IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SAFE NET INSIGHTS

(Web Vulnerability Scanner)

[1]Dr. Gayatri Bachhav, [2]Adit Chemburkar, [3]Sarth Aradhye, [4]Shreyas Bhatkar
[1]Project Guide, [2]Student, [3]Student, [4]Student
[1] Information Technology
[1]Vasantdada Patil Pratishthan's College of Engineering and visual arts , Mumbai, India

Abstract: The growing reliance on the internet for various aspects of daily life has made web applications a prime target for cyberattacks. In this digital age, understanding web vulnerabilities is paramount for the security of both businesses and individuals. This paper provides a comprehensive analysis of web vulnerabilities, focusing on their types, root causes, and potential consequences.

The study first categorizes web vulnerabilities into common types, such as SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and others. It explores the underlying mechanisms and techniques that threat actors employ to exploit these vulnerabilities, shedding light on the reasons behind their prevalence in web applications. To combat these vulnerabilities, this paper presents a systematic framework for mitigation that encompasses proactive measures, including secure coding practices, input validation, and threat modeling, as well as reactive strategies like vulnerability scanning and patch management. It also includes Real-world examples and case studies are used to illustrate the effectiveness of these mitigation techniques.

Index Terms **-Web vulnerabilities, Web applications, Cyberattacks, Security**

I. INTRODUCTION

In an increasingly interconnected world, the internet serves as a fundamental pillar of modern society, shaping how we communicate, conduct business, and access information. However, with this heightened connectivity comes an ever-present threat: web vulnerabilities. These vulnerabilities, often lurking beneath the surface of web applications and services, represent a significant concern for individuals, organizations, and society as a whole.

Web vulnerabilities are security weaknesses and flaws in web applications that can be exploited by malicious actors. These vulnerabilities pose a grave risk, as they can lead to data breaches, unauthorized access, financial losses, and the compromise of user privacy. From small e-commerce websites to large multinational corporations, no entity is immune to the potential devastation that web vulnerabilities can bring.

This introductory exploration seeks to shed light on the critical subject of web vulnerabilities by offering an overview of their nature and significance. We will delve into the common types of web vulnerabilities, dissect the root causes behind their existence, and discuss the far-reaching consequences of exploiting them. Furthermore, we will highlight the importance of understanding and addressing these vulnerabilities, not only for security professionals but for developers, website owners, and end-users alike.

II. RESEARCH GAP

Emerging Threats: As technology evolves, new types of web vulnerabilities may arise. Research should continuously adapt to explore and understand these emerging threats, which may include vulnerabilities associated with IoT (Internet of Things) devices, serverless computing, or evolving web technologies like WebAssembly.

Vulnerability Patching and Management: Research into effective vulnerability patching and management strategies is essential. Understanding how organizations can efficiently and proactively respond to new vulnerabilities, especially in large and complex web applications, remains a critical area for improvement.

Web Security Education and Training: As web technologies and attack vectors evolve, there's a need for research on the most effective methods for educating and training developers and users about web security.

III. AIM

Identify Vulnerabilities: The primary aim is to discover and catalog web vulnerabilities comprehensively, encompassing a wide range of attack vectors, from well-known issues like Cross-Site Scripting (XSS) and SQL injection to emerging threats associated with evolving technologies.

Risk Assessment: Research aims to quantify and assess the real-world risks associated with web vulnerabilities, providing organizations with a clearer understanding of the potential impact of exploitation and helping them prioritize their security efforts.

User Education: Ensuring that both developers and end-users are well-informed about web security is a critical aim. Research seeks to identify effective methods for educating and training individuals in secure practices and risk awareness.

IV. OBJECTIVE

Vulnerability Identification: Identify and catalog various types of web vulnerabilities, including but not limited to Cross-Site Scripting (XSS), SQL injection, Cross-Site Request Forgery (CSRF), and more.

Risk Assessment: Quantify the risks associated with web vulnerabilities to help organizations prioritize their mitigation efforts, focusing on vulnerabilities that pose the greatest threat.

User Education and Awareness: Educate developers, administrators, and end-users about web security best practices, vulnerabilities, and safe online behavior to reduce the likelihood of exploitation.

