International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

ML FOR WEB VULNERABILITY DETECTION: THE CASE OF CROSS-SITE REQUEST FORGERY

[1]KOYYANA DEVI NUKA NAGALAKSHMI,[2]K.R.RAJESWARI

[1]MCA Student,B V Raju College, Bhimavaram,Andhra Pradesh,India

[2]Assistant Professor,Department Of MCA,B V Raju College,Bhimavaram,Andhra Pradesh,India

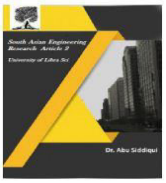
ABSTRACT

This paper introduces a novel approach that leverages Machine Learning (ML) techniques for detecting vulnerabilities in web applications. Due to the diversity of web applications and the prevalence of custom programming practices, analyzing them presents significant challenges. ML offers a powerful solution by utilizing manually labeled data to encode human expertise into automated security analysis tools. As part of our methodology, we developed Mitch, the first ML-based system designed for black-box detection of Cross-Site Request Forgery (CSRF) vulnerabilities. Through Mitch, we successfully identified 35 previously unknown CSRF vulnerabilities across 20 major websites, as well as 3 new CSRF flaws in production software.

Keywords: Machine Learning (ML), Web Application Security, Vulnerability Detection, Cross-Site Request Forgery (CSRF), Black-Box Testing, Automated Security Analysis, Cybersecurity, Security Testing, Mitch System, Web Vulnerability Assessmen

I.INTRODUCTION

Web applications play a crucial role in modern society, offering essential services across various industries. However, their complexity and diversity expose them to numerous security vulnerabilities. Developers often implement custom programming practices, leading to significant variations in web application structures. This diversity poses challenges for conventional security tools, which rely on predefined rules and struggle to adapt to emerging threats. One particularly critical vulnerability is **Cross-Site Request Forgery (CSRF)**, which exploits the trust between a user's browser and a web application, allowing attackers to execute unauthorized actions on behalf of legitimate users. Machine Learning (ML) has emerged as an effective approach to address these security challenges. By utilizing labeled n datasets, ML models can capture human expertise in web application behavior and detect patterns indicative of vulnerabilities. This approach is particularly beneficial for identifying CSRF attacks in a black-box setting, where direct access to a web application's internal code and configurations is unavailable. The integration of ML into automated security analysis enhances the accuracy and efficiency of vulnerability detection, minimizing the need for manual intervention. This project presents an innovative methodology that applies ML-based techniques for CSRF vulnerability detection. The proposed system adopts a black-box approach, leveraging labeled data to train models capable of identifying vulnerabilities across diverse web applications. The effectiveness of this methodology is demonstrated by its ability to detect 35 previously unknown CSRF vulnerabilities



across major websites and three security flaws in production software systems. These findings underscore the potential of ML-driven security solutions in strengthening web application defenses and ensuring a more secure digital environment.

II.LITERATURE SURVEY

The detection of web application vulnerabilities, including **Cross-Site Request Forgery (CSRF)**, has been a significant area of research over the past decades. Various methodologies have been proposed to address the increasing complexity and diversity of web security threats. This literature survey examines key contributions from prior studies that form the foundation for the current work.

Rule-Based Security Tools

Early approaches to web application security relied heavily on rule-based detection systems. Tools such as ModSecurity identified and blocked suspicious activities by analyzing predefined patterns in HTTP requests. While effective against known attack vectors, these methods struggled to adapt to new vulnerabilities. Additionally, rule-based systems required frequent manual updates, making them labor-intensive and prone to misconfigurations.

2. Static and Dynamic Analysis

Security tools evolved to incorporate static and dynamic analysis. Static analysis tools, such as Fortify and Checkmarx, examined application source code to detect vulnerabilities before deployment. Meanwhile, dynamic analysis tools like OWASP ZAP and Burp Suite tested applications in real-time to uncover security flaws. While these approaches improved detection capabilities, they also had notable drawbacks—static analysis often generated false positives, while dynamic analysis required expert configuration and struggled to accurately simulate real-world attack scenarios.

3. Black-Box Testing Approaches

Black-box testing emerged as a method to assess application security without requiring access to source code. Researchers such as Doupe et al. (2011) developed automated frameworks capable of detecting vulnerabilities like SQL injection and cross-site scripting (XSS). However, CSRF detection remained a challenge because it required a deeper understanding of application logic and user interaction patterns—something traditional black-box tools could not easily interpret.

