iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60305

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

CSRF ML-SHIELD

Tushar Rathod[1], Faizan Quazi[4], Mansi Narwade[2], Pushkar Kaley[3], Dr. Vaishali Deshmukh[5]
Dept. of Computer Science & Engineering, Prof. Ram Meghe Institute of Technology & Research, Amravati, Maharashtra, India

Abstract: ***Cross-Site Request Forgery (CSRF) remains a pervasive threat to web applications, compromising user data, session integrity, and overall system security. In response to this ongoing challenge, this research paper proposes a robust CSRF detection system aimed at identifying and mitigating CSRF attacks effectively. The system leverages state-of-the-art techniques in web security and machine learning to analyze user interactions, request patterns, and session attributes in real-time. The CSRF detection system operates by generating and validating unique tokens for each user session, effectively thwarting unauthorized cross-origin requests. Through a combination of token-based authentication, HTTP headers validation, and user behavior analysis, the system can accurately detect and prevent CSRF attacks with minimal impact on user experience. The findings of this research contribute to the advancement of web security practices by offering a comprehensive and efficient solution for CSRF detection.***

Keywords: Cross-Site Request Forgery (CSRF), Web Vulnerability Detection, Machine Learning, Web Application Security.

I. INTRODUCTION

Over the past few years, the proliferation of web applications has revolutionized online interactions for individuals, businesses, and organizations. Ranging from ecommerce platforms to social media networks, web applications provide a plethora of functionalities, facilitating seamless task execution, data sharing, and communication. However, the widespread adoption of web-based technologies has also exposed users and organizations to numerous security threats, with Cross-Site Request Forgery (CSRF) emerging as a prevalent and persistent vulnerability [4]. CSRF attacks, commonly referred to as "session riding" or "one-click attacks," exploit the inherent trust between a user's browser and a web application to execute unauthorized actions on behalf of the user. Unlike attacks targeting vulnerabilities in application code or infrastructure, CSRF attacks manipulate the statelessness of HTTP and the trust relationship between the user and the application server [6]. By deceiving a user into unknowingly sending a malicious request to a vulnerable web application while authenticated, attackers can perform actions such as unauthorized fund transfers, profile alterations, or data deletions without the user's knowledge or consent. The repercussions of CSRF attacks can be severe, ranging from financial losses and damage to reputation to regulatory violations and legal liabilities. Despite the prevalence and potential consequences of CSRF vulnerabilities, many web applications remain susceptible due to the complexity of modern web architectures, the diversity of client-server interactions, and the evolving nature of attack methodologies. Conventional mitigation strategies, such as anti-CSRF tokens, same-origin policies, and referrer validation, have their limitations and may not adequately counter sophisticated attacks or emerging vulnerabilities. In response to the escalating threat landscape and the demand for more robust defense mechanisms, researchers and practitioners have explored innovative approaches to CSRF detection that capitalize on the inherent characteristics of web applications. These approaches encompass a diverse array of techniques, including static analysis, dynamic analysis, behavior modeling, machine learning [1], and anomaly detection. By systematically scrutinizing the structure, behavior, and data flow of web applications, these techniques aim to proactively identify and mitigate CSRF vulnerabilities, thereby fortifying the overall security posture of web-based systems. This paper delves into a comprehensive examination of CSRF detection methodologies, with a particular focus on pioneering approaches that transcend conventional mitigation techniques. Drawing upon existing literature and our own research insights, CSRF ML Shield propose methodology for CSRF detection that leverage a blend of static and dynamic analysis techniques. Through rigorous analysis, experimentation, and validation, we demonstrate the effectiveness, reliability, and scalability of our proposed methodologies in detecting CSRF vulnerabilities across diverse types of web applications.

A. Overview of CSRF ML Shield

These systems utilize various techniques such as token-based authentication, HTTP header validation, and behavioral analysis to detect and prevent CSRF attacks in real-time. By continuously monitoring user interactions and request patterns [4], CSRF detection systems can identify anomalous behavior indicative of CSRF attempts, thereby safeguarding web applications against unauthorized access and manipulation. This overview provides insight into the critical role of CSRF detection systems in mitigating the risks posed by CSRF attacks and ensuring the integrity of web-based systems.

