JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

WEBSITE VULNERABILITY SCANNING

Saravanakumar M [1], Aruitselvan Sha [2], Azees Sulthan S [3], Dharun R [4], Manikandan A [5],

Assistant Professor, Department of C.S.E, Jansons Institute of Technology, Coimbatore, India[1]

UG Students, Department of C.S.E, Jansons Institute of Technology, Coimbatore, India [2-5]

Abstract : In today's interconnected world, cybersecurity has emerged as a critical aspect impacting various spheres including employment and education. However, it's alarming that only a minority are cognizant of significant web vulnerabilities. Statistical analyses reveal that numerous small-scale industries have direct or indirect internet connectivity, making them susceptible to cyber threats. This project aims to conduct penetration testing to address this issue. Websites with vulnerabilities are highly susceptible to hacking. Hence, we propose to develop a scanner using penetration testing techniques to detect vulnerabilities in websites. Vulnerability scanning involves inspecting potential exploit points on a website to identify security vulnerabilities. This process helps in identifying system weaknesses, bugs, and vulnerabilities in communication equipment. Additionally, it aids in predicting the effectiveness of countermeasures.

IndexTerms **-Cybersecurity, vulnerability, penetration testing, website scanning**

INTRODUCTION

Web applications are now an indispensable part of daily life, yet many harbor vulnerabilities. Despite the widespread availability and affordability of website hosting, security measures have not kept pace. These vulnerabilities pose risks ranging from small-scale enterprises to large industries. Swift rectification of these flaws is crucial to restore an organization's reputation following unauthorized exploitation. Hence, vulnerability scanners are invaluable for identifying known weaknesses and vulnerabilities within websites. As the number of online applications continues to rise, ensuring their security becomes paramount. Detecting vulnerabilities before they are exploited is essential for safeguarding user security. Vulnerability assessment entails identifying vulnerabilities across various platforms, not limited to a single application. This comprehensive approach considers all factors relevant to assessing system vulnerability and security. Consequently, vulnerability scanners are utilized for network and software application scanning.

MECHANISM OF SCANNERS

The mechanics of a scanner involve a three-step process: crawling, simulation of attacks (fuzzing), and response analysis. During crawling, the scanner explores the web application and its associated input pages, creating an index of all visited pages. A thorough crawling mode is crucial to ensure no vulnerabilities are overlooked. In the fuzzing step, the scanner sends attacking patterns to previously identified inputs, with the attacker module generating values to trigger vulnerabilities. In the response analysis phase, the results of fuzzing are monitored to determine if the web application is vulnerable, providing feedback to other modules. With the prevalence of website hacking incidents, ensuring web application security has become increasingly vital. Web application security scanning software detects vulnerabilities by testing web applications without scanning the source code directly.Vulnerability management comprises several key components, including identifying vulnerabilities, assessing risks, mitigating risks, and reporting security gaps. Identification of vulnerabilities involves administrators identifying security weaknesses within the network, capturing as many vulnerabilities as possible. While top companies often have dedicated teams for this purpose, automated tools have streamlined these efforts, saving time. The next phase involves evaluating risks, where not all vulnerabilities are equally critical or urgent. Scanning tools identify vulnerabilities and classify them, aiding in prioritization. Once risks are identified, the focus shifts to addressing them. Utilizing the right tools can automate the process of provisioning devices to tackle vulnerabilities. After vulnerabilities are mitigated, scanning software can generate reports to assess the system's security status.

EXISTING SOLUTIONS

For testing and evaluating vulnerability using scanners, a vulnerable web environment has to be formulated. This is fulfilled by vulnerable web applications that are specially designed to provide users, the environment to identify the attacks and the way to rectify it. This section deals with some of the scanners that can evaluate the vulnerabilities of a web Application.

NMAP

Nmap serves as a potent port scanner utilized for comprehensively examining network hosts and their associated ports. By inputting an IP address, Nmap gathers extensive data pertinent to the specified host. It not only identifies the host but also

determines the number of active ports, distinguishes between open and closed ports, and discerns the services associated with these ports, including those utilizing TCP or FTP protocols. Furthermore, Nmap employs sophisticated algorithms to predict the operating system employed by the host. The scan results are often visualized in graphical format, depicting the network topology and gateways accessed by the local machine to reach the target host. Notably, the identification of open ports serves as a critical indicator of potential vulnerabilities, as unauthorized access can be facilitated through such ports. Nmap's versatility allows for scanning a diverse range of ports, empowering security professionals to comprehensively assess network security and fortify defenses against potential threats.

