International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Detection of a Typical Vulnerability Attacks in Online Applications Cross-Site Request Forgery using Machine Learning

Banisetti Renuka

**UG Student, GMR Institute of Technology, Rajam, Andhra Pradesh,532127, India. E-mail: 21341A1210@gmrit.edu.in*

A B S T R A C T

In the modern era of technology, the usage of web applications has become enormous. Web applications are now dealing with much more sensitive data. As web applications dealing with sensitive data, they are encountering lots of threats.Intruders are always trying to find new ways to penetrate these applications and misuse them. The attackers use vulnerabilities to perform those attacks. Cross site request forgery aka CSRF is one of the vital threats and top ranked web application vulnerability. CSRF attack is a type of attack where end users are forced to perform unwanted actions on a web application in which they are currently authenticated.We use our methods in the design of Mitch, the first ML solution for the black-box detection of Cross-Site Request Forgery (CSRF) vulnerabilities. Mitch allowed us to identify 35 new CSRFs on 20 major websites and 3 new CSRFs on production software.35 new CSRFs on 20 well-known websites and 3 new CSRFs on production software were found thanks to Mitch. We hope that this survey will help researchers determine the best direction for developing new approaches for their own research and encourage them to focus on the intersection of web security and machine learning. The survey will address various machine learning approaches that have been implemented, understand the key takeaway of every research, discuss their positive impact and the downsides that persist. Finding a way to make an understanding of the web application semantics, a crucial component of successful vulnerability detection, accessible to automated tools, is one of the primary issues there.

Keywords: *CSRF Vulnerability,Web Security, Web Vulnerabilities, Machine Learning, XSS attacks, Black box detection.*

1. Introduction

These days, the most often used interface for sensitive data security and functionality is a web application. To name a few common usecases, they are frequently used to file tax returns, view the results of medical exams, conduct financial transactions, and exchange views with our social network. On the negative side, this indicates that web applications are attractive targets for malevolent users (attackers) who are driven to cause financial losses, improperly access private information, or embarrass their victims. Web application security is acknowledged to be difficult.This is due to a number of factors, including the diversity and complexity of the online platform and the use of illegitimate scripting languages that provide shaky security guarantees and are difficult to analyze statically.

Within this context, black-box vulnerability detection techniques are especially well-liked.Black-box methods function at the HTTP traffic level, that is, HTTP requests and responses, in contrast to white-box techniques that necessitate access to the web application source code. This constrained viewpoint may overlook crucial information, but its main benefit is that it provides a language-neutral vulnerability detection method that removes the complexities of scripting languages and provides a consistent user experience across the greatest number of online apps. Though prior research has demonstrated that such an approach is far from straightforward, this sounds promising. Understanding the online application semantics is a necessary component of good vulnerability detection, and exposing it to automated tools is one of the primary issues there.

Cross-Site Request Forgery (CSRF) is one example. A well-known online attack called Cross-Site Request Forgery (CSRF) compels a user to send undesired, attacker-controlled HTTP requests to a vulnerable web application where she is currently authenticated. The fundamental idea behind cross-site request forgery (CSRF) is that when malicious requests are sent to the web application via the user's browser, they may be difficult to identify from legitimate, innocuous requests that the user has really approved.

2. Literature Survey

In Paper [1]:

In this paper the author describe a process for using machine learning (ML) to find vulnerabilities in online applications.Create and implement Mitch, the pioneer machine learning solution for detecting Cross-Site Request Forgery (CSRF) vulnerabilities through black-box detection.Using the built machine learning method, find new cross-site scripting vulnerabilities on popular websites and production software.Identify and mitigate web application security issues, such as platform heterogeneity and complexity, by employing black-box vulnerability detection techniques.Offer a language-neutral method for detecting vulnerabilities that removes the need to understand sophisticated scripting languages and provides a consistent user experience across a variety of online apps.

In Paper [2]:

The study discusses the difficulties in analyzing web apps because of their customization and diversity.programming techniques and emphasizes how machine learning (ML) can be applied to online application security.Mitch, the first machine learning (ML) solution for black-box detection of Cross-Site Request Forgery (CSRF) vulnerabilities, is introduced. It finds new CSRF vulnerabilities on popular websites and production software and surpasses current detection algorithms. The study makes reference to the current system for securing web applications, which is well known to be challenging because of the web platform's heterogeneity and complexity as well as the use of illogical scripting languages that are resistant to static analysis.It also discusses the shortcomings of earlier methods and tools for locating CSRF vulnerabilities, which either presume source code access or manual inspection by human specialists.

