**Clickjacking Detection and Prevention Techniques**

**Web Application Security**

**CDAC, Noida**

**CYBER GYAN VIRTUAL INTERNSHIP PROGRAM**

**Submitted By:**

Abhishek Ranjan

Kajal Kashyap, (May-June) 2025

**BONAFIDE CERTIFICATE**

This is to certify that this project report entitled “Clickjacking Detection and Prevention Techniques” submitted to CDAC Noida, is a Bonafide record of work done by Abhishek Ranjan under my supervision from 29/05/2025 to 12/06/2025.

**Declaration by Author(s)**

This is to declare that this report has been written by me. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I aver that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

Name of Author: Abhishek Ranjan

**ACKNOWLEDGEMENT**

I would like to thank CDAC Noida and the Cyber Gyan Virtual Internship Program for giving me the worthy chance to develop the project entitled "Clickjacking Detection and Prevention Techniques."

I am immensely grateful to my project mentor(s) and the entire internship coordination team for their continuous guidance, encouragement, and support during the period of this internship. Their valuable suggestions and effective feedback played a pivotal role in the successful accomplishment of this project.

I also offer my sincere gratitude to my colleagues, friends, and relatives for their constant support and encouragement throughout the process. This process greatly enhanced my knowledge and practical application of web security, and I appreciate the opportunity to implement my learning in a real-world project.

Finally, I would like to thank all the people directly or indirectly involved in the completion of this project.

**TABLE OF CONTENTS**

1. Introduction 1
   1. Problem statement 1
   2. Learning objectives 1
2. Approach 2
   1. Tools used 2
   2. Techniques used 3
   3. Infrastructure setup 3
3. Implementation 3
   1. Indicators of compromise 6
4. Conclusion 7
   1. Recommendations 7
5. References 8

**Clickjacking Detection and Prevention Techniques**

**INTRODUCTION:**

Clickjacking also known as ‘UI redressing’ is a common cybersecurity attack. In this attack, the end user is given a webpage offering some rewards like winning a lottery or giveaway, etc. and says to click the button to claim the reward.  
But in reality, an invisible page is loaded over the UI and fools the user on click the invisible button instead of the button the user thinks he/she is clicking.

**PROBLEM STATEMENT:**

Clickjacking is an attack on the web that misleads a user into clicking on something other than what he/she thinks, possibly revealing sensitive user activities or breaching system security. Since the popularity of dynamic and interactive web applications is increasing, the demand for strong security practices against such attacks becomes imperative. This project proposes to build and analyse detection and prevention strategies to respond to clickjacking vulnerabilities.

**LEARNING OBJECTIVES:**

* Understanding the nature, dangers and consequences of the clickjacking attack.
* How clickjacking works and different methods used to trick users.
* Study real-world examples and impacts of clickjacking in web applications and security.
* How to detect the vulnerable components of the webpage, analysing the user interface, leading to develop scripts and tools to detect potential clickjacking attack.
* Exploration, Evaluation and Implementation of both server-side and client-side protection techniques.
* Learn how extensions work and developing scripts for creating an extension to detect potential attack.
* Gain hands-on experience in development of secure web applications and environments.
* Best practices to secure web applications from clickjacking attacks.

**APPROACH:**

This project involves the creation of a type of clickjack attack, likejacking. It also involves the use and testing of both server-side and client-side techniques to detect and prevent such attacks. Additionally, a browser extension is also developed to review the webpage of potential clickjack attacks.

**TOOLS USED:**

* HTML and CSS – To simulate the clickjack attack.
* JAVASCRIPT – For clickjack detection and prevention.
* JSON – For developing extension’s script.
* CHROME (Browser) – To evaluate, test and analyse the attack and its prevention mechanisms.
* VISUAL STUDIO CODE – For writing source codes and scripts.
* LIVE SERVER (VS code extension) – To deploy and simulate the clickjack attack on local host server.
* GIT and GITHUB – To keep track of changes to code and share project online.
* CHROME DEVTOOLS – For inspection, debugging, monitoring of the clickjack attack.

