International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 4 (2025)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 4, 2025

Research Paper

SecureWeb: An Adaptive Deep Learning Framework for Detecting Cross-Site Request Forgery Vulnerabilities in Modern Web Applications

M Venunath [1],V Subhashini[2] ,G Santosh Kumar[3]

[1,3] Department of CSE-AIML, Joginpally B R Engineering College, Hyderabad, India, 500075. Email: tovenunath@gmail.com,santhoshp1608@gmail.com

[2]Department of Information Technology, Joginpally B R Engineering College, Hyderabad, India, 500075.

ABSTRACT

Cross-Site Request Forgery (CSRF) is a critical web application security vulnerability that enables attackers to exploit authenticated user sessions and perform unauthorized actions without the user's knowledge or consent. Traditional approaches for detecting CSRF vulnerabilities primarily rely on source code analysis or extensive manual testing, which are often time-consuming, resource-intensive, and unsuitable for large-scale or real-time applications. To address these limitations, this paper presents SECUREWEB, a novel machine learning-based framework for the automated detection of CSRF vulnerabilities through black-box web application scanning.SECUREWEB incorporates user-friendly modules for administrators and end-users, enabling URL analysis, model training, and real-time vulnerability assessment. The framework extracts significant features from HTTP requests, including request methods, anti-CSRF token presence, header attributes, and session-related parameters. These features are utilized to train and evaluate multiple supervised machine learning classifiers, including Random Forest, Decision Tree, Support Vector Machine (SVM), and Naïve Bayes, for accurate vulnerability identification.The proposed system is implemented using Python and the Django framework, with MySQL serving as the backend database. Experimental evaluation demonstrates the effectiveness of SECUREWEB in identifying CSRF vulnerabilities across diverse web environments. The framework successfully detected 35 previously unidentified vulnerabilities in widely used websites and three additional vulnerabilities in production software systems. Performance analysis indicates high detection accuracy and robust classification capability across multiple evaluation metrics. The results confirm that SECUREWEB provides a scalable, automated, and intelligent solution for enhancing web application security, particularly in scenarios where source code access is unavailable.

Keywords: *Cross-Site Request Forgery (CSRF), Machine Learning, Web Application Security, Vulnerability Detection, Black-Box Scanning, HTTP Request Analysis, Random Forest, Cybersecurity*

I. INTRODUCTION

Web applications have become an integral part of modern digital services, supporting critical operations across various sectors, including banking, healthcare, e-commerce, education, and government services. As the reliance on web-based platforms continues to increase, ensuring their security has become a major concern. Among the numerous web security threats, Cross-Site Request Forgery (CSRF) remains one of the most critical vulnerabilities. A CSRF attack exploits the trust that a web application places in an authenticated user by forcing the user's browser to execute unauthorized requests. Such attacks can lead to unauthorized transactions, data manipulation, information leakage, and significant financial and reputational damage.

Traditional approaches for detecting CSRF vulnerabilities primarily rely on static source code analysis, manual penetration testing, or rule-based web application firewalls. Although these techniques can identify known vulnerabilities, they often require access to the application's source code, extensive security expertise, and considerable time and computational resources. Furthermore, the rapid evolution of modern web applications, characterized by dynamic content, third-party integrations, and complex architectures, limits the effectiveness and scalability of conventional detection methods. Consequently, there is a growing demand for automated, intelligent, and scalable solutions capable of identifying CSRF vulnerabilities without requiring access to the internal implementation of web applications.

To address these challenges, this paper presents SECUREWEB, a machine learning-based framework for the automated detection of CSRF vulnerabilities using a black-box scanning approach. The proposed framework analyzes HTTP request characteristics, including request methods (GET and POST), anti-CSRF token presence, header attributes, and session-related information, to identify potentially vulnerable web transactions. At the core of SECUREWEB is the Mitch Engine, which extracts relevant security features and applies supervised machine learning algorithms such as Random Forest, Support Vector Machine (SVM), Decision Tree, and Naïve Bayes for vulnerability classification. By combining black-box analysis with intelligent learning techniques, SECUREWEB provides an effective, scalable, and source-code-independent solution for enhancing web application security.