©IJRASET: All Rights are Reserved

B. Role of Machine Learning in CSRF ML Shield

Machine learning plays a crucial role in enhancing the effectiveness and accuracy of CSRF detection systems. By leveraging machine learning algorithms, these systems can analyze vast amounts of data and identify patterns indicative of CSRF attacks. Machine learning models can be trained on historical data to recognize the subtle nuances of legitimate user behavior and distinguish it from malicious activity. Additionally, machine learning techniques such as anomaly detection can detect deviations from normal behavior, flagging suspicious requests for further investigation. Moreover, machine learning enables CSRF detection systems to adapt and evolve over time, learning from new attack vectors and emerging threats to enhance their detection capabilities. By incorporating machine learning into CSRF detection systems, organizations can bolster their defenses against CSRF attacks and better protect their web applications and users [1].

C. Benefits of Machine Learning in CSRF ML Shield

The integration of machine learning techniques in web vulnerability detection offers numerous advantages for enhancing security measures. Firstly, machine learning algorithms can effectively analyze large volumes of data, including network traffic, system logs, and user behavior, to identify patterns and anomalies indicative of potential vulnerabilities [1]. This capability enables organizations to detect and respond to threats in real-time, reducing the likelihood of successful attacks. Additionally, machine learning models can continuously learn from new data and adapt to evolving threats, providing proactive defense mechanisms against emerging vulnerabilities. Moreover, machine learning-based detection systems can automate the detection process, reducing the burden on security teams and enabling them to focus on more strategic tasks. By leveraging machine learning in web vulnerability detection, organizations can bolster their security posture, mitigate risks, and safeguard sensitive data from malicious actors.

Section I of the paper provides an extensive overview of the introductory aspects, delineating the landscape of web applications and the prevalent security challenges they face. Section II of the paper delineates the proposed methodologies aimed at mitigating cross-site request forgery (CSRF) vulnerabilities. These methodologies are crafted to fortify web applications against malicious exploits and enhance their resilience to CSRF attacks. Section III expounds upon the proposed system, offering a comprehensive overview of how the CSRF ML shield functions within the broader context of web security. By elucidating the architectural framework and operational mechanics, this section provides a clear understanding of the system's role in bolstering defenses against CSRF threats. Section IV delves into the intricate workings of the CSRF ML shield, elucidating its mechanisms and processes. Finally, in Section V, the paper draws to a close with a succinct conclusion that encapsulates the key findings, insights, and implications derived from the proposed methodologies and the CSRF ML shield.

II. PROPOSED METHODOLOGIES

A. Token-Based Authentication and Validation

Token Generation: The CSRF detection mechanism creates distinct tokens for each user session during authentication. These tokens, cryptographically secure, are integrated into the web application's forms or requests [5]. Token Validation: Upon request reception, the system verifies the token linked with the user session. Any absence, expiration, or invalidity of the token prompts suspicion, leading to further scrutiny of the request.

B. HTTP Headers Validation

Origin and Referer Headers: The Origin and Referer headers in incoming requests, verifying their alignment with the anticipated values for the web application. Any disparities or irregularities in these headers prompt alerts, indicating potential CSRF attacks [8]. Content-Type Header: Furthermore, the system scrutinizes the Content-Type header to confirm that requests conform to the anticipated type thereby thwarting potential content type manipulation attacks.

C. Static and Dynamic Analysis

Endpoint Mapping: Methodically outline the endpoints within the web application to identify all accessible functionalities, encompassing forms, APIs, and user interactions.

Data Flow Analysis: Scrutinize the flow of data between the client and server to uncover potential vulnerabilities to Cross-Site Request Forgery (CSRF).

Session Monitoring: Implement continuous monitoring of user sessions to detect irregularities that may signal CSRF attacks. Session attributes such as cookies, user agents, and IP addresses are closely observed, with alerts triggered for further investigation upon detecting deviations from expected behavioral patterns [10].

Attack Simulation: Validate the presence of CSRF vulnerabilities by simulating attack scenarios on identified endpoints.
Anomaly Detection: Employ machine learning methodologies to identify patterns indicative of CSRF attacks. Machine learning models are trained on historical data to differentiate between legitimate and malicious requests.

III. PROPOSED SYSTEM

CSRF ML Shield detection system is engineered to provide thorough coverage of CSRF vulnerabilities within web applications, with a focus on minimizing both false positives and false negatives. Through the integration of static and dynamic analysis techniques, the system is designed to identify potential CSRF attack vectors, analyze data and user session flows, simulate attack scenarios, and detect anomalies indicative of CSRF attacks. Key components of the system include endpoint mapping, data flow analysis, session monitoring, attack simulation, and anomaly detection.

