Automated Vulnerability Assessment Tools

Srinivasan. M[1], Nistul Premod[2], Prakash. S[3], Dr. R. Shobarani[4], Ms. G. Priyanka [5]
[1, 2 , 3]Dept of B.sc ISCF
[4]Professor
[5]Assistant Professor
[1, 2, 3, 4, 5] Dr.M.G.R Educational and Research Institute, Deemed to be University, Chennai.

Abstract- *Automated vulnerability scanners play a crucial role in identifying and mitigating security risks in web applications. However, existing scanners often have limitations in terms of coverage and accuracy. To address these limitations, we propose an integrated approach that combines multiple vulnerability assessment tools. Our system integrates several state-of-the-art tools, including [UniScan], [Nmap], and [Nikto], etc each known for its specific strengths in vulnerability detection. The integration is designed to leverage the complementary capabilities of these tools, enhancing the overall accuracy and coverage of vulnerability assessments. The scanner operates by first crawling the target web application to discover its structure and identify potential entry points for attacks. It then applies a series of tests using the integrated tools to identify common vulnerabilities such as SQL injection, cross-site scripting (XSS), and CSRF (Cross-Site Request Forgery).*

I. INTRODUCTION

Web applications are an integral part of modern businesses, providing a platform for interaction and transactions with customers and users. However, the increasing complexity and sophistication of web applications have also made them prime targets for cyber attacks. Vulnerabilities in web applications can lead to data breaches, unauthorized access, and other security incidents, potentially causing significant damage to organizations. Automated web vulnerability scanners play a crucial role in identifying and mitigating security risks in web applications. These tools automate the process of scanning web applications for vulnerabilities, such as SQL injection, cross-site scripting (XSS), and CSRF (Cross-Site Request Forgery), which are commonly exploited by attackers. While automated web vulnerability scanners offer many benefits, such as efficincy and scalability, they also have limitations.

Existing scanners may have limited coverage and accuracy, leading to false positives and false negatives. To address these limitations, this project proposes the development of an automated web vulnerability scanner that integrates multiple tools. By combining the strengths of different tools, the scanner aims to improve the accuracy and coverage of vulnerability assessments, providing organizations with a more comprehensive and effective way to protect their web applications against cyber threats.

II. REQUIREMENT ANALYSIS

2.1 OBJECTIVE OF THE PROJECT

The objective of the project is to develop an automated web vulnerability scanner that integrates multiple tools to enhance the accuracy and coverage of vulnerability assessments in web applications.

The scanner aims to:

- Identify common vulnerabilities such as SQL injection, cross-site scripting (XSS), and CSRF (Cross-Site Request Forgery).
- Leverage the strengths of different tools to provide a more comprehensive assessment of web application security. Improve the efficiency of vulnerability detection and mitigation processes.
- Assist organizations in better protecting their web applications against cyber threats. ation and security.

III. REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENTS

- Processor : Dual core processor 2.6.0 GHZ
- RAM : 4 GB
- Hard disk : 320 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard
- Monitor : 15inch colour monitor

3.2 SOFTWARE REQUIREMENTS

- Operating system : Linux (Kali)
- Front End : Linux CLI
- Back End : Python
- IDE : MS Visual Studio

IV. EXISTING SYSTEM

The existing system for automated web vulnerability scanning typically involves using a single tool or a limited set of tools to identify security vulnerabilities in web applications. These tools often have specific strengths and weaknesses, leading to limitations in terms of coverage and accuracy. The other tools can be more time consuming and it also takes more time and effort to scan in different Vulnerability Assesment tools.

4.1 DISADVANTAGE

- Single-tool systems may have limited coverage of vulnerabilities, leaving certain types of vulnerabilities undetected.
- Some systems may struggle to scale effectively, particularly when scanning large or complex web applications, leading to performance issues.
- Some tools can be complex to configure and use, requiring significant expertise to operate effectively.
- Users need to install each tools to scan using multiple tools which is time consuming

V. PROPOSED SYSTEM

The proposed system for automated web vulnerability scanning aims to revolutionize the way organizations approach web application security. By integrating multiple cutting-edge tools and technologies, the system seeks to overcome the limitations of existing solutions and provide a more robust and effective approach to identifying and mitigating security risks. One of the key advantages of the proposed system is its ability to integrate various vulnerability scanning tools, each with its unique strengths and capabilities. By combining these tools, the system can offer a more comprehensive assessment of web application security, ensuring that a wide range of vulnerabilities are identified and addressed. Furthermore, the proposed system incorporates real-time updates from vulnerability databases and threat intelligence feeds. This ensures that the system is always up-to-date with the latest security vulnerabilities and threats, allowing users to stay ahead of potential attacks.