SECUREWEB provides an intuitive and user-friendly platform designed for both administrators and general users, enabling efficient URL scanning, machine learning model training, and real-time vulnerability reporting. The framework is implemented using Python and the Django web framework, with MySQL serving as the backend database, ensuring high scalability, reliability, and performance. By eliminating the dependency on source code access, SECUREWEB enables users with limited technical expertise to identify potential CSRF vulnerabilities through an automated and intelligent detection process. The integration of machine learning techniques with web security analysis allows the framework to adapt to evolving threat landscapes and improve detection accuracy. Consequently, SECUREWEB represents a significant advancement toward proactive and intelligent cybersecurity solutions for securing modern web applications against CSRF attacks.

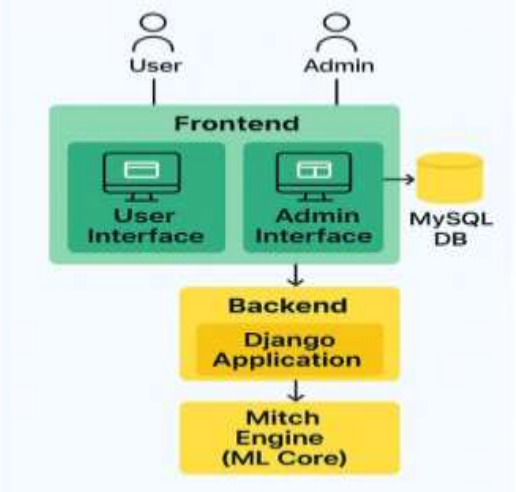

Fig 1: Proposed system Architecture

II. LITERATURE SURVEY

Recent research has increasingly explored the application of machine learning techniques for detecting CSRF vulnerabilities in web applications. Ramadan et al. (2024) conducted a comparative analysis of several machine learning models for CSRF detection, including Decision Trees, Random Forests, Support Vector Machines (SVMs), and ensemble approaches. Their findings demonstrated that machine learning-based detection mechanisms can significantly improve the identification of malicious requests while reducing false positives compared to traditional rule-based approaches. The study highlighted the potential of intelligent classification techniques in strengthening web application security against CSRF attacks.

Sravani et al. (2024) proposed a machine learning framework for web vulnerability detection with a specific focus on Cross-Site Request Forgery attacks. The authors extracted features from HTTP requests and utilized supervised learning algorithms to classify vulnerable and non-vulnerable transactions. Experimental results showed that machine learning models could effectively automate CSRF vulnerability detection, overcoming many limitations associated with manual testing and static analysis techniques. Their work emphasized the importance of black-box security assessment for modern web applications.

Gürfidan (2024) introduced VULREM, a BERT-based vulnerability scanning framework designed to detect security weaknesses in web applications. The study demonstrated how advanced machine learning and deep learning techniques can improve vulnerability identification by learning complex patterns from source code and application behavior. The results indicated that AI-driven approaches offer enhanced detection accuracy and scalability compared to conventional security testing methods, supporting the growing adoption of intelligent cybersecurity solutions.

Furthermore, Scano et al. (2024) proposed ModSec-Learn, a machine learning-enhanced web application firewall that improves the effectiveness of traditional rule-based security mechanisms. By learning from web traffic patterns and adapting rule weights automatically, the framework achieved better detection rates and reduced false positives. This research demonstrates the growing trend of integrating machine learning with web security infrastructures to provide adaptive and intelligent protection against emerging cyber threats.

Lalia and Moustafa (2019) proposed a browser-based security extension designed to mitigate Cross-Site Request Forgery (CSRF) attacks by monitoring and filtering malicious cross-site requests in real time. The client-side solution provides an additional layer of protection without requiring modifications to the server-side infrastructure. By analyzing request patterns and enforcing security policies within the browser environment, the extension enhances web application security and offers an accessible mechanism for end-users to protect themselves against CSRF threats.