V. LITERATURE

[1] have proliferated in recent years, and are considered as the main platform for future business transactions such as financial banking, e-commerce, infotainment and administrative reforms. However, with the popularization of the Internet and the rapid evolution of web technologies, online security is facing increasingly severe challenges And due to the increase in the availability of online information and services, the security risks of web applications have reached an unprecedented level.

[2] In order to avoid the attacks from hackers, quite a mass of research on web vulnerability have been constructed such as web vulnerability identification, web vulnerability scanning, web vulnerability classification etc. Among them, web vulnerability scanning has been considered as a fundamental process in web security, and cooperates with firewalls or intrusion detection systems to effectively improve the reliability of web applications. In essence, vulnerability scanning can be regarded as the penetration testing which is widely adopted to eliminate the potential threats of web systems. More specifically, it is a test-based testing approach that simulates the attacks of malicious hackers to evaluate the security of web applications

[3] At present, massive scanners are developed to detect possible web vulnerabilities. However, they commonly only focus on a single target and carry out independent vulnerability scanning, which greatly restricts the efficiency of revealing vulnerabilities. To address this problem, in this paper, an automatic web vulnerability scanner is proposed based on the latest AWVS framework to collect the related information and carry out comprehensive vulnerability detection with a given target.

[4] In detail, the proposed scanner has finally realized the three main functions as follows, that is, related information collection, ports service scanning and web vulnerability detection. And, it is proved by the experimental results that, considering the scanning scope and the scanning effectiveness, our scanner presents great performance.

[5] The Open Web Application Security Project(OWASP) has summarized the top 10 web security vulnerabilities(shown on Table II). Among them, the most common and harmful vulnerabilities are the SQL injection and the cross site scripting(XSS). To evaluate the effectiveness of the proposed scanner, we mainly focus on these two vulnerabilities and construct a series of experimental testings. In this section, detailed descriptions of the two vulnerabilities are given below

[6] SQL Injection SQL injection has been widely accepted as one of the most common vulnerabilities, and brings high-level security risk to web applications. The means of SQL injection is to maliciously modify the SQL query parameters in URL or HTTP request package. Under the case that the server executed the modified SQL statements without strictly checking, thus, attacks are initiated successfully and hackers are capable of performing illegal operations. Moreover, firewalls cannot defend against the attacks launched by SQL injection. The underlying reason is that, to satisfy the normal communication of web applications and databases, the connections must be established between clients and web servers. Hence, once the web application contains a SQL injection vulnerability, the server is potentially threatened. In addition, since SQL injection statements are generally embedded in ordinary HTTP requests, it is difficult to distinguish them from normal statements. Therefore, the traditional methods of defense against SQL injection commonly present quite unsatisfactory effectiveness.