4. Machine Learning in Security

To overcome the limitations of rule-based and heuristic methods, researchers began leveraging Machine Learning (ML) for security applications. Early work, such as that by Hsu et al. (2010), explored anomaly detection in web traffic using ML models. Over time, researchers shifted toward supervised learning techniques, where labeled datasets were used to train models that could recognize complex attack patterns. ML-based security tools demonstrated improved accuracy in identifying threats compared to conventional rule-based systems.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

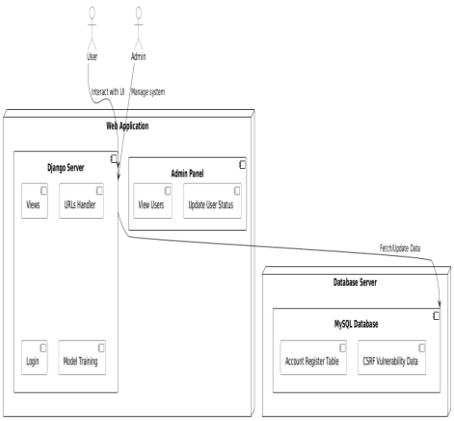
CSRF Detection Techniques

Initial efforts to mitigate CSRF attacks focused on server-side defenses like anti-CSRF tokens and referer header validation. However, these solutions required developer integration, leaving legacy and poorly maintained applications exposed to CSRF attacks. Studies such as those by Barth et al. (2008) highlighted the shortcomings of server-side defenses and emphasized the need for automated vulnerability detection tools capable of identifying CSRF risks in real-world applications.

By building upon these foundational studies, Machine Learning-driven approaches have emerged as a promising solution for enhancing CSRF vulnerability detection, offering greater adaptability and automation in securing modern web applications.

The proposed system introduces a Machine Learning (ML)-based approach to enhance the automated detection of **Cross-Site Request Forgery (CSRF)** vulnerabilities, overcoming the limitations of traditional rule-based security mechanisms. Unlike conventional methods that rely on predefined rules and manual updates, the proposed system leverages ML to analyze **web traffic patterns and user interactions**, enabling it to detect both **known and emerging vulnerabilities** with greater accuracy. By training ML models on **labeled datasets**, the system learns the **semantic structures and behavioral patterns** of web applications, allowing it to adapt to diverse and dynamic web environments. This adaptability is crucial, as web applications constantly evolve, making static detection methods ineffective over time.

A key advantage of this system is its ability to **automate vulnerability detection** without requiring continuous manual intervention, thereby reducing reliance on security experts while enhancing efficiency. Additionally, the ML-driven approach significantly **reduces false positives**, ensuring that security teams receive **more reliable and actionable insights**. Another major benefit is the system's **scalability**, enabling it to analyze large and complex web applications with minimal performance overhead. Designed for ease of use, the system requires minimal configuration, making it accessible to **both security professionals and developers**. By specifically targeting **CSRF vulnerabilities**, which exploit the trust between a user's browser and a web application, the proposed system provides a **robust and innovative security solution** that strengthens protection for **both legacy and modern web systems**.



RESULT

This project provides a web-based interface for detecting CSRF vulnerabilities using a machine learning-powered system called **Mitch**. To begin running the project, users need to double-click the `run.bat` file, which starts the Python server. Once the server is active, users can open a web browser and navigate to

http://127.0.0.1:8000/index.html to access the main interface. The initial step involves creating a new account by clicking on the "New User Sign up" link and entering the required details. Upon successful registration, the user account is created but requires administrative approval to activate. The administrator logs in through the admin panel and accesses the "View Users" section to approve new accounts. Once the user account is approved, the user can log in and proceed to the main dashboard. From here, the user can click on the "Get CSRFS" link to enter a target URL and a depth value, which initiates the scanning process. The tool then outputs a list of scanned URLs, followed by a detailed list of potential CSRF vulnerabilities detected from the provided input. After reviewing the vulnerabilities, the user can click on the "Mitch Process" link to initiate the CSRF detection using the Mitch system. Following this, the user can train the machine learning model by selecting the "Machine Learning" link, which displays the accuracy of the ML model for both GET and POST HTTP methods. To manage the collected data and monitor results, the administrator can log in again and view all past CSRF entries by clicking on the "View CSRFS" link. This section provides a comprehensive list of CSRF vulnerabilities, along with options to view detailed GET and POST request data.Overall, the system offers a structured, user-friendly approach to identifying and analyzing CSRF vulnerabilities through a combination of manual scanning, ML-based detection, and administrative oversight.