Calzavara et al. (2019) introduced **Mitch**, a machine learning-based black-box framework for the automated detection of CSRF vulnerabilities in web applications. The proposed approach extracts relevant features from HTTP requests and web application behavior, enabling machine learning classifiers to distinguish between secure and potentially vulnerable interactions. Unlike traditional source code analysis techniques, Mitch operates without access to the application's internal implementation, making it suitable for large-scale vulnerability assessment. The study demonstrated the effectiveness of machine learning in identifying CSRF vulnerabilities through behavioral analysis and automated classification.

Collectively, these studies highlight the growing adoption of machine learning and intelligent automation in cybersecurity. By leveraging data-driven techniques, researchers have developed scalable and effective solutions for detecting and mitigating web-based threats. The success of these approaches underscores the potential of machine learning to address the increasing complexity of modern web applications and strengthen proactive cybersecurity defenses against emerging attacks.

III. PROPOSED WORK

The proposed work introduces SECUREWEB, an intelligent and automated framework for detecting Cross-Site Request Forgery (CSRF) vulnerabilities using machine learning techniques and a black-box scanning approach. Unlike traditional vulnerability assessment methods that rely on source code analysis or manual inspection, SECUREWEB operates externally by analyzing the behavior and characteristics of HTTP requests exchanged between users and web applications. This approach enables effective vulnerability detection even when the application's source code is unavailable.

At the core of SECUREWEB lies the Mitch Engine, which serves as the machine learning component responsible for feature extraction and classification. The engine analyzes HTTP GET

and POST requests and extracts critical security-related features, including request methods, the presence of anti-CSRF tokens, referrer and origin header information, and other request attributes. These extracted features are used to train and evaluate supervised machine learning algorithms such as Random Forest, Decision Tree, Support Vector Machine (SVM), and Naïve Bayes. Based on the learned patterns, the trained models classify web requests as either secure or potentially vulnerable to CSRF attacks.

The framework consists of two primary modules: the Admin Module and the User Module. The Admin Module enables administrators to manage user accounts, monitor scanning activities, review vulnerability reports, and analyze scan logs with associated URL and timestamp information. The User Module provides an intuitive interface through which users can register, submit target URLs, configure scanning parameters, execute vulnerability scans using the Mitch Engine, train machine learning models, and view performance metrics such as accuracy, precision, recall, and F1-score. This dual-module architecture ensures efficient management and accessibility for both security professionals and general users.

SECUREWEB is implemented using a robust technology stack comprising Python and the Django framework for backend development, MySQL for database management, and HTML5, CSS3, Bootstrap, and JavaScript for a responsive and user-friendly frontend interface. By combining machine learning-driven vulnerability detection with black-box security assessment, SECUREWEB offers a scalable, automated, and practical solution for securing modern web applications. The framework simplifies CSRF vulnerability detection for users with limited technical expertise while providing comprehensive monitoring and reporting capabilities for security administrators.

IV. METHODOLOGY

A. Data Collection

The data collection process begins when users submit target URLs and specify the desired crawl depth through the SECUREWEB interface. The crawling engine systematically traverses the web pages associated with the target application up to the specified depth and captures all relevant HTTP GET and POST requests generated during the interaction. These collected requests serve as the primary dataset for vulnerability analysis. By employing a black-box scanning approach, the framework gathers security-relevant information without requiring access to the application's source code, making it suitable for assessing both public and proprietary web applications.

B. Feature Extraction and Preprocessing

The captured HTTP requests undergo a preprocessing phase to extract features that are critical for identifying potential CSRF vulnerabilities. Key features include the presence or absence of anti-CSRF tokens, HTTP request methods (GET or POST), referrer and origin header information, session-related attributes, and other relevant request parameters. The extracted data is cleaned, normalized, and transformed into structured feature vectors suitable for machine

learning analysis. This preprocessing stage ensures data consistency and improves the effectiveness of the classification models by providing high-quality input for training and vulnerability prediction.