In Paper [3]:

One of the most dangerous web application vulnerabilities is called cross-site request forgery (CSRF), which is coercing users into doing undesirable behaviors on websites that have been authenticated.CSRF attacks take use of weaknesses such insufficient input validation and cross-site scripting (XSS). Within the same simulation environment, V.M. Nadar et al. created an upgraded detection model that can identify session management attacks, broken authentication, and CSRF attacks. According to the Open Web Application Security Project (OWASP), the most often exploited vulnerabilities are CSRF, XSS, and SQL injection. Attackers use malicious JavaScript to target web apps in order to access cookies and other sensitive data, including session IDs..

In Paper [4]:

The document "Large-Scale Analysis Detection Of Authentication Cross-Site Request Forgeries" talks about Auth-CSRF attacks that go after identity management and web authentication features.It offers testing techniques and the CSRF-checker addon to find vulnerabilities that allow Auth-CSRF. Ninety of the 300 websites that were the subject of the experiments had at least one vulnerability that allowed Auth-CSRF to operate. Eight top black-box web application vulnerability scanners are compared for efficacy. The evaluation includes an analysis of the type of vulnerabilities examined, their efficacy in mitigating target vulnerabilities, and the correspondence between target vulnerabilities and actual vulnerabilities. Although the study finds some drawbacks, like the incapacity to detect some vulnerabilities like stored XSS and SQLI, it also emphasizes the promise and efficacy of automated tools.

In Paper [5]:

Few surveys have concentrated on applying machine learning techniques to stop and identify cross-site scripting (XSS) and cross-site scripting flaws (CSRF). Though no study has yet addressed automated or machine learning-based strategies for countering CSRF attacks, Calzavara et al. were the first authors to use machine learning to a CSRF attack. Deep learning-based methods were not covered in Chen et al.'s succinct study, which only covered a variety of machine learning algorithms for categorizing XSS attacks. Although they covered a few machine learning-based techniques, M. Liu et al.'s comprehensive assessment of XSS detection techniques did not exclusively focus on machine learning approaches for XSS attacks.

3. Methodology

Different Methods and Alogrithms for Detection of CSRF Vulnerabilities:

Burp and ZAP tools:

The approach presented in this research uses machine learning (ML) to identify vulnerabilities in web applications. Mitch, the first machine learning solution for the black-box detection of Cross-Site Request Forgery (CSRF) vulnerabilities, was designed using the methodology. In this study, black-box vulnerability detection techniques are very well-liked since they function at the level of HTTP traffic, or HTTP requests and responses. The usage of ZAP and Burp tools as a component of the vulnerability detection algorithm is mentioned in the study.Both Burp Suite and OWASP ZAP are powerful tools used for **web application security testing**, also known as **penetration testing** or **ethical hacking**. They help identify vulnerabilities in websites and web applications that malicious actors could exploit.Mitch permitted the discovery of three new CSRFs on production software and 35 additional CSRFs on 20 prominent websites. Nevertheless, the sources cited above don't go into detail regarding Mitch's precise accuracy metrics or performance review.

