IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Orchestrating Cyber Elegance: The Artistic Evolution of Automated Penetration Testing

[1]Kirti Prashant Khollam, [2]Dr. Manisha Prakash Bharati

[1]Student, [2] Guide

[1]Department of Technology,

[1]Savitribai Phule Pune University, Pune-411007, India.

Abstract: As organizations, corporations, and political entities increasingly rely on information technology, their exposure to cyberattacks grows. Penetration testing (PT) has become essential for evaluating and enhancing their security against these threats. This article examines automated penetration testing, discussing its evolution, effectiveness, and benefits over traditional manual methods. It thoroughly reviews the main tools used in automated penetration testing and compares them with manual techniques. Additionally, the study explores various methods for developing automated penetration testing platforms, emphasizing their advantages and disadvantages. By integrating information from multiple sources, this research aims to provide a comprehensive understanding of how automated penetration testing can significantly improve cybersecurity measures.

Keywords - Penetration test,Automation

I. INTRODUCTION

Penetration testing, also known as pen testing, tests information system or network security by simulating an attack from a malicious source The goal of penetration testing is to identify real-world security weaknesses and vulnerabilities by attackers can be used for the purpose. However, as modern IT systems have become more complex and larger, there is growing interest in automating investment testing. Automated penetration testing uses software tools and scripts to automate many of the tasks involved in penetration testing. This includes fraud detection, vulnerability assessment, management, and reporting. Management tools can analyze networks and systems faster than a human tester and test for vulnerabilities. Some of the major advantages of automated pen testing include:

1. Larger networks and faster system testing
2. Ability to manipulate.
3. A consistent and repeatable testing process
4. Cost savings compared to manual pen testing.
5. It requires the ability to experiment repeatedly.

However, automated pen testing also has some limitations compared to manual testing. Tools can miss some vulnerabilities that require human analysis and creativity to find. Automation in real-world systems may not always work as expected. And automated tools can produce false positives that require manual verification.

This paper will provide an overview of the current state of automated penetration testing. It will explore the basic tools and techniques, the advantages and limitations of automated testing, and the future prospects for this field. The paper will also present the results of an experiment comparing the effectiveness of automatic and manual penetration testing on a sample network. The aim is to provide an overview of the current capabilities and limitations of automated pen testing, and where the technology is headed.

II. LITERATURE REVIEW

1. A critical component of cybersecurity is penetration testing, or pen testing, which finds weaknesses in networks or systems before bad actors take advantage of them. It involves modelling cyberattacks to assess how well security measures are working. Pen testing can target individuals, offices, and buildings in addition to computer systems. Its objective is to guarantee that systems, firewalls, and gateways are set up correctly to avoid illegal access or service interruption. Penetration testing is a common practice among organizations to strengthen cybersecurity defenses as part of thorough information security audits.

2. J.P. McCarthy McDermott's paper "Attack Net Penetration Testing [1]" discusses methods to detect, approach and eliminate system errors . For this topic, the idea is simply to find and exploit errors, but the process remains the same. The "error scheduling method" breaks the process into a sequence of steps: 1. Define the objective, 2. Change background, 3. Create hypothetical vulnerabilities, 4. Confirm vulnerabilities, 5. Normalize defects, 6. Eliminate defects. For the purpose this topic, we start with the sixth step in this process, enhance the error and then apply the error. McDermott's view of the attack network used for the MiTM attack is the structure of this thesis'; Although, he was pursuing another end goal. The idea of an attack network is that every previous step must be completed to pave the way to the next step, which in turn provides access to the target until the next step. This can be modeled by identifying intermediate and final goals as areas or nodes on a map, and by commands. Input as a transition between these nodes, is a "token" used for the current position on the map as a location. Under ideal circumstances it is possible to find an easier route to the target or if, in a secondary attack, using this route provides the route open for further use