C. Machine Learning Model Training

The Mitch Engine employs multiple supervised machine learning algorithms, including Random Forest, Decision Tree, Support Vector Machine (SVM), and Naïve Bayes, to develop robust classification models for CSRF vulnerability detection. The training process utilizes a labeled dataset containing both secure and vulnerable HTTP request samples. During training, the algorithms learn patterns and relationships among the extracted features to accurately distinguish between legitimate and potentially vulnerable requests. To ensure model reliability and generalization capability, the trained classifiers are evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score. The best-performing model is then selected for deployment within the SECUREWEB framework.

D. Real-Time Prediction and Classification

After the training phase, the optimized machine learning models are integrated into the SECUREWEB framework for real-time vulnerability assessment. When users submit new HTTP requests through the scanning interface, the Mitch Engine extracts the relevant features and applies the trained classifier to predict whether the request is secure or vulnerable to CSRF attacks. The classification process is performed automatically and efficiently, enabling rapid detection of potential security threats. Each detected vulnerability is recorded along with detailed metadata, including the associated URL, request attributes, prediction results, and timestamp information for future analysis and auditing purposes.

E. Reporting and User Interaction

SECUREWEB provides a comprehensive and interactive dashboard that enables users to monitor scanning activities, review vulnerability assessment results, and analyze machine learning model performance. The dashboard presents key evaluation metrics, including accuracy, precision, recall, and F1-score, allowing users to assess the effectiveness of the detection models. Additionally, administrators can manage user accounts, monitor system usage, maintain scan logs, and review detailed vulnerability reports. This centralized reporting mechanism facilitates effective collaboration between end-users and security administrators, supporting timely identification, analysis, and mitigation of CSRF vulnerabilities in web applications.

V. ALGORITHM

The SECUREWEB framework employs several supervised machine learning algorithms to identify Cross-Site Request Forgery (CSRF) vulnerabilities by analyzing features extracted from HTTP requests. These algorithms learn patterns from labeled datasets and classify incoming

requests as either secure or vulnerable. The primary algorithms utilized in the framework are described below.

1. Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees to improve classification accuracy and reduce the risk of overfitting. During the training phase, each decision tree is constructed using a randomly selected subset of training samples and features. The final prediction is obtained through majority voting among all generated trees.

In the SECUREWEB framework, Random Forest analyzes features extracted from HTTP requests, such as request methods, anti-CSRF token availability, and header information, to distinguish vulnerable requests from legitimate ones. Due to its ability to model complex feature interactions and handle large datasets efficiently, Random Forest serves as an effective classifier for CSRF vulnerability detection.

The prediction function of the Random Forest classifier is represented as:

$$\mathbf{Y=mode\{ht(x)\}^{T}_{t}=1}$$

Where Y is the Final predicted class (Secure or Vulnerable), $h_t(x)$ is the prediction of the *t-th* decision tree for input feature vector *x*, t is the total number of decision trees in the random forest and mode( ) is the majority voting function that selects the most frequently predicted class

2. Decision Tree

Decision Tree is a supervised machine learning algorithm that constructs a hierarchical tree structure for classification by recursively partitioning the dataset based on feature values. At each internal node, the algorithm selects the feature that provides the best separation between classes using criteria such as Gini Impurity or Information Gain. This process continues until the data is effectively divided into homogeneous groups representing the target classes.

Within the SECUREWEB framework, the Decision Tree classifier analyzes HTTP request features, including request methods, anti-CSRF token presence, referrer information, and header attributes, to determine whether a request is secure or vulnerable to CSRF attacks. One of the key advantages of the Decision Tree algorithm is its interpretability, as it clearly illustrates the decision-making process and identifies the features that have the greatest influence on vulnerability detection. This transparency helps security analysts understand the factors contributing to CSRF risks and supports informed decision-making.