NESSUS

Nessus functions as a potent vulnerability scanner, meticulously identifying vulnerabilities within remote hosts. It offers both internal and external scans, with internal scans focusing on hosts within a specific router, while external scans extend to hosts beyond a designated router. Additionally, Nessus facilitates web application testing capabilities. Scanning operations can be initiated either instantly or by configuring a template tailored to the host, allowing for efficient and simultaneous scanning of multiple hosts. Vulnerabilities are assessed based on four severity levels: high, medium, low, and informational. Upon completion of a scan, results are automatically saved, presenting vulnerabilities categorized by plug-in and host. The former categorization lists all vulnerabilities detected during the scan and the hosts affected by them, enabling the generation of comprehensive reports for vulnerability remediation. The latter focuses on host-related issues, facilitating follow-up scans and assessments as necessary. Moreover, Nessus supports the export of results in various formats to suit specific requirements. Operating on a client-server architecture, each scanning session is initiated and controlled by the client, with tests executed on the server side. Notably, Nessus offers scalability, enabling the scanning of over 100 websites simultaneously.

ACUENTIX

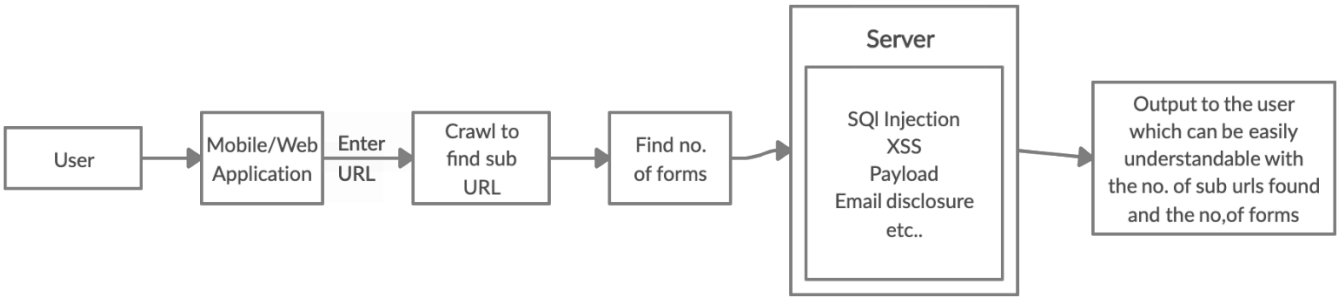
Acunetix stands out as an automated web application security testing tool designed to meticulously assess web applications for vulnerabilities such as SQL injection, cross-site scripting, and exploit vulnerabilities. This tool conducts comprehensive scans of websites or web applications accessible via web browsers. Notably, Acunetix excels in handling custom-based web applications leveraging JavaScript and AJAX technologies. Its advanced crawler is adept at uncovering any file within the target application.The scanning process unfolds in three key phases: target specification, site crawling and structure mapping, and pattern analysis. During target identification, Acunetix actively probes the target to verify its presence and assesses whether it hosts any web applications. Vital information is gathered regarding the target's web technologies, server type, and responsiveness, which are crucial for tailoring subsequent tests.In the subsequent phase of structure mapping and site crawling, Acunetix retrieves the index file of the target web application based on the provided URL. This process involves capturing responses to extract links and input fields, thereby compiling a comprehensive list of directories and files within the web application.The final step involves pattern analysis, wherein Acunetix systematically executes various analysis techniques against the web application to identify vulnerabilities and potential security threats. Through its meticulous scanning process, Acunetix empowers organizations to bolster the security posture of their web applications, mitigating the risk of potential cyber threats and ensuring robust protection against vulnerabilities.

NIKTO

Nikto is a command-line tool utilized for scanning specific targets, employing Perl language for its scanning operations. It conducts comprehensive security checks to identify potential vulnerabilities within web applications, particularly targeting dangerous files that may pose security risks. Attackers often exploit vulnerabilities in web applications, particularly those hosted on outdated Apache servers, to gain unauthorized access. Nikto, being a free and open-source scanning tool, serves as a valuable resource for IT enterprises to identify security flaws within their systems and take proactive measures to enhance protection and upgrade their infrastructure. Moreover, Nikto is adept at discovering servers that were not developed internally by the organization, thereby enabling comprehensive vulnerability assessment across diverse environments.

PROPOSED SYSYTEM

As internet usage continues to surge, web applications face escalating vulnerabilities, rendering them susceptible to unauthorized attacks. To combat this growing threat, numerous online scanners have emerged in the market. However, many of these scanners fall short in detecting all vulnerabilities, leaving systems vulnerable to exploitation. In scenarios where the deployed scanner fails to detect a vulnerability, attackers can easily infiltrate the system, compromising data and resources. To address this challenge, we propose a comprehensive vulnerability scanner capable of detecting a range of vulnerabilities, including SQL injection, cross-site scripting, broken authentication, payload, and email disclosure. Our scanner conducts thorough website scans to identify these vulnerabilities, ensuring robust protection against potential threats. The scanner is available in both mobile and web application formats, providing users with flexibility in accessing its functionalities. Users can input the URL into the application, prompting the scanner to crawl through the URL and its sub-URLs. Upon completion, the scanner generates a detailed report listing any identified vulnerabilities, along with additional information such as server details, technology utilized, and certification information. This approach empowers organizations to proactively identify and address vulnerabilities, bolstering their overall cybersecurity posture.



