**A PARADIGM SHIFT IN XSS-DOM MITIGATION VIA**

**DCSP AND PROXY ORCHESTRATION**

**REPORT**

**IT5712 - PROJECT I**

***Submitted by***

**Krishnaa S (2020506045)**

**Jawahar A S (2020506035)**

**Thamizharasi M (2020506102)**

***Guided by***

**Dr. P. Kola Sujatha**

## BACHELOR OF TECHNOLOGY

***in***

**INFORMATION TECHNOLOGY**

**MADRAS INSTITUTE OF TECHNOLOGY**

**ANNA UNIVERSITY : CHENNAI 600044**

**25th October 2023**

**INTRODUCTION:**

Cross-Site Scripting (XSS) and Document Object Model (DOM) attacks are critical security concerns for web applications. XSS vulnerabilities allow attackers to inject malicious scripts into web pages viewed by other users, potentially compromising their data or session information. These attacks exploit the trust a user has in a website, enabling unauthorized access or data theft. DOM-based XSS attacks specifically manipulate a webpage's Document Object Model, which represents the structure of a web page in a hierarchical manner. In DOM-based XSS, malicious scripts manipulate the DOM of a web page dynamically, often targeting client-side scripts and affecting the behavior of web applications. Detecting and mitigating DOM-based XSS attacks is essential for safeguarding user data, maintaining application integrity, and preventing unauthorized access to sensitive information. Security measures involve input validation, output encoding, and monitoring for unusual script behavior to thwart these attacks and ensure a secure web environment.

Our focus is on using pattern matching techniques to detect XSS-DOM vulnerabilities, adding an extra layer of security to web applications. This proactive approach helps identify potential attacks early, enhancing overall security. We also explore preventive strategies, such as input validation and Content Security Policy (CSP), in detail. These strategies are effective in reducing the risks associated with XSS-DOM attacks, making web environments more secure against potential breaches.

**OBJECTIVES:**

**Understanding Vulnerabilities:** Gain insight into the nature and mechanics of Cross-Site Scripting (XSS) and DOM-based attacks.

**Detection Techniques:** Investigate methods (Pattern matching) for identifying XSS-DOM vulnerabilities, enabling proactive defense against potential attacks.

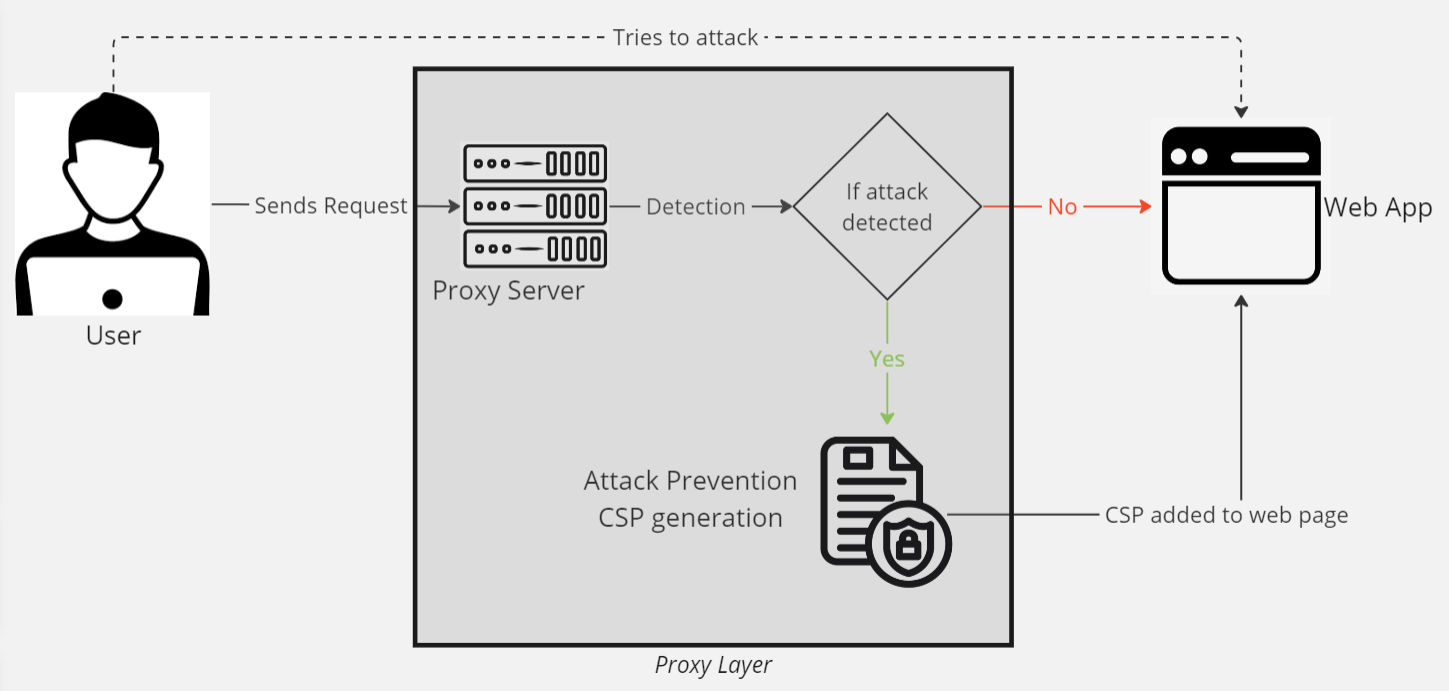
**Preventive Strategies:** Explore strategies such as input validation and Content Security Policy (CSP) to mitigate the risks of XSS-DOM attacks.