3. Mirjalili [2] and so on. proposed the design and development of a distributed framework for automated web graphics testing, whose main components are an executor executing an attack and control unit calls orchestrator that organizes it in a series of layers Besides providing an integrated method, they defined a flexible method for integrating external tools to achieve desired hacking goals by mapping vulnerabilities of framework integrated tools to realize the full potential of this distributed framework, such as scalability, distributed nature, and ease of use, enabling users to quickly create and deploy applications that leverage the resources available. This framework provides a comprehensive solution for building a distributed hacking system, allowing users to access, edit, and add tools to the system with minimal user-friendly effort make their attacks more efficient and respond quickly to changing circumstances to exploit vulnerabilities before they are patched. As a result, the system enables users to be nimble and agile when exploiting vulnerabilities, allowing them to stay ahead of their opponents. This gives users unprecedented power and flexibility in their hacking activities. In conclusion, a distributed hacking system provides a valuable resource for users looking to increase their cybersecurity. Despite the advantages of distributed hacking, it suffers from process synchronization, although it requires global expertise; resources, the importance of global security awareness; fault tolerance; and error recovery.

4. Fredz et al. [3] described 10 critical web application-related attacks based on well-known web vulnerabilities identified by the Open Web Application Security Project (OWASP) project and, the main recommended strategies to counter those web threats of the edge, associated countermeasures were discussed. Their report provided an overview of the OWASP Top 10 Web Application Security Risks, with detailed descriptions and details of how each risk developed. The report provided an overview of the web security landscape, showing how threat types interact and how organizations should prioritize their securitytheforts in order to address them. Security policies and best practices for managing security risks were discussed. In conclusion, the report provides a valuable resource for organizations looking to understand web application security and implement appropriate countermeasures By providing this overview of the OWASP Top 10 , the report provides a good tool for organizations to take a proactive approach to protecting their web applications The current threat landscape was identified, OWASP Top 10 security risks were highlighted, and organizations included risk mitigation plans to better protect their web applications Furthermore, the report found the need to implement secure coding practices, educate web application developers about OWASP Top 10 vulnerabilities They also discussed how to avoid them and how to the use of automated security scans that can detect vulnerabilities in web applications is discussed.

5. Neha Samantha's thesis Automated Penetration Testing [4]explores the need for automated penetration testing to reduce the time and cost of manual processing penetration analysis. By doing penetration testing early and often during network development or software development stages, it's easier and cheaper. It takes time to identify or resolve issues as opposed to waiting until the end of the process. The paper developed a web-based system, which launched a variety of vulnerable exploitation (DoS) attacks. The application was able to handle basic DoS attacks using three protocols: Hypertext Transfer Protocol (HTTP), Session Initiation Protocol (SIP) and Message Control Protocol/Internet Protocol (TCP/IP). There were up to two to twenty attacks for each protocol. The rich man a friendly interface accepts the input of an IP address

for an attack, and the port number that identifies that attack. The application successfully completed this attack with one user Friendly interface that made penetration testing easier as opposed to manual attacks. The weakness of this approach is that it will be scaled up in the future and there is a need to continue using external systems to gather information before attacking. It wasn't his first Architecture established, application development is limited and possibly limited in efficiency. The lack of integration on a port or IP discovery program is reduced . The intelligent and dynamic nature of the system. The object of this thesis is to build a framework that can be established before developing a web application to ensure long-term viability and to integrate systems for full system functionality.

6. In their comprehensive analysis of adversarial reconnaissance strategies, Roy Shanto et al. [5] offer a taxonomy-based survey. Reconnoitring tactics such as target footprinting, social engineering, network scanning, and local discovery are categorised by the taxonomy according to their technological approach.

7. Doupé et al. [6] evaluated 10 web application evaluation tools: Acunetix, AppScan, BurpSuite, Arachni, Hailstorm, NTOSpider, Paros, N-Stalker, Webinspect, and W3af. These resources were chosen with consideration for both commercial and open-source scanners.

8. Lilan Hu et al. [7]provided a thorough analysis of the state-of-the-art in security vulnerability detection technology, along with the process of developing machine learning technology. A cross-site scripting security vulnerability detection model for web applications is designed and implemented, and the requirements of the security vulnerability detection model are fully analysed. By adding the verification code identification function, the issue of data submission to the server requiring the verification code only being entered is resolved. This function is based on the current network vulnerability detection technology and tools.

9. Jinfeng Li [8] coordinated industry norms Development teams can review and address code vulnerabilities to reduce false positives found in static scans and penetration tests, aiming for increased accuracy of the findings. This application security framework is produced by combining the top 10 OWASP vulnerabilities and the top 25 dangerous CWE/SANS software errors in a matrix with Checkmarx vulnerability queries. By using Checkmarx vulnerabilities queries to map OWASP/SANS, defects and vulnerabilities are shown to be managed more effectively.

