However, we have seen that it is often straightforward for an attacker to circumvent these protections. Consequently, server driven protocols have been devised that constrain browser iframe usage and mitigate against clickjacking.

Clickjacking is a browser-side behavior and its success or otherwise depends upon browser functionality and conformity to prevailing web standards and best practice. Server-side protection against clickjacking is provided by defining and communicating constraints over the use of components such as iframes. However, implementation of protection depends upon browser compliance and enforcement of these constraints. Two mechanisms for server-side clickjacking protection are X-Frame-Options

References

- What is Click Jacking?
- Click Jacking attack Example?
- How to prevent Click Jacking attacks?
- https://portswigger.net/web-security/clickjacking
- https://www.imperva.com/learn/application-security/clickjacking/