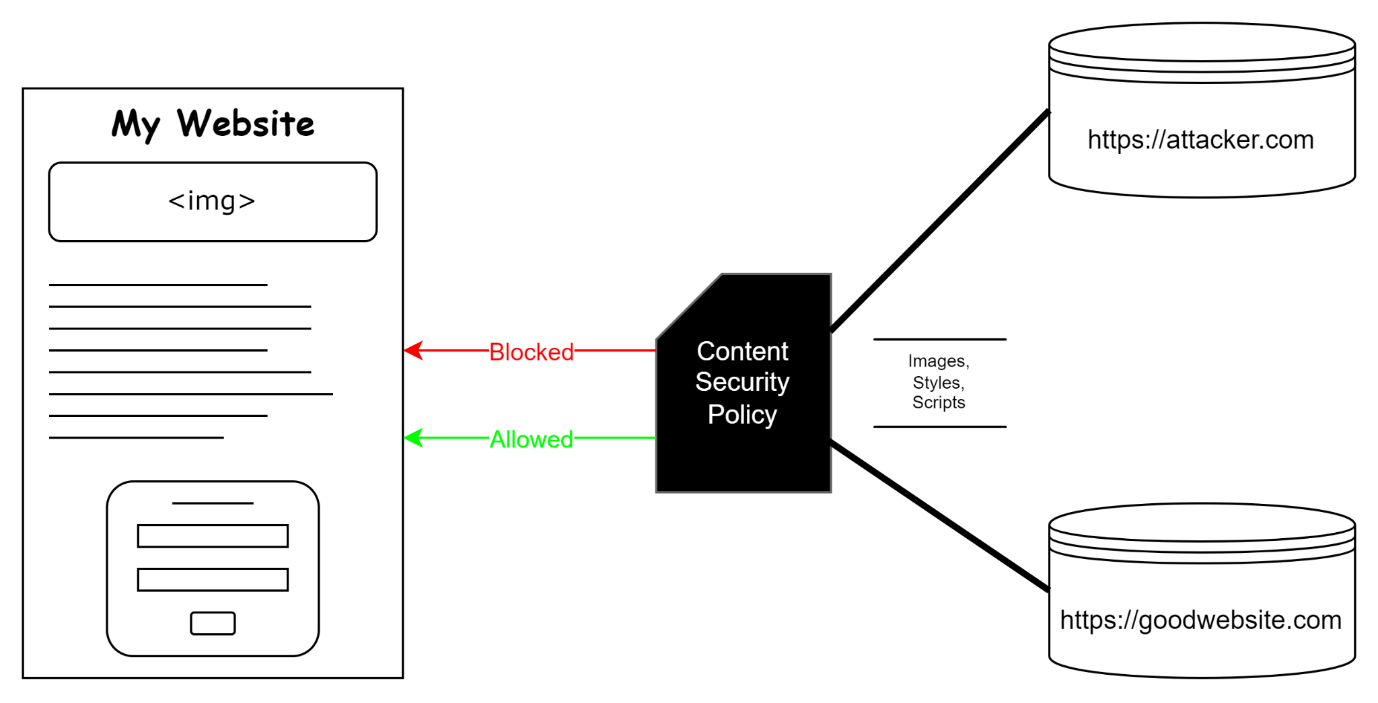
**Real-world Insights:** Analyze real-world cases of notable attacks to grasp the potential consequences and implications of these vulnerabilities.

**PROBLEM STATEMENT:**

Cross-Site Scripting (XSS) vulnerabilities persist in web applications, allowing malicious scripts to compromise user data and privacy. Thus, lack of adaptive security measures leaves dynamic pages exposed to XSS attacks, necessitating a novel solution.

**DIAGRAM:**





**TECH STACK:**

* Python flask
* JavaScript.
* Sqlite

**SYSTEM REQUIREMENTS:**

* Windows 10 and above
* Vs code
* Chrome browser.

**BENEFITS:**

**Uncompromised User Confidence**: Mitigating XSS-DOM vulnerabilities fosters user trust in application's security and reliability.

**Preserved Data Integrity**: By neutralizing these threats, you maintain the integrity of user data, preventing unauthorized access and tampering.

**Enhanced Brand Reputation**: A secure application reflects positively on a brand, positioning the owner as a responsible and security-conscious provider.

**Reduced Legal and Financial Risk**: Preventing attacks helps you avoid potential legal liabilities and financial losses that can arise from data breaches or compromised user information.

**BASE PAPER:**

G. Xu et al., "JSCSP: A Novel Policy-Based XSS Defense Mechanism for Browsers," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 2, pp. 862-878, 1 March-April 2022, doi: 10.1109/TDSC.2020.3009472.

**REFERENCES:**

1. K. Ali, A. Abdel-Hamid and M. Kholief, "Prevention Of DOM Based XSS Attacks Using A White List Framework," 2014 24th International Conference on Computer Theory and Applications (ICCTA), Alexandria, Egypt, 2014, pp. 68-75, doi: 10.1109/ICCTA35431.2014.9521633.
2. S. K. Mahmoud, M. Alfonse, M. I. Roushdy and A. -B. M. Salem, "A comparative analysis of Cross Site Scripting (XSS) detecting and defensive techniques," 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, 2017, pp. 36-42, doi: 10.1109/INTELCIS.2017.8260024.
3. J. C. Pazos, J. -S. Légaré and I. Beschastnikh, "XSnare: Application-specific client-side cross-site scripting protection," 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Honolulu, HI, USA, 2021, pp. 154-165, doi: 10.1109/SANER50967.2021.00023.
4. Z. Jingyu, H. Hongchao, H. Shumin and L. Huanruo, "A XSS Attack Detection Method Based on Subsequence Matching Algorithm," 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), Guangzhou, China, 2021, pp. 83-86, doi: 10.1109/AIID51893.2021.9456515.