10. In order to stimulate more study by constructing a system of the current approaches on a larger horizon, Amir Manzor et al. [9]provided a literature assessment summarising security solutions and significant flaws. They stated that as there is no way to fully mitigate all online vulnerabilities, more research on digital information security is needed. Additionally, they spoke about how complicated the procedures in the survey were and offered some advice on how to use them.

11. Ouissem ben Fredz et al. [10] OWASP identifies the most important network vulnerabilities according to the Top 10, including their associated attacks and countermeasures. They also proposed an application that provides protection against the most severe attacks by web applications and prevents some zero-day exploits.

12. Richard Amankwah et al. [11] use two insecure online apps, WebGoat and Damn insecure online Application, to compare the vulnerability detection skills of eight web vulnerability scanners (both open and commercial). OWASP ZAP, IBM AppScan, Acunetix, HP WebInspect, Vega, Iron WASP, Skipfish, and Arachni were the eight WVSs that were examined. Multiple assessment metrics were used to assess the performance, including the web application security scanner evaluation criteria, Youden index, precision, recall, and OWASP web benchmark evaluation. The findings of the experiment demonstrate that, although commercial scanners are effective in identifying security flaws, some open-source scanners (like ZAP and Skipfish) can also be successful in doing so.

III. GAP ANLAYSIS

Table 1 Gap analysis between the research papers

#	Source	Gap Analysis	Suggested Solution
1.	McDermott's paper "Attack Net Penetration Testing" [1]	Lacks a detailed description of how to leverage automated tools and frameworks for efficient and scalable testing.	Integrate automated penetration testing tools and frameworks into McDermott's workflow. Consider using machine learning techniques to identify and prioritize vulnerabilities for enhanced productivity and scalability.
2.	Mirjalili et al.'s Framework for Automated Web Graphics Testing [2]	Lacks robust synchronization mechanisms and fault tolerance capabilities in its distributed approach.	Enhance the framework with improved process synchronization techniques and fault tolerance mechanisms. Incorporate advanced error recovery strategies to ensure reliability and stability of the distributed hacking system.
3.	Fredz et al.'s Report on Web Application Security Risks and Countermeasures [3]	Provides an overview of security risks and countermeasures but lacks detailed guidance on implementing secure coding practices and educating developers.	Develop comprehensive training programs for web developers on secure coding practices and OWASP Top 10 vulnerabilities. Integrate automated security scanning tools into the development process for early vulnerability detection.
4.	Neha Samantha's Thesis on Automated Penetration Testing [4]	Introduces a web-based system for automated penetration testing but lacks scalability and integration with external systems for information gathering.	Enhance the system's scalability by optimizing resource utilization and implementing parallel processing techniques. Integrate external systems for reconnaissance and information gathering to enhance effectiveness.
5.	Evaluation of adversarial reconnaissance techniques by Roy Shanto et al [5].	Classifies analytical methodsbut lacks practical application guidelines and real-worldcase studies.	Develop practical guidelines and case studies for the use of adversary survey techniques in real-world situations. Provide hands-on training to security professionals for better understanding and implementation.
6.	Doupé et al.'s Assessment of Web Application Assessment Tools [5]	Provides insight into various web application analysis tools but lacks a comparative analysis of their effectiveness in detecting specific vulnerabilities.	Perform a detailed comparative analysis of web application analysis tools based on their ability to detect specific vulnerabilities. Develop and incorporate a scoring system for evaluating the effectiveness of the tools.
7.	Research by Lilan Hu [6] and colleagues on Security Vulnerability Detection Technology	Identifies a model for detecting security vulnerabilities but lacks demonstration of effectiveness through real-world application.	Conduct comprehensive testing and deployment in varied environments. Form partnerships with industry stakeholders to collect feedback and enhance the model according to real-world utility.
8.	Jinfeng Li's Application Security Framework [7]	Focuses on elevating the precision of identifying vulnerabilities but falls short in aligning with current development methodologies and tools.	Assimilate Li's framework into existing software development practices and create supplementary tools such as plugins for widely used Integrated Development Environments (IDEs) and CI/CD systems. This facilitates automatic vulnerability assessments during development phases.