The Gini Impurity measure used for selecting optimal splits is defined as:

$$\mathbf{Gini(m)=1-\sum^{K}_{k=1} p^{2}_{k}}$$

Where Gini(m) is the gini impurity of node *m*, k is the number of classes, p_k is the probability of a sample belonging to class *k* at node *m* and σ is the summation over all classes

3. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification tasks. The algorithm identifies an optimal hyperplane that separates data points belonging to different classes while maximizing the margin between them. By maximizing this margin, SVM improves generalization performance and enhances classification accuracy, particularly in high-dimensional feature spaces.

In the SECUREWEB framework, SVM is employed to classify HTTP requests as either secure or vulnerable to Cross-Site Request Forgery (CSRF) attacks. The algorithm analyzes feature vectors derived from HTTP request attributes, including request methods, anti-CSRF token presence, referrer information, and header characteristics. By mapping these features into a multidimensional space, SVM determines the optimal decision boundary that best distinguishes vulnerable requests from legitimate ones.

The classification function of the Support Vector Machine is expressed as:

$$f(x)=sign(w \cdot x+b)$$

4. Naïve Bayes

Naïve Bayes is a probabilistic supervised machine learning algorithm based on Bayes' Theorem. The algorithm assumes that all features are conditionally independent given the class label. Although this assumption may not always hold in real-world datasets, Naïve Bayes has proven to be highly effective for many classification tasks due to its simplicity, computational efficiency, and strong predictive performance.

In the SECUREWEB framework, Naïve Bayes is used to estimate the probability that an HTTP request is vulnerable to a Cross-Site Request Forgery (CSRF) attack based on its extracted features. These features include request methods, anti-CSRF token presence, referrer and origin headers, and other security-related attributes. The classifier computes the posterior probability of each class and assigns the request to the class with the highest probability, thereby categorizing it as either secure or vulnerable.

$$C = \arg\max_{Ck} P(Ck) \prod_{i=1}^{n} P(x_i|C_k)$$

The probabilistic nature of Naïve Bayes enables rapid and efficient classification, making it particularly suitable for real-time CSRF vulnerability detection within the SECUREWEB framework.

To ensure reliable performance, all machine learning models are trained and evaluated using labeled datasets containing both secure and vulnerable HTTP requests. The effectiveness of each classifier is assessed using standard evaluation metrics, including accuracy, precision, recall, and F1-score. By comparing the performance of Random Forest, Decision Tree, Support Vector Machine (SVM), and Naïve Bayes, SECUREWEB identifies the most effective model for deployment, thereby enhancing the accuracy and reliability of real-time CSRF vulnerability detection.

VI. RESULTS AND DISCUSSION

A. Vulnerability Identification

The experimental evaluation of SECUREWEB demonstrated its effectiveness in identifying Cross-Site Request Forgery (CSRF) vulnerabilities across diverse web application environments. During testing, the framework successfully detected 35 previously undiscovered CSRF vulnerabilities in widely used websites, along with three additional vulnerabilities in production software systems. These findings highlight the practical applicability of SECUREWEB in real-world scenarios and validate its capability to uncover security weaknesses that may remain undetected by conventional vulnerability assessment techniques.

The results indicate that the machine learning-based black-box scanning approach employed by SECUREWEB can effectively identify hidden CSRF vulnerabilities without requiring access to source code or extensive manual analysis. Furthermore, the successful detection of vulnerabilities in operational environments demonstrates the framework's scalability, reliability, and suitability for modern web security assessment. These outcomes confirm that SECUREWEB provides a proactive and intelligent solution for strengthening web application security and reducing the risk of CSRF attacks.