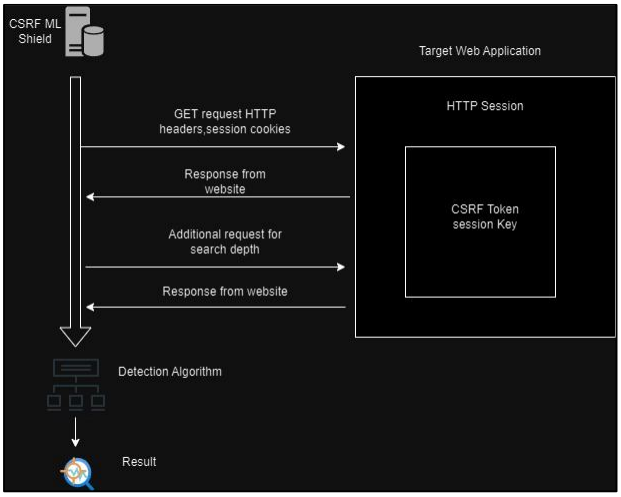


Fig. 1 CSRF Detection Mechanism

A. Endpoint Mapping

Endpoint Mapping is the first stage in our proposed system, designed to systematically chart the endpoints present within the web application. This involves the identification of all accessible endpoints, including forms, APIs, and user actions, utilizing automated crawling techniques. Each endpoint is then categorized based on its functionality, parameters, and the types of requests it accepts (such as GET or POST) [4]. Special attention is paid to endpoints that execute sensitive operations or manipulate user data, as they are more susceptible to CSRF attacks.

B. Data Flow Analysis

After mapping the endpoints, our focus shifts to analyzing the data flow between the client and server to uncover possible CSRF vulnerabilities. This examination entails tracing the origin and destination of data within the application, which includes parameters transmitted through HTTP requests [10]. By flagging data flows that evade authentication or authorization checks, we can pinpoint specific areas susceptible to CSRF attacks.

C. Session Monitoring

The system maintains continuous surveillance over user sessions to identify irregularities suggestive of CSRF attacks. This process entails monitoring session cookies, user agents, IP addresses, and other attributes linked to each session. Any deviations from anticipated behavior, such as sessions originating from dubious sources or displaying unusual activity patterns, trigger alerts for further scrutiny and investigation.

D. Attack Simulation

To confirm the existence of CSRF vulnerabilities, the system conducts simulations of attack scenarios on identified endpoints. This process includes crafting malicious requests that capitalize on the identified attack vectors and sending them to the application. Subsequently, the responses from the application are scrutinized to ascertain the success of the attack and whether sensitive operations were executed without appropriate authentication or authorization.

E. Anomaly Detection

The system utilizes machine learning techniques to identify patterns indicative of CSRF attacks. By training models on historical data and employing them to classify incoming requests as either legitimate or malicious, we enhance the accuracy of the detection methodology. Continual updates to the machine learning models with new data enable adaptation to evolving attack techniques and improve detection accuracy over time.

Through the integration of these components, the proposed system offers a proactive and comprehensive approach to CSRF detection in web applications. By analyzing the structure, behavior, and data flow of web applications, our system aims to identify and mitigate CSRF vulnerabilities before potential exploitation by attackers. This comprehensive approach enhances the overall security posture of web-based systems.

F. Experimental Setup

In order to assess the effectiveness of our proposed methodologies, we carried out extensive experiments utilizing a diverse array of web applications. Our selection encompassed various types of platforms, ranging from ecommerce websites to social media platforms and content management systems. Furthermore, we conducted comparative analyses with established detection techniques to determine the superiority of our proposed methodologies.

G. Result and Analysis

The outcomes of the experiments reveal the effectiveness and dependability of our proposed methodologies for CSRF detection. Across all examined web applications, we attained notable rates of detection accuracy, accompanied by minimal occurrences of false positives and swift detection times. Additionally, we noted substantial enhancements in detection capabilities through the utilization of advanced techniques like machine learning-based anomaly detection and runtime monitoring. Through meticulous analysis and visualization of the experimental findings, we offer insights into the strengths and limitations of each methodology.