9.	The literature survey by Amir Manzor and team [8]	Catalogs prevalent web security measures and vulnerabilities but does not provide tailored strategies to combat nascent security threats.	Conduct ongoing investigative efforts to pinpoint and mitigate emerging threats in web security. Establish a collaborative network with industry and academic entities to ensure a continual update on cutting-edge security developments.
10.	The initiative by Ouissem Ben Fredge and associates [3]	Proposes a system to safeguard web applications but overlooks the necessity for ongoing surveillance and periodic updates to keep pace with new threats.	Implement real-time threat detection systems and establish a routine for regular software updates and patches to maintain robust defence mechanisms against evolving threats.
11.	The study by Richard Amankwah and his group [9]	Provides an analysis of the efficacy of web vulnerability scanners but lacks an exhaustive evaluation of their user-friendliness and integration features.	Broaden the assessment criteria to encompass factors like ease of use, integration flexibility, and customization support. Offer detailed guidelines for choosing and setting up web vulnerability scanners tailored to meet the unique needs of different organizations to enhance their practicality and effectiveness.

IV. PROPOSED SOLUTION

This project aims to develop a cutting-edge graphical user interface (GUI) that brings together several top-tier automated penetrations testing tools into one streamlined platform. Crafted using advanced Python programming, this platform is designed to greatly enhance the efficiency and accuracy in identifying and managing web vulnerabilities. This strategic integration ensures a seamless user experience, facilitating thorough security assessments through a unified interface.The backend of the GUI leverages Python for its flexibility and the extensive libraries that support complex security tasks. The Python framework is carefully developed to automate interactions with the APIs of the integrated tools, thereby streamlining the scanning process and improving data management and error-handling tasks. The system's performance is further enhanced through the implementation of advanced programming techniques such as multi-threading and multiprocessing, allowing multiple scans to run concurrently while maintaining speed and accuracy. Additionally, secure coding practices are emphasized to protect data integrity and confidentiality throughout the platform's lifecycle. Designed with simplicity and functional efficiency in mind, the GUI features a central dashboard that provides real-time updates on scan progress and detailed analyses of detected vulnerabilities. Users can directly adjust the settings of each integrated tool, allowing for customized scans tailored to specific security needs and organizational policies. The GUI's analysis module offers in-depth evaluations of vulnerabilities, classifies them by severity, and provides actionable remediation strategies. Furthermore, the platform includes robust report generation capabilities to meet compliance and documentation requirements.To address the constantly evolving cybersecurity landscape, the GUI incorporates a proactive updating and enhancement mechanism. This mechanism ensures that the integrated tools and scanning methodologies are regularly updated and improved based on the latest security research and user feedback. This proactive approach ensures the platform remains at the forefront of technology, ready to tackle new and emerging security challenges effectively. In summary, the proposed GUI for automated penetration testing is designed to democratize the process of conducting advanced cybersecurity assessments. It aims to make comprehensive security evaluations accessible to a wide range of users, from security professionals to organizational IT teams. By integrating the capabilities of various specialized tools into a cohesive and intuitive platform and reinforcing it with sophisticated Python programming focused on optimizing user experience and data security, this solution is set to significantly enhance an organization's ability to defend against complex web-based threats. This user- centric approach ensures the platform not only meets current cybersecurity needs but is also poised to adapt and evolve, maintaining its relevance and effectiveness in a rapidly changing digital environment. Types of vulnerabilities scanned by the application include sqli, xss, os command injection, clickjacking, etc.

V. RESULTS

- Successfully detected multiple vulnerabilities.
- Generated PDF reports with severity levels.
- Reduced manual effort in penetration testing.
- Improved scanning efficiency.