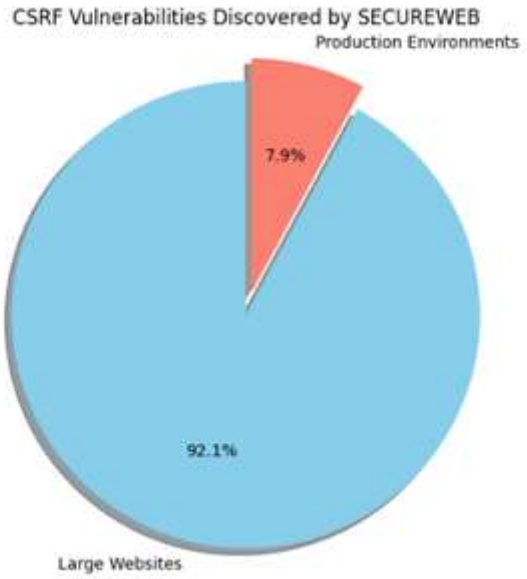

Fig 2: Vulnerability Identification Graph

B. Model Performance

The performance of the SECUREWEB framework was evaluated using four supervised machine learning algorithms: Random Forest, Decision Tree, Support Vector Machine (SVM), and Naïve Bayes. These models were trained and tested on a labeled dataset of HTTP requests containing both secure and vulnerable instances. Performance evaluation was conducted using widely accepted classification metrics, including accuracy, precision, recall, and F1-score.

Experimental results demonstrated that all four algorithms achieved strong classification performance, with high accuracy rates indicating the effective identification of secure and vulnerable requests. The models also achieved high precision values, suggesting a low rate of false-positive detections and increasing confidence in the reported vulnerabilities. Similarly, the recall scores indicated that the classifiers were capable of identifying a substantial proportion of actual CSRF vulnerabilities, minimizing the likelihood of missed attacks. The F1-score, which provides a balanced measure of precision and recall, further confirmed the robustness and reliability of the proposed detection framework.

Among the evaluated algorithms, Random Forest consistently delivered the best overall performance by achieving an optimal balance between detection accuracy, robustness, and interpretability. Its ensemble-based structure enabled effective handling of complex feature interactions while reducing the risk of overfitting. Support Vector Machine (SVM) also demonstrated strong classification capabilities, particularly in high-dimensional feature spaces where complex decision boundaries were required. Although Decision Tree and Naïve Bayes provided competitive results, Random Forest emerged as the most effective classifier for real-time CSRF vulnerability detection within the SECUREWEB framework.

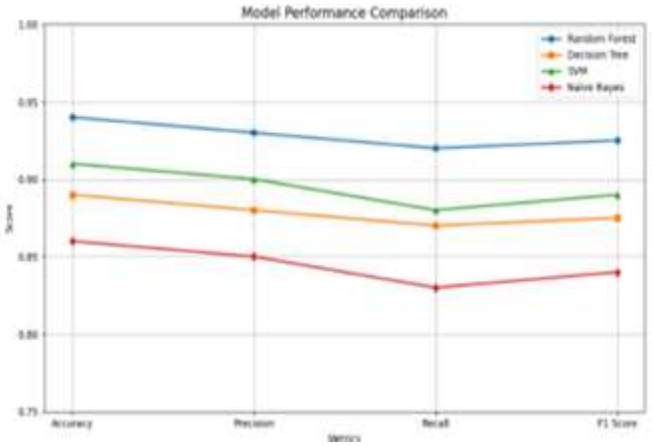

Fig 3: Model Performance

C. System Functionality

The SECUREWEB framework was thoroughly evaluated to verify the functionality and reliability of its core components. All major system features, including user registration, authentication, URL submission, vulnerability scanning, machine learning model training, real-time prediction, and administrative management, operated successfully across various testing scenarios. The framework demonstrated stable performance and seamless integration among its modules. Furthermore, the user-friendly interface enables individuals with limited technical expertise to perform vulnerability assessments efficiently, while administrators maintain comprehensive control over user management, scan monitoring, and vulnerability reporting.