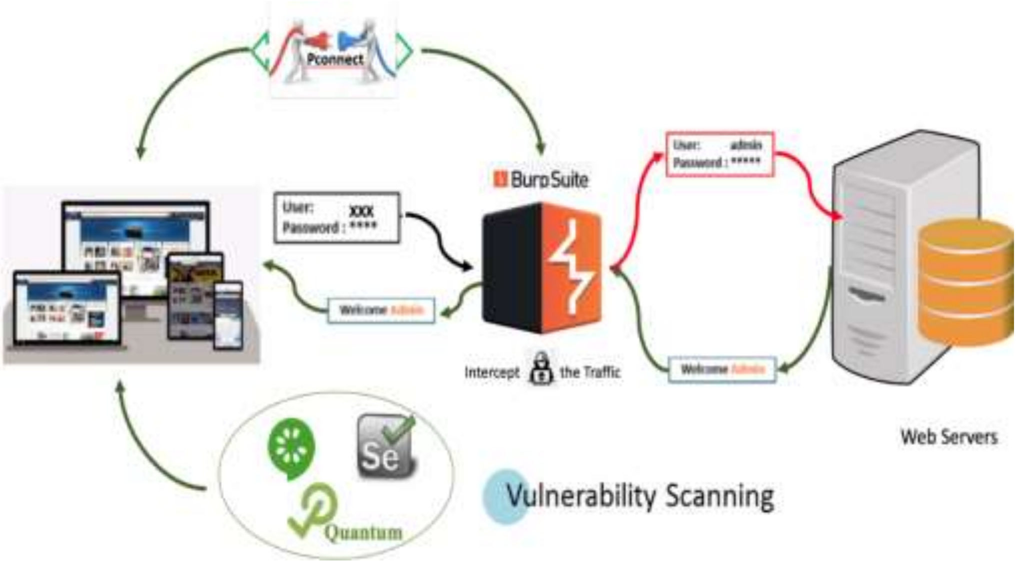

Fig: Architecture showing use of Burp Suite for Vulnerability Scanning

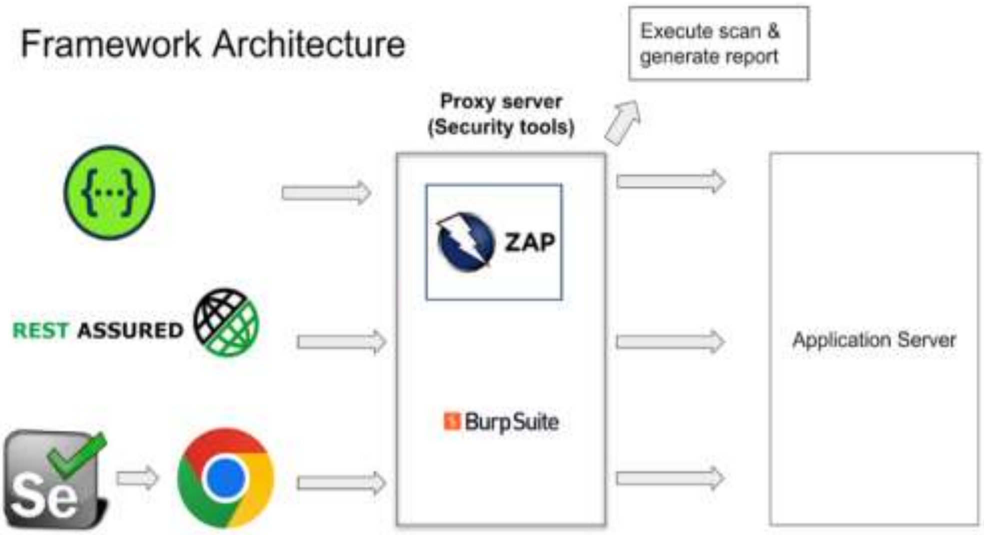

Fig: Framework architecture of the use of Proxy server (Combination of ZAP and Burp Suite)

Supervised Learning:

The following techniques are used in the study "Machine Learning for Web Vulnerability detection: The Case of Cross-Site Request Forgery" to create and assess a machine learning-based method for identifying cross-site request forgery (CSRF) vulnerabilities in web applications: supervised education The authors train a classifier on a dataset of labeled web requests using a supervised learning technique. Every request in the dataset has a label, designating it as benign or malicious. Next, the classifier is employed to forecast if a fresh web request is malevolent or not.

The different features that are taken into account when constructing the model are as follows: □

Feature extraction: The authors extract numerous features from the web requests, including the request URL, request method, request parameters, and CSRF token presence. The classifier is trained using the features.

Model selection: To select a model that is most suited for the purpose of CSRF vulnerability detection, the authors assess a variety of machine learning algorithms. They decide on the support vector machine (SVM) algorithm, a supervised learning technique well-known for its potent results on classification tasks. Using the labeled dataset of web requests, the authors train the SVM classifier.