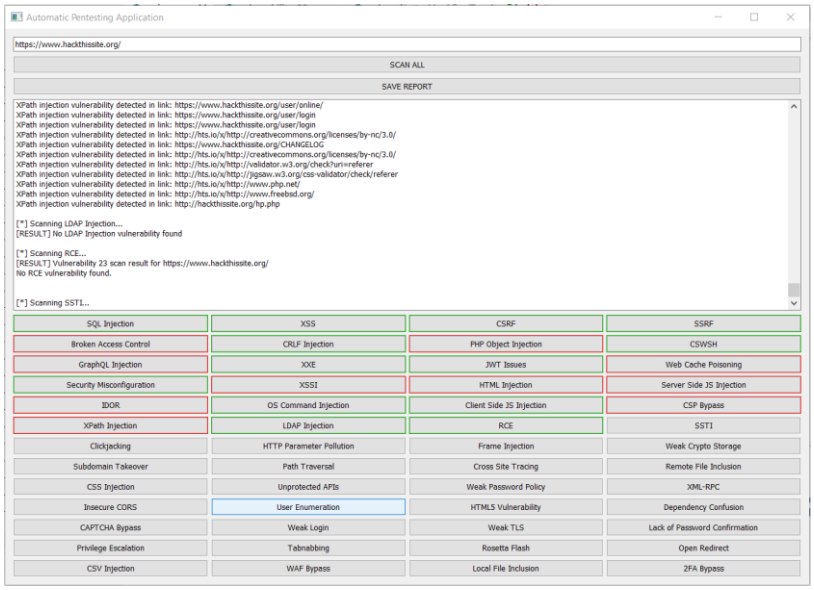

Figure 1 Detected Multiple vulnerabilities for the target url (scan all button)

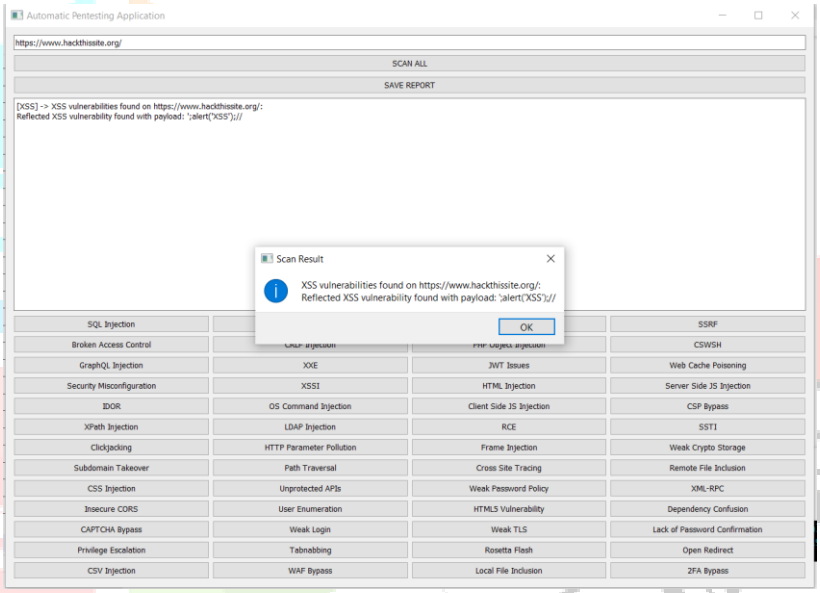

Figure 2 Scanned for individual vulnerability by selecting particular vulnerability from the options (xss in this case)

