January 02, 2025

DoubleClickjacking: New Attack Puts Website Accounts at Risk

Overview: A new form of a well-known cyberattack technique called **DoubleClickjacking** has been discovered, raising significant concerns about online security. This attack bypasses traditional clickjacking defenses and targets websites, exposing users to potential account takeovers. Unlike previous methods, DoubleClickjacking exploits a two-click sequence to manipulate the timing of user interactions, granting attackers unauthorized access to sensitive accounts.

CTI Analysis: DoubleClickjacking leverages a unique timing and event-order exploit to manipulate user interactions across multiple windows without utilizing popunder techniques. Attackers initiate the attack by opening a seemingly harmless window that prompts the user to "double-click". During this action, the attacker's site swaps the content of the parent window to a sensitive page, such as an OAuth authorization dialog, just before the second click occurs. This subtle timing manipulation of the mousedown and click events allows the attacker to hijack the second click, which unknowingly authorizes malicious actions.

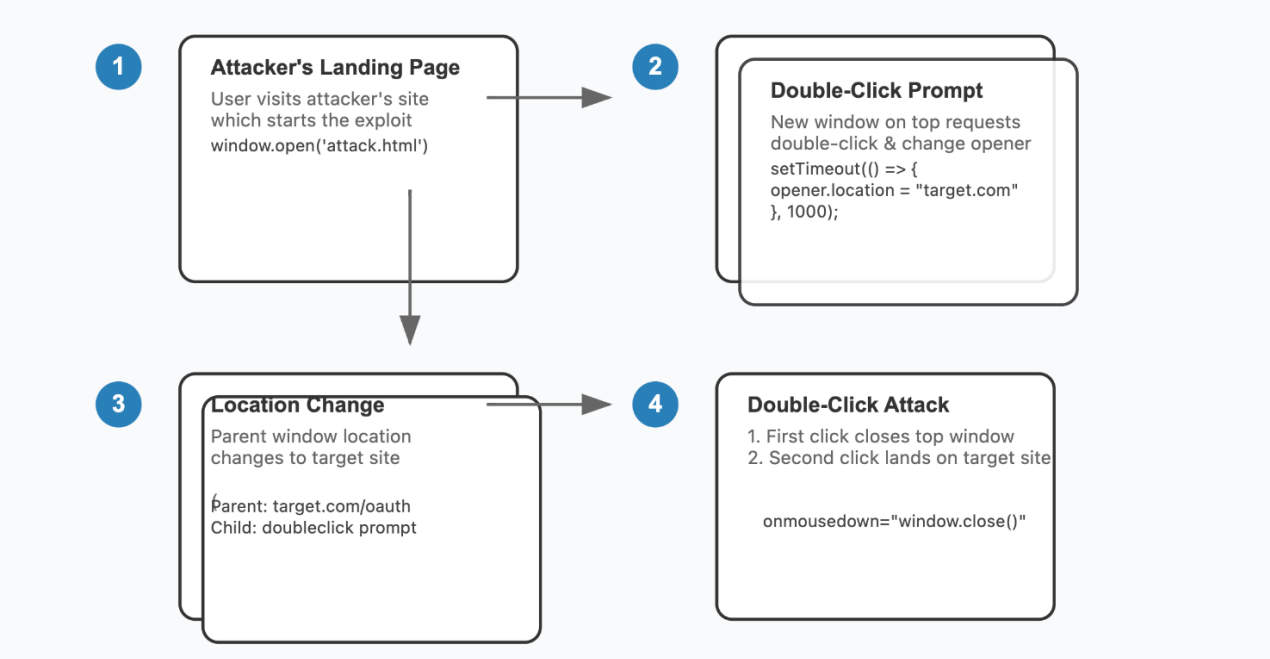

Figure 1: DoubleClickjacking Attack Flow

Impact Analysis: The attack can be exploited to gain unauthorized access to user accounts by bypassing OAuth security mechanisms. With this method, attackers can trick users into granting applications with extensive permissions, potentially leading to account takeovers. Even if detected and revoked, the attacker can perform malicious actions immediately after authorization. Moreover, DoubleClickjacking can also be used for one-click account changes, such as disabling security settings, transferring funds, or deleting accounts, putting users and platforms at significant risk. This attack method has significant real-world consequences, especially for platforms using OAuth for user authentication. Websites like Salesforce, Slack, and Shopify, which rely on OAuth, are particularly vulnerable. Additionally, existing protections such as X-Frame-Options headers, Content Security Policies (CSP), and SameSite cookies fail to prevent DoubleClickjacking due to the unique nature of the exploit. The exploit only requires a simple double-click, making it highly deceptive and easy to fall victim to.

Mitigation

- **Client-Side Protections**: Developers can implement JavaScript solutions that disable sensitive buttons by default until intentional user interaction is detected. This can be done by using event listeners like mouse movement or keypress to ensure user activity before enabling buttons.
- **Long-Term Browser Solutions**: Browsers should introduce new standards, such as a Double-Click-Protection HTTP header and improved CSP directives, to address the root cause of the vulnerability and prevent rapid context-switching during double-click actions.
- **Best Practices for Developers**: Add protective scripts to sensitive pages, such as those handling OAuth permissions or payment confirmations, and enforce stricter controls on embedded windows or opener-based navigation to prevent unauthorized access.

Conclusion: DoubleClickjacking introduces a new, sophisticated method of exploiting user interactions to bypass existing web security mechanisms. It highlights the need for ongoing vigilance in web development and browser security. Developers must implement immediate

client-side protections, and browser vendors should consider long-term solutions to address this growing threat. Staying proactive against innovative attack methods like DoubleClickjacking is crucial for safeguarding user data and maintaining trust in online platforms.

Source:

https://www.paulosyibelo.com/2024/12/doubleclickjacking-what.html

https://thehackernews.com/2025/01/new-doubleclickjacking-exploit-bypasses.html

https://cybersecuritynews.com/doubleclickjacking/