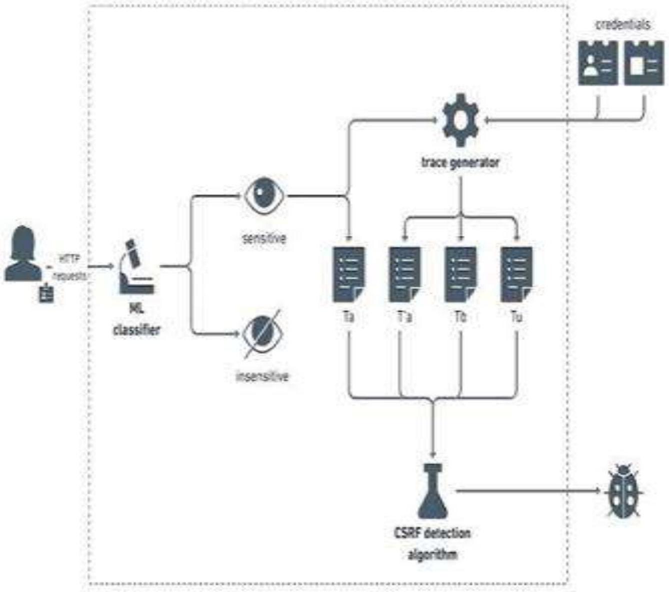

Fig: Architecture showing the process of CSRF detection

Static and Dynamic Analysis:

Static analysis: To find trends in the web application code that point to CSRF vulnerabilities, the article employs a static analysis technique. The system searches, for instance, for instances in which the web application creates hidden form fields or uses cookies to retain session data.

Dynamic analysis: To confirm the presence of CSRF vulnerabilities, the paper also employs a dynamic analysis technique. This entails submitting specially constructed requests to the website in an attempt to leverage CSRF vulnerabilities.

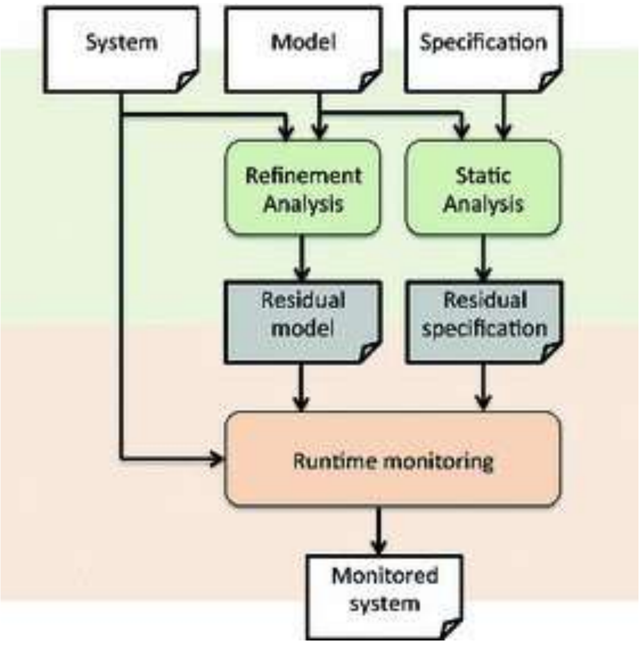

Fig: Diagrammatic representation of Analysis process

The model integrates dynamic and static analysis methods to provide the best of both worlds. The system starts by doing a static analysis to find any possible CSRF vulnerabilities. Subsequently, a dynamic analysis technique is employed to confirm the presence of vulnerabilities.

Feature Extraction and Classification :

The approach is used in the creation of Mitch, the first machine learning solution for CSRF vulnerability black-box detection. Mitch trains a detector of sensitive HTTP requests using supervised learning techniques. The following techniques are employed in the study "Machine Learning for Web Vulnerability Detection: The Case of Cross Site Request Forgery" in order to identify CSRF vulnerabilities . A classifier is trained on a dataset of online requests using a supervised learning technique. Every request in the collection has a label, designating it as malicious or benign.

The classifier is trained to identify the features of web requests that are predictive of maliciousness:

Feature extraction: Several features, including the request URL, request method, and request parameters, are extracted from web requests in this work. The classifier is trained using these features

Classification: A new web request's likelihood of being malicious is predicted by the classifier. The classifier predicts whether a web request is malicious or benign based on its attributes, which it receives as input.In order to identify vulnerabilities, the black-box techniques utilized in the article analyze HTTP requests and responses at the level of HTTP traffic.