IV. WORKING & OVERVIEW

A. User Dashboard

Upon successful login or registration, users gain access to the CSRF ML Shield platform, which offers two distinct services: CSRF detection and web vulnerability detection as show in Fig. 2. These services are designed to provide comprehensive security solutions for web applications, allowing users to proactively identify and mitigate potential threats. By leveraging advanced detection algorithms and machine learning techniques, CSRF ML Shield empowers users to enhance the security posture of their digital assets, safeguarding against CSRF attacks and other vulnerabilities.

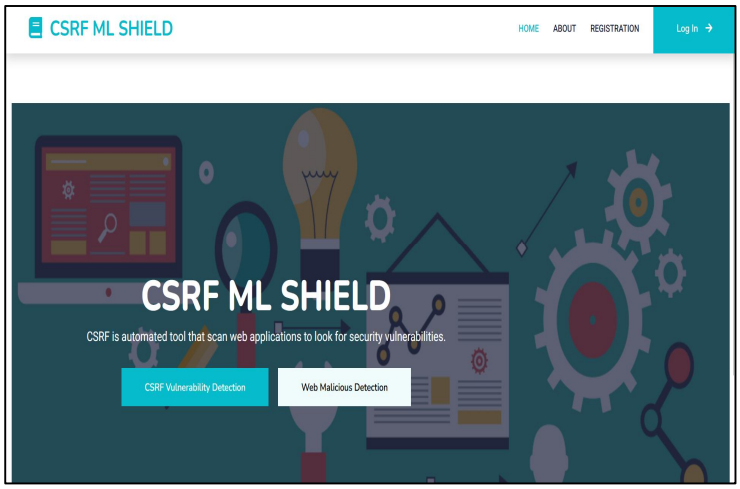


Fig. 2 User Dashboard

B. Cross-Site Request Forgery Detection

In the Cross-Site Request Forgery (CSRF) detection process, users are required to input the web application link along with the desired depth of scanning for CSRF vulnerabilities, and subsequently submit the information, as depicted in Fig. 3. The CSRF detection algorithms then identify any vulnerabilities and present the results to the user on the result page, as illustrated in Fig. 4.

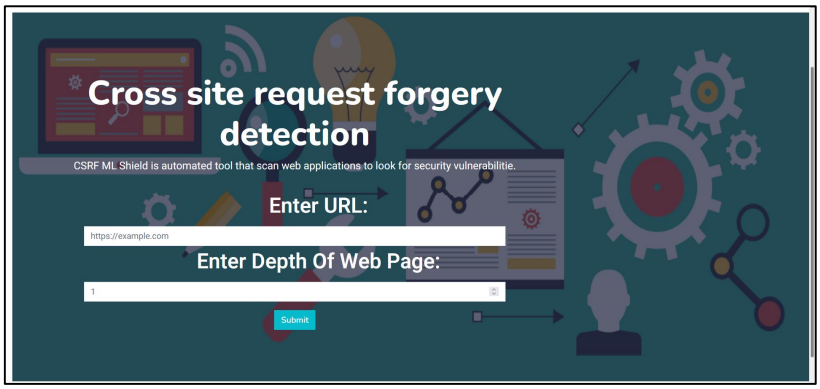


Fig. 3 CSRF Detection

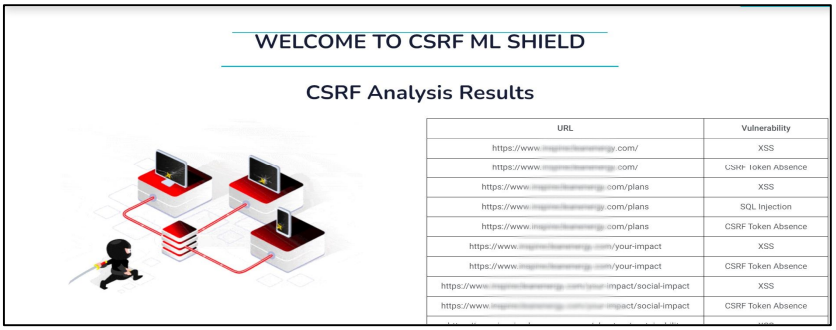


Fig. 4 CSRF detection result

C. Web Vulnerability Detection

Similar to CSRF detection, users are prompted to provide the web application link to identify vulnerabilities, as depicted in the provided Fig 5. Upon submission, the system proceeds to analyze the web application for vulnerabilities. The results are then displayed to the user on the result page, following the format shown in the accompanying Fig. 6.