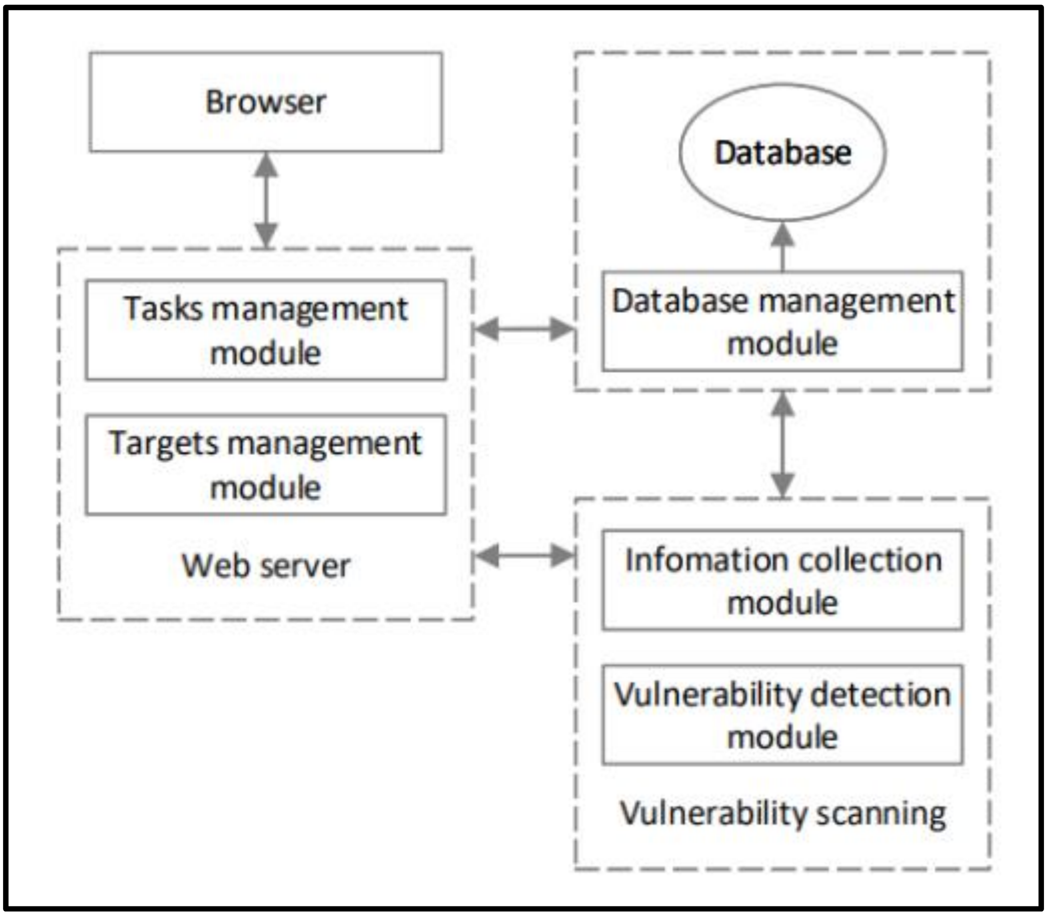
VI. METHODOLOGY AND TECHNIQUES

6.1 Methodology Used

The development and deployment of web applications have become an integral part of modern life, offering convenience, connectivity, and a wealth of services. However, these digital platforms are also exposed to an array of security risks, making the role of web vulnerability scanners indispensable in safeguarding these systems. This section focuses on the methodology and techniques underpinning web vulnerability scanners, shedding light on how they operate to detect and mitigate potential security weaknesses.

Methodology and techniques for web vulnerability scanners encompass a systematic approach to identifying vulnerabilities within web applications and websites. The key objective is to emulate potential attack vectors, discover vulnerabilities, and provide actionable insights for remediation.

6.2 Block Diagram


Browser
Database
Tasks management module
Database management module
Targets management module
Web server
Infomation collection module
Vulnerability detection module
Vulnerability scanning

6.3 Software Implementation

Module Designs The proposed web vulnerability scanner is designed based on python programming language. And the B/S architecture(Browser/Server) has been adopted for realization. More specifically, our database, the information collection module and the vulnerability detection module are deployed on the server side. For users, they can manage the running tasks through the browser. As the framework shown in Fig. 2, the main function modules are presented as follows, including (1) Database and the management module, (2) Information collection module, (3)Vulnerability detection module, (4) Tasks and targets management module. We have introduced these modules in detail below.

6.3.1 Database and the management module

Our database is established based on SQLite in version 3.24.0 which has great flexibility without configuration requirements. The main function of the database is to store the vulnerability detection results and the collected information such as related assets and the description of scanned ports, etc. For each project, we have respectively constructed the corresponding database. In detail, the asset collection database will store the name, ip, title, banner and other information of the target website; For the port database, it saves the port numbers and the service identifications of those open ports; The vulnerability database collects the results from vulnerability detection including scanning targets, utilized vulnerability scripts and the types of vulnerability, etc.

6.3.2 Information collection module

The information collection module realizes the following two functions, that is, asset collection and port scanning, service identification. For asset collection, it expands the scope of target assets by subwebsite exploration. The mainly utilized technologies for DNS discovery including enumeration, online interface, DNS querying and searching engines. Moreover, we also implement a high-speed DNS stub resolver namely Massdns with multi-threading to further improve the efficiency of our scanner. During the penetration testing, the information about the related ports and the inside services are required. The underlying reason is that, there are commonly some running services behind the open ports, and the services may be vulnerable to attackers. In order to verify whether the open services are vulnerable, we firstly scan the target port, and then fingerprint recognition is implemented to access its open services. Finally, a series of testing activities can be carried out.