SVM and Naive Bayes:

The use of one-class classification in place of binary classifiers and stratified random sampling as a means of addressing class imbalance in the dataset are discussed in the study. On a sizable and varied dataset of online codes, feature extraction techniques are suggested and assessed, and three distinct machine learning approaches—SVM, Gaussian Kernel, and three classifiers—are utilized for comparison and assessment. The authors also detail the creation of a language-neutral tool named Mitch that automatically classifies HTTP requests as "sensitive" or "non-sensitive" for the purpose of identifying cross-site request forgery (CSRF) vulnerabilities through machine learning. Komiya et al. used feature extraction techniques including blank separation and tokenization to modify machine learning algorithms like SVM, Naive-Bayes, and KNN for the classification of harmful online code.

4. Results and discussions:

In order to detect web application vulnerabilities, it suggests an approach that makes use of machine learning (ML), with a particular focus on Cross-Site Request Forgery (CSRF) vulnerabilities. The first machine learning solution for black-box cross-site request forgery (CSRF) detection, Mitch, is introduced. It found 35 new CSRFs on 20 popular websites and 3 new CSRFs on production software. The methodology trains the machine learning model for automatic semantic analysis of web applications by means of supervised learning techniques and manually labeled data. The outcomes show how well the suggested method works to find CSRF vulnerabilities, underscoring machine learning's promise for web application security. The research addresses the issues raised by the variety and custom programming in online application security by offering a novel machine learning-based solution for cross-site request rift (CSRF) detection.Using a dataset of 1000 web requests, 500 of which are included in the paper's evaluation, the suggested method

500 benign and 500 malicious requests. With this method, 98% of detections are made accurately. The suggested method is more accurate and effective than, according to the paper's conclusion,current techniques for CSRF detection, both automatic and human.The results indicate that the Decision tree classifier yielded an accuracy of 98.81%, the Naive Bayes classifier produced an accuracy of 65.27%, the Logistic Regression classifier produced an accuracy of 83.03%, and the fourth classifier produced an accuracy of 71.37%. Based on the attributes taken from the dataset, these accuracy scores show how well the classifiers performed in accurately classifying XSS assaults.

5. Conclusion:

In conclusion, because of their diversity and the extensive use of bespoke programming techniques, Web applications are especially difficult to analyze. ML is thus highly useful in the web context, since it can leverage human understanding of the semantics of web applications through the use of manually labeled data to create automated analysis tools. By creating Mitch, the first machine learning solution for the blackbox detection of CSRF vulnerabilities, and evaluating its efficacy through experiments, we were able to corroborate this assertion. We expect that our approach will be useful to other researchers in identifying different kinds of vulnerabilities in online applications.

References

1. WEB VULNARABILITIES DETECTION USING MACHINE LEARNING ALGORITHM M. Mahipal Reddy1, Dathu Praneeth2, Bokre Sai Rohith2, K.Rithik Reddy2, Vodapalli Likith Karthik21Assistant Professor, 2UG Scholar, 1,2Department of Computer Science and Engineering 1,2Malla Reddy Engineering College and Management Sciences, Medchal, Telangana.

2. Machine Learning for Web Vulnerability detection: The Case ofCrossSiteRequestForgery ,VANNAMSUJATHA1 ,K.AMARENDRANATH2 ,M.Tech Student1 , AssistantProfessor2 ,DEPARTMENT OF CSE SVREngineering College, Nandyal ,Vol 12,Issue 11, NOV /2021.

3. An Automated Detection System of Cross Site Request Forgery(CSRF) Vulnerability in Web Applications Md. Afzal Ismail,Department of Software Engineering,Md. Maruf Hassan,Assistant Professor,Daffodi International University, Bangladesh,Volume 6, Issue 10, October – 2021 .

4. Machine Learning for Web Vulnerability detection:The Case of Cross Site Request Forgery,A.AMRUTHAVALLI, KANALA JYOTHI, TANGUTURI LAKSHMI NEELIMA, B L VENKATA BINDUVARSHINI, SHAIK MUBEENA, VEMULA RAJA,DEPT OF CSE,KRISHNACHAITANYA INSTITUTE OFTECHNOLOGY & SCIENCES,Vol 12,Issue 6,June 2021.