**TECHNIQUES USED:**

SERVER-SIDE TECHNIQUES –

* Content Security Policy (CSP): Define trusted sources for resources like scripts and frames.
* X-Frame-Options header: Control show the website can be embedded in iframes.
* Frame busting JavaScript: Use JavaScript to detect and break out of malicious iframes.

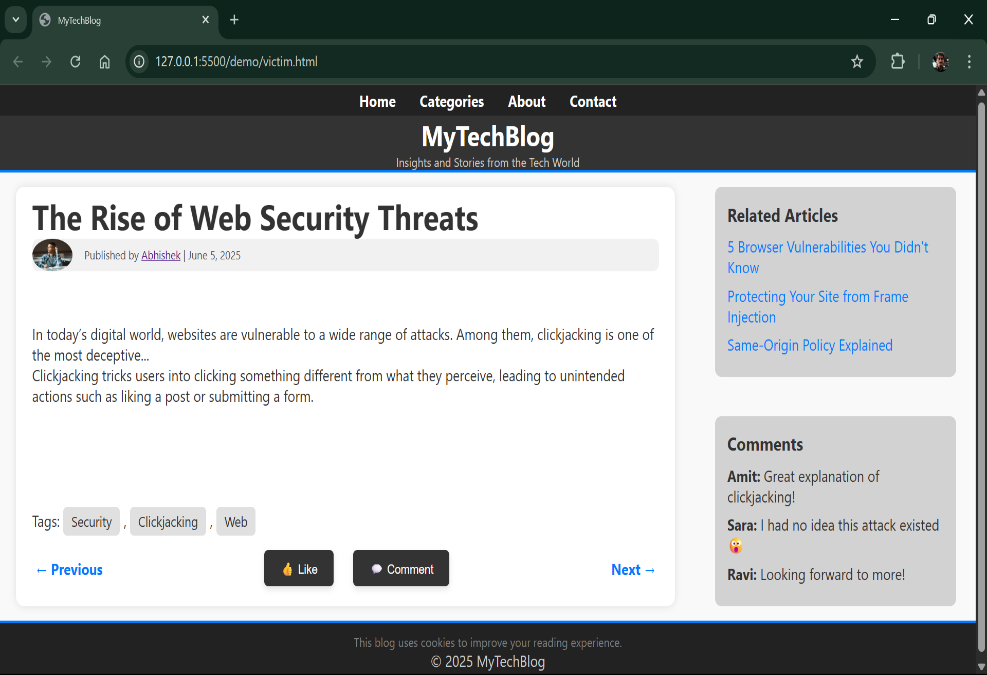
CLIENT-SIDE TECHNIQUES –

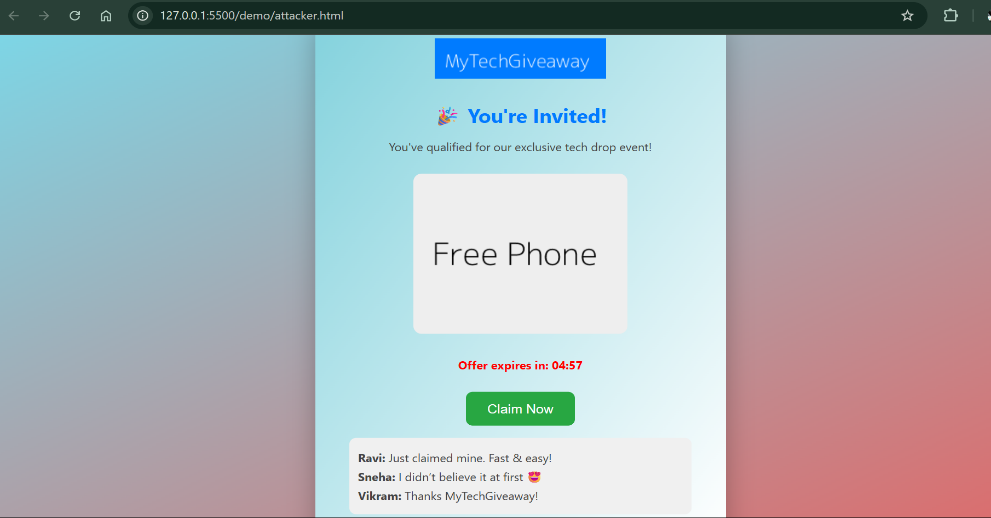
* Browser extensions: Analyse web pages for clickjacking vulnerabilities.
* Intersection Observer API: Utilize JavaScript API to detect the visibility of clickable elements and prevent clicks on hidden ones.