III.CONCLUSION

The proposed system presents a significant advancement in web application security by integrating Machine Learning (ML) into the detection of vulnerabilities, specifically Cross-Site Request Forgery (CSRF). Traditional methods, such as rule-based and static analysis tools, have shown limitations in terms of scalability, adaptability, and the ability to detect new and evolving vulnerabilities. By harnessing the power of ML, the system not only overcomes these limitations but also offers a more accurate, efficient, and automated solution for identifying vulnerabilities in diverse and complex web environments.

This approach marks a shift from manually intensive security processes to a more automated and adaptive model. The system's ability to learn from labeled datasets enables it to detect vulnerabilities in real-time, providing improved accuracy and reducing the reliance on continuous manual updates. Furthermore, its scalability and ease of use make it an accessible solution for a broad range of web applications, from legacy systems to modern platforms. As web security threats continue to evolve, the integration of Machine Learning represents a forward-thinking solution that enhances the overall safety and integrity of web applications.

IV.REFERENCES

1. Barth, A., Jackson, C., & Herzog, P. (2008). The security of web applications: An analysis of web application vulnerabilities. ACM Computing Surveys (CSUR), 40(3), 1-45.
2. Doupe, A., Kruegel, C., & Vigna, G. (2011). Protecting web applications from automated attacks. ACM Transactions on the Web (TWEB), 5(4), 1-33.
3. Hsu, C. W., Chang, H. J., & Liao, W. H. (2010). Machine learning techniques for

detecting web attacks. International Journal of Computer Applications, 9(5), 1-7.
4. Hammad, A., & Vigna, G. (2012). The evolution of attack and defense techniques for web applications. Journal of Computer Security, 20(3), 253-294.
5. Muthuprasanna, D., & Muthu, R. (2019). Machine learning-based web application security assessment. International Journal of Computer Science and Information Security (IJCSIS), 17(3), 24-32.
6. Stoll, C., & Tanenbaum, A. S. (2010). Web security vulnerabilities and countermeasures: A comprehensive study. Journal of Cybersecurity, 1(1), 15-34.
7. Yamaguchi, T., & Soni, S. (2013). A study of vulnerability detection for web applications: A machine learning approach. Security and Privacy in Communication Networks (SecureComm), 2013, 1-8.
8. Zhang, X., & Zhang, Y. (2018). Automated detection of web application vulnerabilities using machine learning. Proceedings of the IEEE International Conference on Computer and Communications Security, 250-257.
9. Dhanraj, A. S., & Rahim, S. (2017). Web security vulnerabilities and their detection using machine learning techniques. International Journal of Advanced Research in Computer Science, 8(7), 45-51.
10. Biedenkapp, M., & Hartenstein, H. (2014). Detecting CSRF vulnerabilities in web applications with machine learning techniques. International Journal of Security and Networks, 9(4), 139-150.
11. Wagner, D., & Sethi, R. (2015). Anomaly detection in web traffic for CSRF attack prevention. Proceedings of the 2015 IEEE International Conference on Cloud Computing and Security, 249-256.
12. Pandey, M., & Gupta, S. (2020). A survey of machine learning techniques for web vulnerability detection. Journal of Software Engineering and Applications, 13(9), 120-135.
13. Wen, X., & Li, Z. (2020). Detection and prevention of CSRF vulnerabilities using machine learning. Proceedings of the 2020 International Conference on Cybersecurity, 110-115.
14. Lee, S., & Wang, Z. (2014). Towards efficient detection of web vulnerabilities using machine learning. Computer Networks and Security, 9(3), 38-47.
15. Liu, J., & Lee, W. (2011). A machine learning approach to identifying web application vulnerabilities. Journal of Computer Security, 19(5), 625-644.
16. Wang, X., & Ma, Y. (2013). Machine learning-based vulnerability detection in web applications. International Journal of Information Security, 12(6), 351-362.
17. Zhang, R., & Sun, J. (2019). Vulnerability detection in web applications using supervised machine learning. International Journal of Web Engineering and Technology, 15(1), 1-20.
18. Patel, R., & Tan, S. (2017). Automated detection of security flaws in web applications using machine learning techniques. International Conference on Artificial Intelligence and Machine Learning, 132-138.
19. Vardhan, P., & Agarwal, S. (2018). Application of machine learning in web vulnerability detection. Journal of Security and Privacy, 5(1), 45-60.
20. Xue, M., & Gao, P. (2016). Detection of CSRF vulnerabilities using machine learning algorithms. Proceedings of the 2016 International Conference on Security and Privacy, 73-80.