5.1 ADVANTAGES

- The system integrates multiple vulnerability scanning tools, each known for its specific strengths in vulnerability detection. This integration allows for a more comprehensive assessment of web application security.
- Despite its advanced features, the system is designed to be user-friendly, with intuitive interfaces and automated workflows.
- By leveraging the strengths of multiple tools, the system aims to reduce false positives and false negatives, thereby improving the overall accuracy of vulnerability
- Users no longer need to use different tools to scan for different types of vulnerabilities

VI. MODULES

- User Interface (UI):
- Input Processor
- Crawler
- Initial Analysis
- Reporting Module
- Vulnerability Assessment Tools Integration
- Notification System

VII. USER INTERFACE (UI):

1. Provides a graphical interface for users to interact with the scanner.
2. Allows users to input target URLs, configure scan settings, and view scan results.
3. 3.Includes features for managing scan reports
4. 4.Offers a user-friendly experience with intuitive navigation and clear feedback on scan progress.

7.1.Input Processor:

1. Validates user inputs to ensure they are correctly formatted and free of malicious content.
2. Parses user inputs to extract relevant information such as target URLs, scan configurations, and authentication credentials.
3. Transforms inputs into a format that can be used by other modules, such as the crawler and vulnerability assessment tools.

7.2Crawler:

1. Discovers and maps the structure of the target web application by following links and analyzing page content.
2. Identifies entry points for vulnerability testing, such as forms, parameters, and APIs.

7.3Initial Analysis:

1. Analyzes the crawled data to identify potential vulnerabilities based on known patterns and heuristics.
2. Performs static analysis of web pages to detect common vulnerabilities such as SQL injection, XSS, and CSRF.
3. Prioritizes vulnerabilities based on severity, potential impact, and likelihood of exploitation.

VIII. ARCHITECTURE DIAGRAM

It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

IX. IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the users, what it will work efficient and effectively. It involves careful planning, investing of the current system, and its constraints on implementation, design of methods to achieve the changeover methods The coding step translates a detail design representation into a programming language

Realization. Programming languages are vehicles for communication between human and computers programming language characteristics and coding style can profoundly affect software quality and maintainability. The coding is done with the following characteristics in mind.

- Ease of design to code translation.
- Code efficiency.
- Memory efficiency.
- Maintainability.

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective

X. CONCLUSION

In conclusion, an automated vulnerability scanning tool with multiple tools on the Linux platform offers numerous advantages for organizations seeking to enhance their overall security posture. By automating the scanning process and combining multiple scanning techniques, these tools can efficiently detect vulnerabilities and help organizations prioritize and address them promptly. With the continuous advancement of cyber threats, implementing such tools is crucial to ensure the protection of sensitive information and maintain a robust security infrastructure.0

REFERENCES

[1] Odion, T. O., Ebo, I. O., Imam, F. M., Ahmed, A. I., Musa, U. N. (2023). "VulScan: AWeb-Based Vulnerability Multi-Scanner for Web Application." IEEEXplore.DOI:10.1109/SEB-SDG57117.2023. 10124601

[2] RiskOptics. (2022). "Vulnerability Scanners: Passive Scanning vs. Active Scanning." Retrieved from https//reciprocity.com/blog/vulnerability-scanners-passive-scanning-vs-active-scanning/

[3] NCSC (2021). Vulnerability Scanning Tools and Services. [online] www.ncsc.gov.uk. Available at: https://www.ncsc.gov.uk/guidance/vulnerability-scanning-tools-and-services.

[4] Pandey, S., Chaudhary, A. (2022). "Vulnerability Scanning." techrxiv. DOI: 10.36227/techrxiv.20317194

[5] RSI Security. (2023). "7 Types of Vulnerability Scanners." RSI Cybersecurity Blog. Retrieved from https://blog.rsisecurity.com/7-types-of-vulnerability-scanners/

[6] Basan, M. (2023). "12 Types of Vulnerability Scans & When to Run Each." eSecurityPlanet. Retrieved

[7] Grance, T., Stevens, M., & Myers, M. (2003). Guide to Selecting Information Technology Security Products. National Institute of Standards and Technology, NIST Special Publication 800-36.

[8] Tenable. (2023). Tenable Vulnerability Management FedRAMP Moderate User Guide. Retrieved from
[9] NCSC (2021). Vulnerability Scanning Tools and Services. [online] www.ncsc.gov.uk. Available at: https://www.ncsc.gov.uk/guidance/vulnerability-scanning-tools-and-services.
[10] Pandey, S., Chaudhary, A. (2022). "Vulnerability Scanning." techrxiv. DOI: 10.36227/techrxiv.20317194