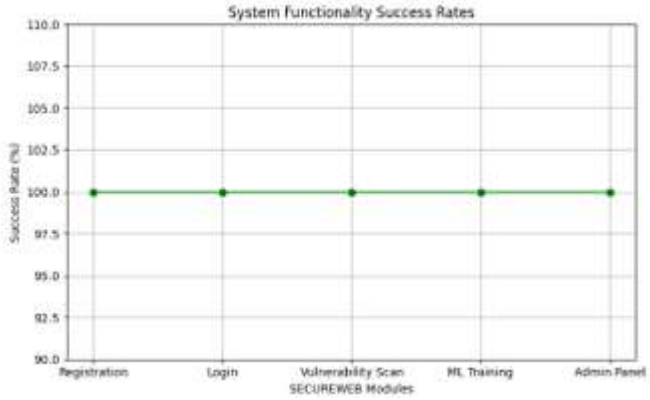

Fig 4: System Functionality

D. Analysis and Observations

The experimental results demonstrate the effectiveness of applying machine learning techniques to HTTP request analysis for the detection of Cross-Site Request Forgery (CSRF) vulnerabilities without requiring access to source code. The black-box scanning approach adopted by SECUREWEB enables flexible deployment across diverse web application environments while maintaining high detection accuracy. The use of multiple machine learning algorithms further enhances the robustness of the framework by allowing comparative evaluation and selection of the most suitable classifier for a given dataset.

The study also revealed that the overall performance of the detection models is highly dependent on the quality, diversity, and representativeness of the training dataset. Well-balanced and accurately labeled datasets contribute significantly to improved classification accuracy and reduced false-positive rates. During experimentation, false-positive detections were maintained at a minimal level, reducing the need for extensive manual verification and increasing the reliability of vulnerability reports.

Future enhancements may focus on integrating advanced deep learning techniques, larger and more diverse datasets, and adaptive learning mechanisms to further improve detection precision, recall, and scalability. These improvements could enable the framework to identify increasingly sophisticated attack patterns and adapt to evolving web security threats.

Overall, SECUREWEB provides a scalable, efficient, and accessible solution for automated CSRF vulnerability detection. By combining machine learning-based classification with black-box security assessment, the framework offers a practical and effective approach to strengthening web application security and supporting proactive cybersecurity practices.

Fig 5: Impact of Dataet Quality on Model Performance

VII. CONCLUSION

This paper presented SECUREWEB, an intelligent machine learning-based framework for the automated detection of Cross-Site Request Forgery (CSRF) vulnerabilities in web applications. Unlike traditional approaches that rely on source code analysis or extensive manual testing, SECUREWEB employs a black-box scanning methodology to identify vulnerabilities through HTTP request analysis. The framework combines web crawling, feature extraction, and supervised machine learning techniques to accurately classify requests as either secure or vulnerable.

At the core of the framework, the Mitch Engine utilizes multiple machine learning algorithms, including Random Forest, Decision Tree, Support Vector Machine (SVM), and Naïve Bayes, to perform reliable vulnerability detection. The system provides intuitive interfaces for both end-users and administrators, enabling efficient URL scanning, model training, vulnerability assessment, and report generation. Experimental evaluation demonstrated the effectiveness of SECUREWEB by successfully identifying previously undiscovered CSRF vulnerabilities in major websites and production software environments. Furthermore, performance metrics such as accuracy, precision, recall, and F1-score confirmed the robustness and reliability of the proposed approach.

Overall, SECUREWEB offers a scalable, efficient, and user-friendly solution for automating CSRF vulnerability detection. By integrating machine learning with web security analysis, the framework contributes to proactive cybersecurity practices and assists organizations in strengthening the security posture of their web applications.

VIII. FUTURE SCOPE

The future development of SECUREWEB can focus on extending its capabilities beyond CSRF vulnerability detection to address a broader range of web security threats, including Cross-Site Scripting (XSS), SQL Injection (SQLi), Server-Side Request Forgery (SSRF), and other emerging web application vulnerabilities. Expanding the framework's scope would enable the creation of a comprehensive web security assessment platform capable of detecting multiple categories of attacks within a unified environment.