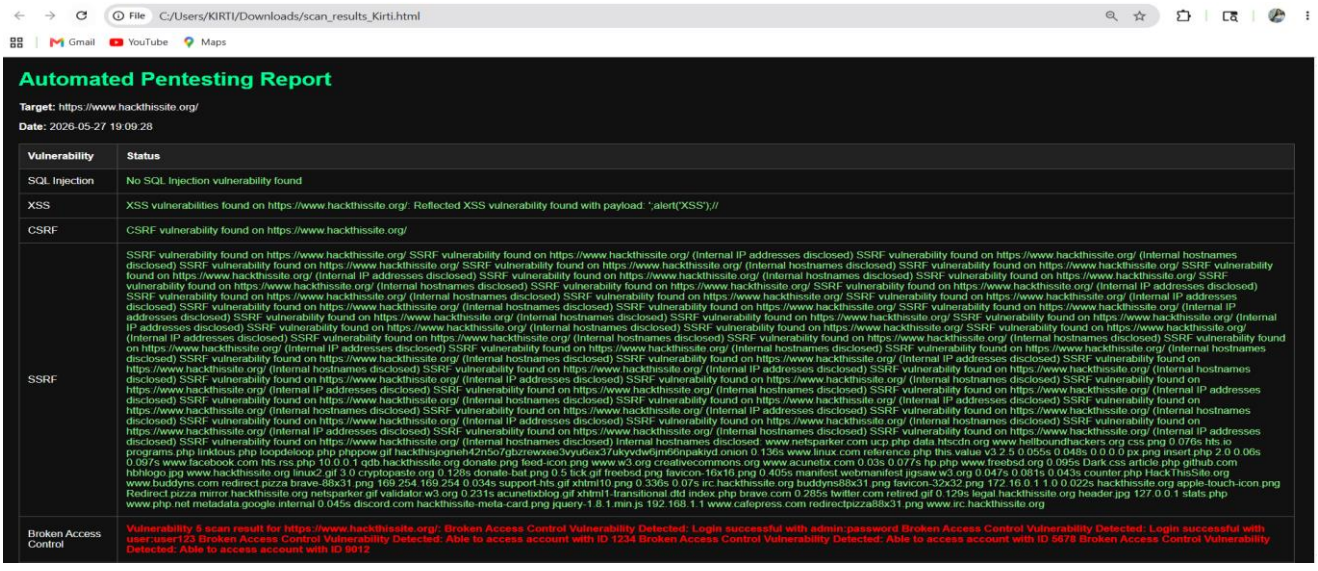

Figure 3 Automated Pentesting Report (save report option- html report file)

VI. ACKNOWLEDGMENT

I would like to express my deepest appreciation to all those who provided the possibility to complete this research paper. I give special gratitude to my project guide, Dr. Manisha Bharati, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project, especially in writing this research paper.

Lastly, I would like to express my gratitude to my parents for their undivided support and attention, who inspired me to strive towards my best.

REFERENCES

[1] J. P. Mcdermott, "Attack net penetration testing," Attack net penetration testing, p. 8, February 2001.

[2] M. Mirjalili, A. Nowroozi and M. Alidoosti, "A survey on web penetration test," A survey on web penetration test, p. 16, November 2014.

[3] O. B. Fredj, O. Cheikhrouhou, M. Krichen and H. Hamam, "An OWASP Top Ten Driven Survey on Web Application Protection Methods," An OWASP Top Ten Driven Survey on Web Application Protection Methods, p. 19, February 2021.

[4] N. Samant, AUTOMATED PENETRATION TESTING, Master's Projects, 2011.

[5] A. Doup´e, M. Cova and G. Vigna, "Why Johnny can't pentest: An analysis of black-box web vulnerability scanners," International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Bonn, Germany, pp. 111-131, 2010.

[6] L. Hu, J. Chang, Z. Chen and B. Hou, "Web application vulnerability detection method based on machine learning," Journal of Physics: Conference Series; IOP Publishing: Guiyang, China, vol. 1827, no. 021061, p. 7, 2021.

[7] J. Li, "Vulnerabilities mapping based on OWASP-SANS: A survey for static application security testing," Annals of Emerging Technologies in Computing (AETiC), vol. 4, no. 3, p. 8, 2020.

[8] M. Noman, M. Iqbal and A. Manzoor, "A Survey on Detection and Prevention of Web Vulnerabilities," (IJACSA) International Journal of Advanced Computer Science and Application, vol. 11, no. 6, p. 20, 2020.

[9] R. Amankwah, J. Chen, P. K. Kudjo and D. Towey, "An empirical comparison of commercial and open-source web vulnerability scanners," Software: Practice and Experience, vol. 50, no. 1842-1857, p. 22, 2020.

[10] P. Pandit, Nessus: Study of a Tool to Assess Network Vulnerabilities, Mumbai , 2021.

[11] S. ROY, N. SHARMIN, J. C. ACOSTA, C. KIEKINTVELD and A. LASZKA, "Survey and Taxonomy of Adversarial Reconnaissance Techniques," ACM Computing Surveys, vol. 55, no. 6, p. 112, 2022.

[12] A. Doupé, M. Cova and G. Vigna, "Why Johnny Can't Pentest: An Analysis of Black-Box Web Vulnerability Scanners," p. 22, 2010.

[13] D. Stuttard, "Burp Suite's web vulnerability scanner," PortSwigger.

[14] O. B. Fredj, . o. cheikhrouhou, M. Krichen, H. Hamam and A. Derhab, "An OWASP top ten driven survey on web application protection methods," 15th International Conference, CRiSIS, vol. 12528, no. 235, p. 17, 2021.