Integrating Machine Learning approach with open-source tool for detecting and evaluating vulnerabilities in pursuit of ensuring and implementing security guidelines in Web Applications

Sofi Musha

Department of Computer Science

Univeristy of New York Tirana, Albania

sofimusha@unyt.edu.al

***Abstract*—The security of web applications has always sought to suggest and offer the best practices to overcome potential vulnerabilities and risks in web applications. The main objective of this paper is to integrate the necessary tools with machine learning models as well as employ adequate analysis to discover and diminish well-known vulnerabilities in web applications such as XSS, CSRF or injection attacks. The tool chosen to carry out the task is OWASP ZAP which, throughout this work, is used to scan and analyze the website for potential vulnerabilities and web security threats. Depending on the results, the most severe risks are being mitigated. After obtaining the results from the software, BERT machine learning model is implemented to predict the severity of risks found by the tool and a comparison is made between the outcome of the model and that of the tool. Key advantages regarding the efficiency of the software were observed throughout the security analysis of a web application created in the framework of this paper. Furthermore, the results and insights taken from this study have opened the path to suggestions and recommendations made to efficiently establish the ground for secure web application implementations. It was found that there exists a discrepancy between the outcome of the tool and that of the model suggesting that attention must be paid to also mitigate the vulnerabilities classified as not severe by the tool but severe according to the model. The outcome of this work relies heavily on the structure of the application to be tested, especially on the lack of implementation of predefined security measures.**

Keywords: *security, web application, vulnerabilities, machine learning*

I. INTRODUCTION

Security is an important factor to be considered when designing web applications since it accounts for protection against different vulnerabilities that may lead to identity theft, application malfunctioning, data breaches and other various security risks. The aim this paper pursues is to use OWASP ZAP open-source web application security testing tool in combination with a machine learning model to identify and analyze potential threats in web applications as well as suggest and compile comprehensive guidelines and strategies to address various modern-day security threats. The primary reason behind choosing the tool relies on the fact that OWASP ZAP was found to be very efficient in assisting the user to conduct penetration testing due to its easiness in running automative scans across all operating systems such as in Windows or Linux and generating reports. The model chosen to predict vulnerabilities' severity is BERT which was trained using a dataset compiled by scraping data from NVD [1].

II. APPROACH

A web application needed to perform the analysis of web security has been developed and can be found in the following: [3] . The application is a NodeJS-based web application that implements CRUD functionalities to offer the user the capability to manage books and comments. OWASP ZAP is then downloaded and configured to act as an intercepting proxy that is responsible for capturing and analyzing http and https request and response.[4] In order to enhance the methodology of detecting web vulnerabilities, the tool is set to perform active scanning which in turn sends malformed or special data and requests to the web application for the purpose of identifying common security issues such as cross-site scripting, injections or server configuration insecurities. Upon obtaining the results, the necessary analysis is performed based on the reports and alerts taken from the program so that the most severe risks according to the tool are being mitigated. Techniques such as the usage of cryptographic nonces in conjunction with Helmet.JS [2] module are considered to diminish XSS attacks caused by absence of CSP header while Anti-CSRF tokens and adequate headers are implemented in order to alleviate CSRF and other types of attacks presented in the accompanying poster. In order to evaluate the severity, a BERT model is trained with the data of the National Vulnerability Dataset, and the risks found by the tool are given as input so that a comparison can be made between ZAP evaluation and model classification.

III. EXPERIMENTAL SETUP

The main tool used throughout the work to detect and overcome vulnerabilities in web applications is OWASP ZAP, the benefits of which are due to the fact that it is an open-source beginner-friendly, and efficient project [5]. It is a well-known tool that offers techniques like spidering, active scanning, passive scanning, or fuzzing to effectively identify and report security issues. The BERT model has been trained for three epochs using vulnerability descriptions from NVD classified as low, medium, or high severity. Finally, the risks that the tool has identified are fed into the model in order to predict their severity and relate the output to that of the software.

IV. CONCLUSION

This paper focused on the utilization of OWASP ZAP as an open-source project, to explore the potential vulnerabilities that a web application may have. A BERT model has been implemented and trained using data from NVD so that predicted vulnerabilities' severities are being compared and evaluated against the tool output. The approach and purpose of this work are thought in the framework of establishing rules and best practices that can be followed to guarantee the secure implementation of web applications. First, an outline regarding the roadmap of security in related areas was given as well as a short description of the web application created. Using the above-mentioned tool, a security scanner was carried out so potential vulnerabilities could be detected.Many factors, including the structure of the application, security mechanisms, and server configuration of different web applications affect the results obtained from the tool. Following the results, this paper is developed by providing the appropriate suggestions and implementations to prevent potential security threats that have been classified as the most severe according to ZAP. Then, a BERT model was implemented and used to evaluate risks' severity. The model reached an accuracy of 82.12% suggesting that its output should also be considered when mitigating web application security risks.Finally, a comparison is made between the tool output and the model predictions. From the comparison it was found that there exist differences between the risks severity according to the software and that according to the tool.The study concluded that there are numerous vulnerabilities that should be considered when creating a web application and that it is indispensable to also take into account the output of the model when evaluating the severity of each threat.

References

[1] National vulnerability database. NIST. URL https://nvd.nist.gov/.

[2] helmetjs. Helmetjs. github. URL https://github.com/helmetjs/helmet.

[3] Sofi Musha. Web application. github, 2023. URL https://github.com/sofimusha/unsecuredLibApp.

[4] ZAP. Getting started. URL https://www.zaproxy.org/getting-started/.

[5] Zehntech. Benefits of using zap tool for secuirty testing. 2022. URL https://www.zehntech.com/7-benefits-of-using-zap-tool-for-security-testing/.