**INFRASTRUCTURE SETUP:**

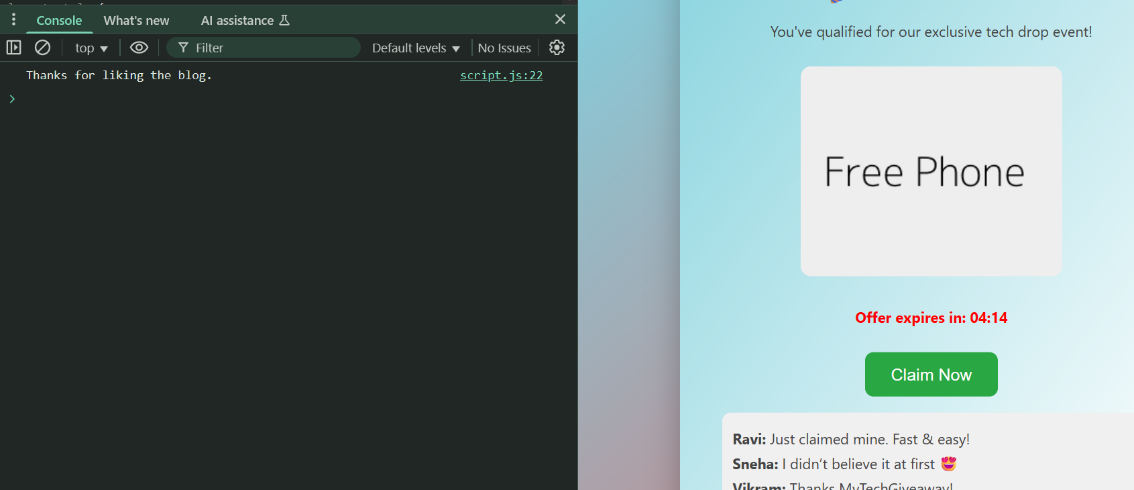
* **Local Server:** Live server for hosting webpages.
* **Extension:** Self designed extension linking in browser.
* **Testing Environment:** Chrome DevTools.