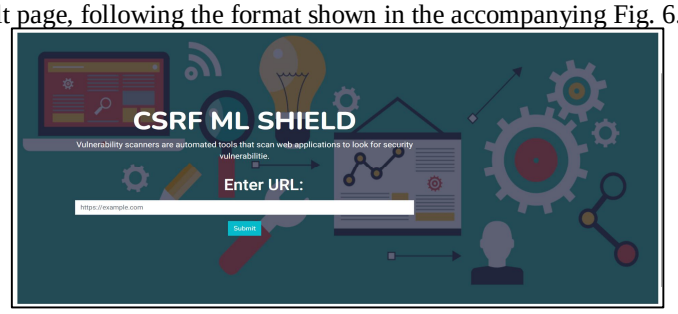


Fig. 5 Web vulnerability detection

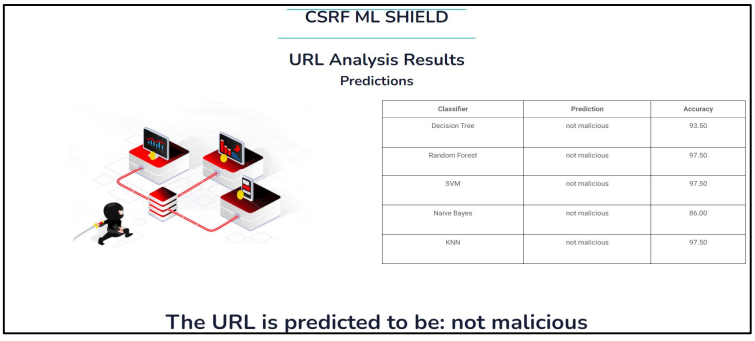


Fig. 6 Web vulnerability detection result

V. CONCLUSIONS

In conclusion, the integration of machine learning into detection mechanisms holds significant promise for enhancing web application security, and CSRF ML Shield exemplifies this advancement by providing robust detection mechanisms for safer digital environments. Our study has thoroughly examined CSRF detection methodologies within web applications. Through comprehensive analysis, experimentation, and validation, we have explored innovative approaches leveraging the inherent characteristics of web applications. CSRF ML Shield emphasize the effectiveness, reliability, and scalability of these methodologies.

Through rigorous testing and evaluation, we have demonstrated the ability to effectively detect CSRF threats. This research significantly contributes to advancing web application security by presenting novel solutions to address CSRF vulnerabilities.