6.3.3 Vulnerability detection module

After collecting the related information of the set target, we may obtain a large scope of testing objects to carry out vulnerability detection.that there are two detection models provided by our scanner, including comprehensive detection and special detection. For comprehensive detection, we mainly focus on the common web security vulnerabilities such as SQL injection, XSS, framework vulnerability, etc. It is worth mentioning that our scanner is able to be deployed in other remote servers. Hence, the distributed scanning is achieved without using local sources, which greatly improves the scanning efficiency. The special vulnerability detection module is developed based on the piste 3. And it utilizes the outcome of the previous information collection to perform targeted scanning of specific objectives such as CMS, frameworks, special vulnerabilities, special scripts, etc.

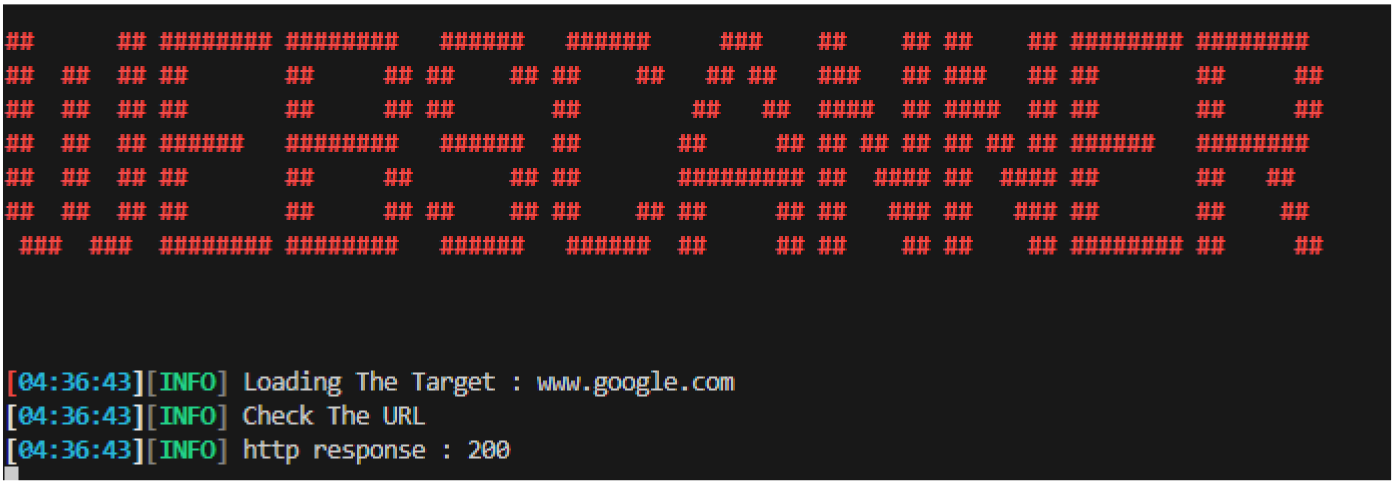
6.3.4 Tasks and targets management module

Both the tasks management module and the targets management module are deployed on a web server, and the verified users can control the scanner to perform desired operation through the browser. The main function of the tasks management module is to adjust the settings of our scanner such as detection model selection and results displaying, etc. A major advantage of the proposed scanner is that it can expand the scanning scope by adding the related targets according to a specific target. To further control the scanning scope, users are able to modify the scanning targets by the targets management module.