**IMPLEMENTATION:**

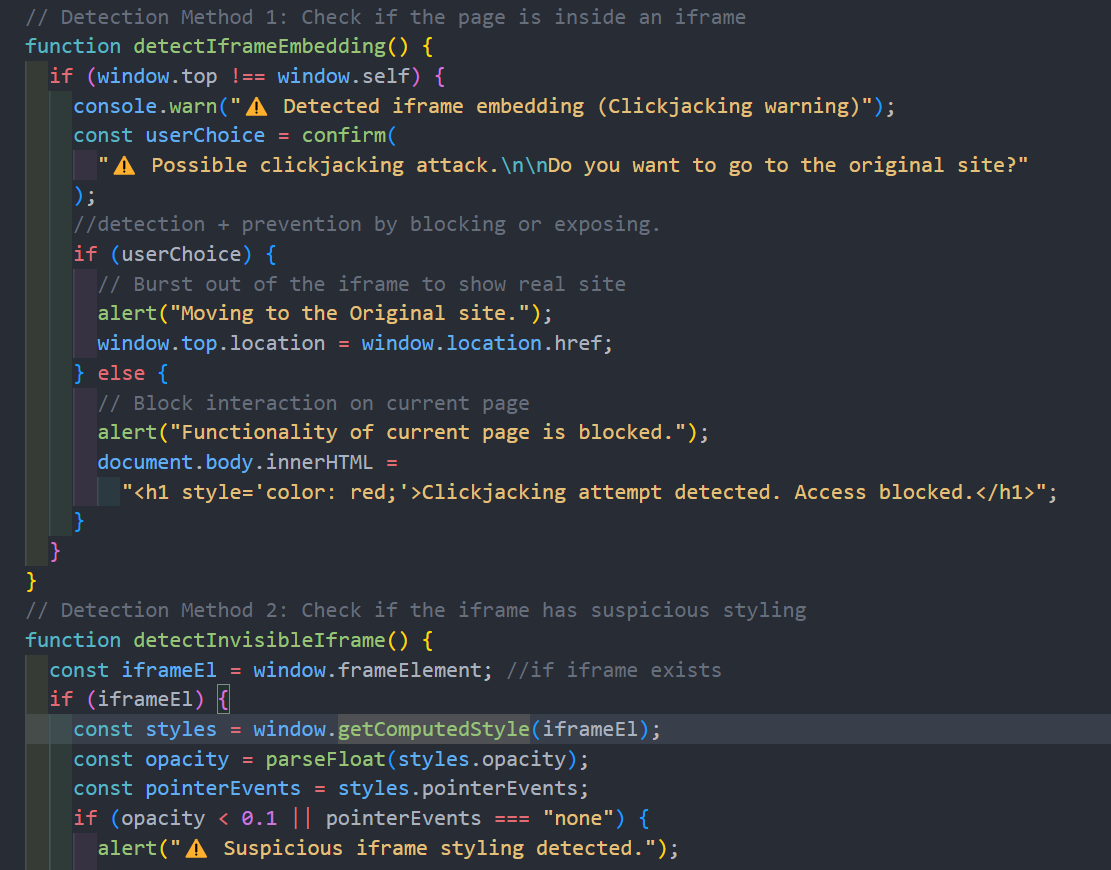
1. Setup Victim Webpage
   * Designing a webpage ‘victim.html’ that will get affected by the attacker.
   * This must contain basic components like button for clickjack attack.
   * This page is initially vulnerable to iframe embedding. 
2. Setup Attacker Webpage
   * This webpage (attacker.html) contains iframe embedding of victim.html
   * Designed to lure users into clicking disguised buttons.



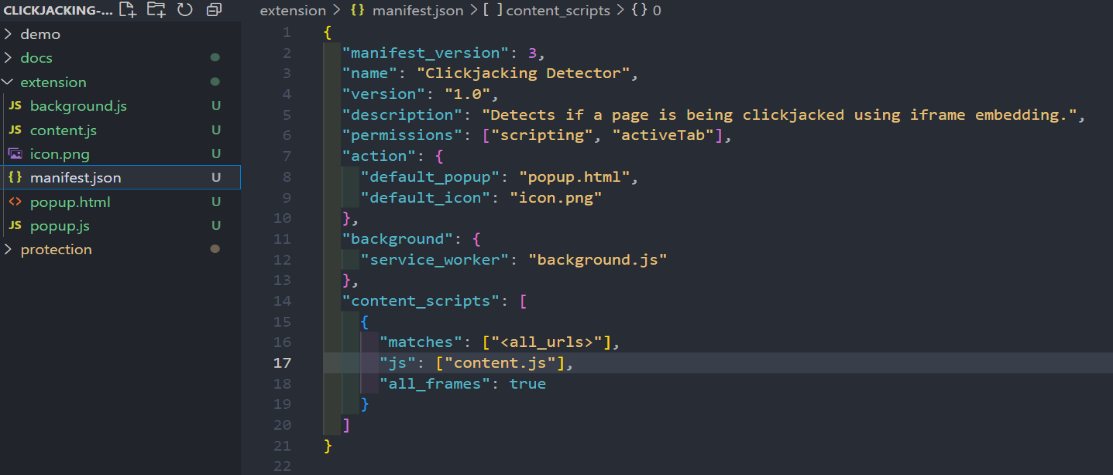
1. Simulating Clickjacking attack (Likejacking)
   * Ensuring that every webpage is linked to or embedded on each other perfectly.
   * Clickjacking is performed successfully.