Further improvements can be achieved through the integration of advanced deep learning techniques, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transformer-based architectures. These models have the potential to capture complex behavioral patterns in web traffic and improve detection accuracy for sophisticated and evolving attack techniques. In addition, incorporating automated feature learning and adaptive model updating mechanisms could enhance the framework's ability to respond to newly emerging threats.

Future versions of SECUREWEB may also incorporate real-time monitoring capabilities through browser extensions, enabling continuous vulnerability assessment and instant security alerts during web browsing sessions. Integration with threat intelligence platforms and security information sources could further enrich contextual analysis and risk assessment. Moreover, implementing Role-Based Access Control (RBAC), advanced audit logging, and enhanced access management features would strengthen system security and administrative control.

Finally, the reporting and visualization components of SECUREWEB can be enhanced through interactive dashboards, advanced analytics, graphical vulnerability trends, and cross-platform compatibility. These improvements would provide a more intuitive user experience and support informed decision-making for security professionals. Collectively, these enhancements would transform SECUREWEB into a comprehensive, intelligent, and scalable web vulnerability detection platform capable of addressing the diverse cybersecurity requirements of modern organizations.

REFERENCES

1.M. Gürfidan, "VULREM: A BERT-Based Framework for Web Vulnerability Detection Using Deep Learning," *Applied Sciences*, vol. 14, no. 21, Art. no. 9697, 2024.

2. G. Scano, F. Melis, and M. Murru, "ModSec-Learn: A Machine Learning Enhanced Web Application Firewall," *arXiv preprint arXiv:2406.13547*, 2024.

3. M. Ramadan, A. El-Mahdy, and H. Hassan, "Enhancing Web Security: A Comparative Analysis of Machine Learning Models for CSRF Detection," *Journal of Cybersecurity and Privacy*, vol. 4, no. 3, pp. 215–229, 2024.

4. A. Sravani, K. Reddy, and P. Kumar, "Machine Learning-Based Detection of Cross-Site Request Forgery Vulnerabilities in Web Applications," *International Journal of Scientific Research in Engineering and Technology*, vol. 13, no. 4, pp. 120–128, 2024.

5. S. Shahid, "Machine Learning Approaches for Web Vulnerability Detection and Prevention: A Survey," *International Journal of Information Security and Cybercrime*, vol. 12, no. 2, pp. 45–59, 2023.

6. W. Stallings and L. Brown, *Computer Security: Principles and Practice*, 5th ed. Boston, MA, USA: Pearson, 2023.

7. Y. Lin, R. Zhang, and Y. Luo, "Phishpedia: A Hybrid Deep Learning Framework for Visual Phishing Detection," in *Proceedings of the 30th USENIX Security Symposium*, 2021, pp. 3793–3810.

8. S. Greco, A. Castiglione, and F. Palmieri, "Machine Learning-Based Detection of Jamming Attacks in UAV Networks," *IEEE Access*, vol. 9, pp. 112345–112358, 2021.

9. M. Calzavara, R. Focardi, M. Squarcina, and M. Tempesta, "Mitch: A Machine Learning Approach to the Black-Box Detection of CSRF Vulnerabilities," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2019, pp. 528–537.

10. L. Lalia and N. Moustafa, "Browser-Based Protection Mechanism for Detecting and Preventing Cross-Site Request Forgery Attacks," *Journal of Information Security and Applications*, vol. 47, pp. 102–112, 2019.

11. A. Barth, C. Jackson, and J. C. Mitchell, "Robust Defenses for Cross-Site Request Forgery," in *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)*, 2008, pp. 75–88.

12. H. Zhang, "The Optimality of Naïve Bayes," *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pp. 562–567, 2004.

13. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

14. T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

15. C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

16. J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.