VII. CONCLUSION

A web vulnerability scanner system plays a pivotal role in modern cybersecurity, addressing the critical challenge of safeguarding web applications from an ever-evolving landscape of security threats. By identifying and mitigating vulnerabilities in web applications, organizations can protect sensitive data, ensure regulatory compliance, and maintain their reputation. However, building and maintaining an effective scanner system is not without its challenges. These include the need to accurately identify vulnerabilities, reduce false positives and false negatives, adapt to complex web application architectures, and continuously monitor for emerging threats. Prioritizing vulnerabilities, integrating with development workflows, addressing the performance impact of scanning, and respecting data privacy and compliance regulations are also significant considerations. Human expertise and scalability are vital, as is the ability to allocate resources effectively and stay resilient against emerging threats. In a constantly changing threat landscape, an effective web vulnerability scanner system must address these challenges to provide comprehensive and reliable protection for web applications, ultimately contributing to an organization's security and risk management efforts.

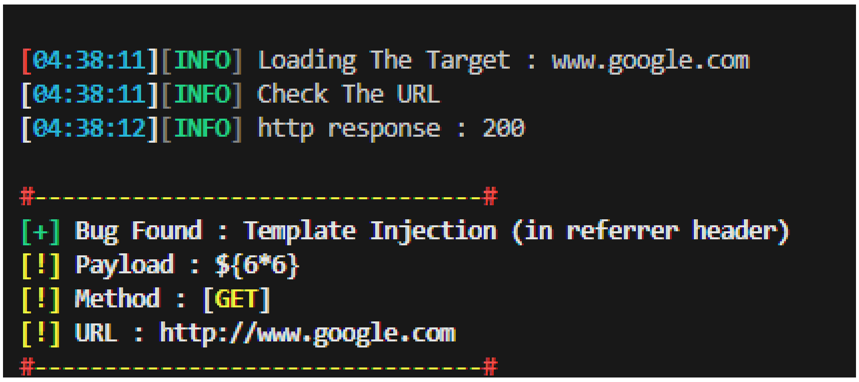
VIII. RESULT



```
##      ## ######## ########   ######   ######     ###    ##    ## ##    ## ######## ########
##  ##  ## ##       ##     ## ##    ## ##    ##   ## ##   ###   ## ###   ## ##       ##     ##
##  ##  ## ##       ##     ## ##       ##        ##   ##  ####  ## ####  ## ##       ##     ##
##  ##  ## ######   ########   ######  ##       ##     ## ## ## ## ## ## ## ######   ########
##  ##  ## ##       ##     ##       ## ##       ######### ##  #### ##  #### ##       ##   ##
##  ##  ## ##       ##     ## ##    ## ##    ## ##     ## ##   ### ##   ### ##       ##    ##
 ###  ###  ######## ########   ######   ######  ##     ## ##    ## ##    ## ######## ##     ##



[04:36:43][INFO] Loading The Target : www.google.com
[04:36:43][INFO] Check The URL
[04:36:43][INFO] http response : 200
```



```
[04:38:11][INFO] Loading The Target : www.google.com
[04:38:11][INFO] Check The URL
[04:38:12][INFO] http response : 200

#--------------------------------#
[+] Bug Found : Template Injection (in referrer header)
[!] Payload : ${6*6}
[!] Method : [GET]
[!] URL : http://www.google.com
#--------------------------------#
```

IX. REFERENCE

1. T. Heath and C. Bizer, “Evolving the web into a global data space,” in Linked Data: Evolving the Web into a Global Data Space, 2011.
2. P. Colton and U. Sarid, “System and method for developing, deploying, managing and monitoring a web application in a single environment,” 2009.
3. X. U. Feng, N. University, Nanjing, N. University, and Nanjing, “Research and development of trust management in web security,” Journal of Software, vol. 13, no. 11, pp. 2057–2064, 2002.
4. S. M. Bellovin, W. R. Cheswick, S. M. Bellovin, T. W. Hacker, W. R. Cheswick, and S. M. Bellovin, “Firewalls and internet security: Repelling the,” Pearson Schweiz Ag, 2003.
5. Y. W. Huang, S. K. Huang, T. P. Lin, and C. H. Tsai, “Web application security assessment by fault injection and behavior monitoring,” 2003.
6. E. Reshef, Y. El-Hanany, G. Raanan, and T. Tsarfati, “System for determining web application vulnerabilities,” 2002.
7. D. Maynor, Metasploit Toolkit for Penetration Testing, Exploit Development,, 2007.