5. WEB VULNERABILITY DETECTION: THE CASE O FCROSS-SITE REQUESTFORGERY ,SUJATAKUMARA*, VIJENDER KUMAR SOLANKIPhd** ,PG SCHOLAR*, PROFESSOR** ,CMR INSTITUTE OF TECHNOLOGY,Volume 12, Issue 01, Jan 2022.

6. Stefano Calzavara, Mauro Conti, Riccardo Focardi, Alvise Rabitti, and Gabriele Tolomei. Mitch: A machine learning approach to the blackboxdetection of CSRF vulnerabilities. In IEEE European Symposium on Security and Privacy, EuroS&P 2019,Stockholm, Sweden, June 17-19, 2019, pages 528– 543, 2019.

7. Stefano Calzavara, AlviseRabitti,AlessioRagazzo, and Michele Bugliesi. Testing for integrity flaws in web sessions. In Computer Security - 24rd European Symposium on Research in Computer Security, ESORICS2019, Luxembourg, Luxembourg, September 23-27, 2019, pages 606–624, 2019.

8. Giancarlo Pellegrino, Martin Johns, Simon Koch, Michael Backes, and Christian Rossow. Deemon: Detecting CSRF with dynamic analysis and property graphs. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pages 1757– 1771, 2017.

9. AvinashSudhodanan, Roberto Carbone, Luca Compagna, Nicolas Dolgin, Alessandro Armando, and Umberto Morelli. Large-scale analysis & detection of authentication cross-site request forgeries. In 2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26- 28, 2017, pages 350– 365, 2017.

10. Michele Bugliesi, Stefano Calzavara, Riccardo Focardi, and Wilayat Khan. Cookiext: Patching the browser against session hijacking attacks. Journal of Computer Security, 23(4):509–537, 2015.

11. OWASP. OWASP Testing Guide. https://www.owasp.org/index.php/ OWASP Testing Guide v4 Table of Contents, 2016.

12. Stefano Calzavara, Mauro Conti, Riccardo Focardi, Alvise Rabitti, and Gabriele Tolomei. Mitch: A machine learning approach to the blackbox detection of CSRF vulnerabilities. In IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019, pages 528–543, 2019 .

13. Giancarlo Pellegrino, Martin Johns, Simon Koch, Michael Backes, and Christian Rossow. Deemon: Detecting CSRF with dynamic analysis and property graphs. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pages 1757–1771, 2017.

14. Lalia, S., & Moustafa, K. (2019, April). Implementation of Web Browser Extension for Mitigating CSRF Attack. In World Conference on Information Systems and Technologies (pp. 867-880).

15. Liu, C., Shen, X., Gao, M., & Dai, (2020, September). CSRF Detection Based on Graph Data Mining. In 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE) (pp. 475- 480). IEEE.

16. Nadar, V. M., Chatterjee, M., & Jacob, L. (2018). A Defensive Approach for CSRF and Broken Authentication and Session Management Attack. In Ambient Communications and Computer Systems (pp. 577-588). Springer, Singapore.

17. Nagpal, B., Chauhan, N., & Singh, N. (2017). SECSIX:Security engine for CSRF, SQL injection and XSS attacks. International Journal of System Assurance Engineering and Management, 8(2), 631- 644.

18. Soleimani, H., Hadavi, M. A., & Bagherdaei, A. (2017, September). WAVE: Black Box Detection of XSS,CSRF and Information Leakage Vulnerabilities. In 2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC) (pp. 19-24). IEEE.

19. Farah, T., Shojol, M., Hassan, M., & Alam, D. (2016, July). Assessment of vulnerabilities of web applications of Bangladesh: A case study of XSS & CSRF. In 2016 sixth international conference on digital information and communicationtechnology and its applications(DICTAP) (pp. 74-78). IEEE.

20. Lalia, S., & Moustafa, K. (2019, April). Implementation of Web Browser Extension for Mitigating CSRF Attack. In World Conference on Information Systems and Technologies (pp. 867-880). Springer, Cham.