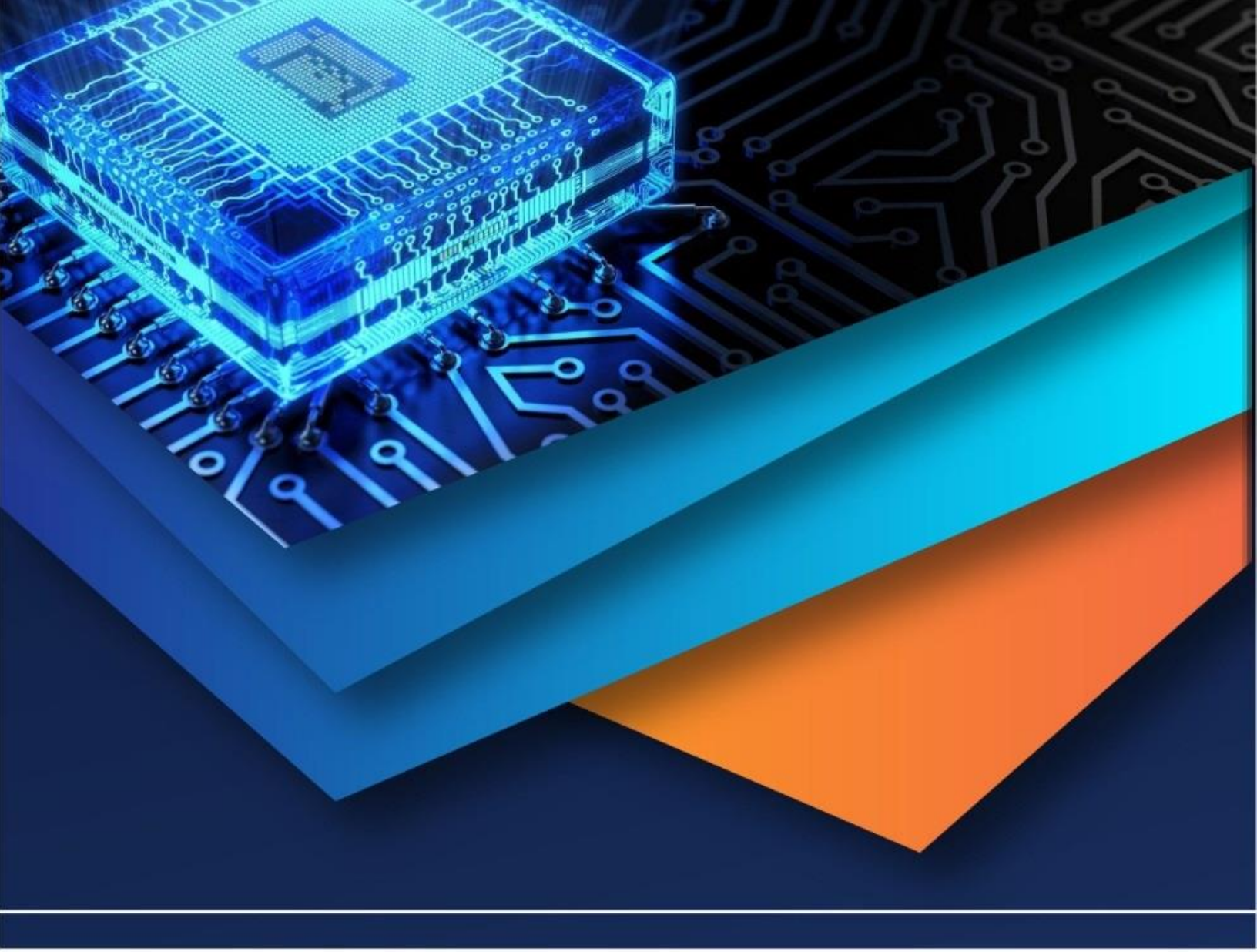
Looking ahead, we expect our work to significantly enhance the overall security posture of web applications. By increasing awareness of CSRF threats and offering practical solutions, we aim to empower developers and security practitioners to better safeguard their systems against malicious attacks.

Ultimately, our research aims to create a safer digital environment by mitigating CSRF risks and strengthening web application security. We remain dedicated to advancing this goal through continued exploration, collaboration, and innovation in cybersecurity.

REFERENCES

[1] Stefano Calzavara, Mauro Conti, Riccardo Focardi, Gabriele Tolomei, Machine Learning for Web Vulnerability Detection ‘The Case of Cross-Site Request Forgery’, IEEE 2020.
[2] Huangcun Zeng, Research on Developing a Lab Environment for Cross-Site Request Forgery,2013.
[3] Ajjarapu Kusuma Priyanka, Siddemsetty Sai Smruthi, Web Application Vulnerabilities: Exploitationand Prevention, IEEE 2020.
[4] W.H. Rankothge, S M.N. Randeniya, Identification and Mitigation Tool For Cross-Site Request Forgery (CSRF), 2020.
[5] Boyan Chen, Pavol Zavarsky, Ron Ruhl Dale Lindskog, A Study of the Effectiveness of CSRF Guard,2011.
[6] Mohd. Shadab Siddiqui, Deepanker Verma, Cross Site Request Forgery: A common web application weakness,2011.
[7] Chaohai Ding, Cross-Site Request Forgery Attack and Defence: Literature Search
[8] Avinash Sudhodanan, Roberto Carbone, Luca Compagna, Nicolas Dolgin,Alessandro Armando, Umberto Morelli, Large-scale Analysis & Detection of Authentication Cross-Site Request Forgeries,2017.
[9] Hossain Shahriar, Mohammad Zulkernine, Client-Side Detection of Cross-Site Request Forgery Attacks,2010.
[10] Nenad Jovanovic, Engin Kirda, Christopher Kruegel, Preventing Cross Site Request Forgery Attacks,2006.
[11] Robin Tommy, Gullapudi Sundeep, Hima Jose, Automatic Detection and Correction of Vulnerabilities using Machine Learning,2017.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)