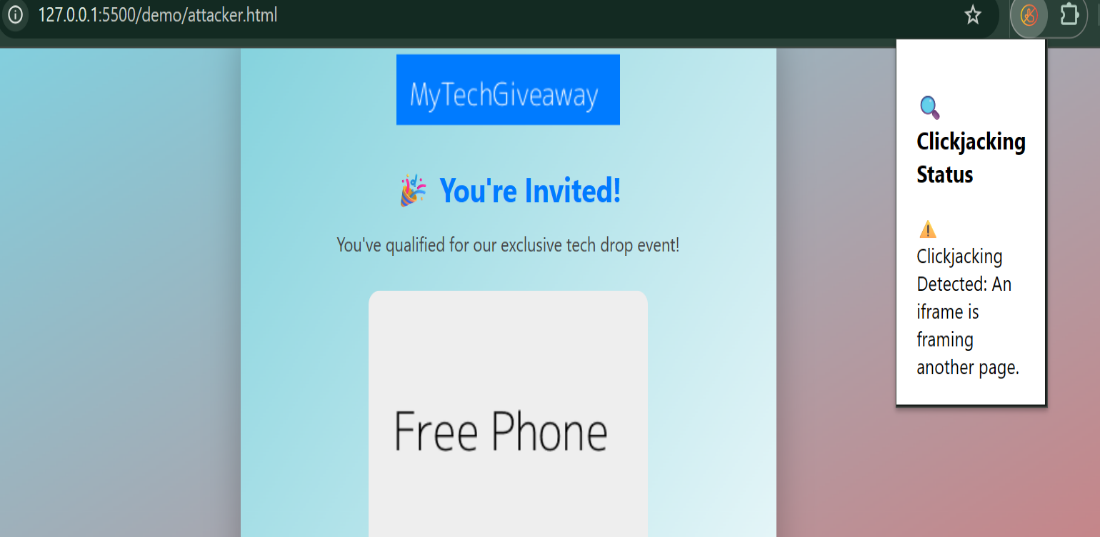
1. Applying Server-side protection
   * Set X-frame-options: Deny in headers
   * Applying CSP directives
   * Using JavaScript frame busting code



1. Applying Client-side protection
   * Creating a basic browser extension prototype that warns user of the embedded iframes.



1. Testing Clickjacking protection mechanisms
   * Simulating attack with and without protection and test if everything works fine.
   * Using browser dev tools to test iframe and the extension and their behaviour.



**Indicators of Compromise (IoCs):**

* Presence of ‘iframes’ in webpages – iframes are one of the major components used in clickjacking attack.
* Transparent components – Components like buttons or iframes are hidden from user to trick them.
* Unusual clicks – Click triggering actions without any visible UI feedback
* Embedded frames – Embedding of webpages in iframes from external webpages and unauthorised domains without the knowledge of users.
* Webpage redirection – Sudden page redirection to random webpages upon clicks or interactions.

**CONCLUSION:**

The project successfully illustrated the vulnerability of clickjacking in modern web applications and explored various methods for its detection and prevention. Server-side techniques such as CSP and X-Frame-Options were effective against iframe-based attacks. Frame busting JavaScript provides an additional layer of protection. On the client side, Intersection Observer APIs and Browser Extensions were found to be effective methods for detecting malicious activity.

**RECOMMENDATIONS:**

* Implement security measures – Always specify proper security headers on production environments.
* Adopt frame busting techniques – Using JavaScript frame-busting scripts are useful when headers cannot be applied.
* Security Audits – Regularly checking and scanning the web applications for iframe vulnerabilities.
* Education – Educating developers and programmers team about secure coding practices and stay updated on the latest clickjacking techniques and countermeasures.
* Browser support – Encourage browser’s anti-clickjacking features and extensions.

**LIST OF REFERENCES:**

* PortSwigger - <https://portswigger.net/web-security/clickjacking>
* Auth0 - <https://auth0.com/blog/preventing-clickjacking-attacks/>
* MDN web docs - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CSP>
* GFG - <https://www.geeksforgeeks.org/http-headers-x-frame-options/>
* OWASP - <https://owasp.org/www-community/attacks/Clickjacking>
* Chrome for Developers - <https://developer.chrome.com/docs/extensions/get-started>
* Stack Overflow - <https://stackoverflow.com/questions/958997/frame-buster-buster-buster-code-needed>