SQL INJECTION

SQL Injection (SQLI) exploits a vulnerability in a web application's handling of user inputs, allowing attackers to execute malicious SQL commands on the underlying database. This vulnerability arises when user inputs are not adequately sanitized or validated by the application before being passed to the database query.During an SQLI attack, attackers input specially crafted SQL commands into vulnerable input fields of a web application. These commands are then executed by the application's database without proper validation, enabling attackers to manipulate the database and potentially gain unauthorized access to sensitive information.Attackers can exploit SQLI vulnerabilities to perform various malicious actions, including accessing, modifying, or deleting database records, executing arbitrary SQL commands, and even gaining unauthorized access to the underlying operating system through techniques like command injection.The consequences of SQLI attacks can be severe, ranging from unauthorized access to sensitive data to complete compromise of the affected system. Therefore, it is crucial for developers to implement robust input validation and parameterized queries to mitigate the risk of SQL injection vulnerabilities. Additionally, regular security assessments and vulnerability scanning can help identify and address potential SQLI vulnerabilities in web applications.

CROSS SITE SCRIPTING

Cross-Site Scripting (XSS) vulnerability arises when unauthorized users can inject malicious code into a web application. This vulnerability stems from inadequate validation or sanitization of input parameters. There are three main types of XSS vulnerabilities: Non-Persistent (Reflected XSS), Persistent (Stored XSS), and Document Object Model (DOM)-based XSS.Non-Persistent XSS occurs when a web application accepts a malicious request from a user, which is then echoed back in the application's response in an unsafe manner. Persistent XSS arises when a web application stores a malicious request in a data source and subsequently displays this information to a wide range of users. DOM-based XSS, on the other hand, does not require server-side validation and operates within the victim's web browser environment, modifying the DOM to execute the payload.In XSS attacks, attackers exploit the trust users place in vulnerable web applications. They insert malicious JavaScript or HTML code through user input fields, which is then executed when the application sends this input data as part of a webpage without proper validation to the user's browser.The risks associated with XSS include session hijacking, unauthorized alteration of application contents, redirection to other websites, and insertion of malicious codes or links. These risks are significant, as session hijacking can lead to the unauthorized acquisition of sensitive information.

SUBNET SCANNNER

A subnet scanner is a network utility used to identify active hosts within a specified range of IP addresses within a network subnet. It works by sending out ICMP (Internet Control Message Protocol) or ARP (Address Resolution Protocol) packets to each IP address in the subnet and listening for responses.The purpose of a subnet scanner is to discover devices connected to a network and gather information about them, such as IP addresses, MAC addresses, and sometimes open ports or services running on those devices. This information can be useful for network administrators to manage and monitor their network, troubleshoot connectivity issues, detect unauthorized devices or activity, and perform security assessments.Subnet scanners can be either passive or active. Passive subnet scanners listen to network traffic to detect hosts and gather information, while active subnet scanners actively send out probes to discover hosts. Active subnet scanners are typically faster but may generate more network traffic and potentially disrupt network operations.Overall, subnet scanners are valuable tools for network management and security, providing insights into the devices connected to a network and helping administrators maintain a secure and efficient network environment.

PORT SCANNER

A port scanner is a network utility used to identify open ports on a target system or network device. It works by sending network packets to a range of IP addresses and port numbers and analyzing the responses to determine which ports are open and accessible.The primary purpose of a port scanner is to assess the security posture of a network by identifying potential entry points that could be exploited by malicious actors. Open ports may indicate running services or applications that could be vulnerable to attacks if not properly secured or patched.Port scanners can be classified into two main categories: TCP (Transmission Control Protocol) port scanners and UDP (User Datagram Protocol) port scanners. TCP port scanners establish a full connection with the target port, while UDP port scanners send UDP packets and analyze responses to determine the status of UDP ports.

HTTP SNIFFER

An HTTP sniffer, also known as an HTTP packet sniffer or HTTP traffic analyzer, is a tool used to intercept and analyze HTTP (Hypertext Transfer Protocol) traffic between a client and a server within a network. It captures and inspects HTTP packets, allowing users to view the contents of HTTP requests and responses exchanged between the client (such as a web browser) and the server.While HTTP sniffers can be powerful tools for network analysis and troubleshooting, it's important to use them responsibly and ethically. Unauthorized interception and analysis of network traffic may violate privacy laws and regulations, so HTTP sniffing should only be performed on networks and systems where you have explicit permission to do so. Additionally, sensitive information such as usernames, passwords, and personal data transmitted over HTTP should be handled with care to prevent unauthorized access and data breaches.

TESTING

Testing is a crucial process aimed at identifying errors and ensuring quality assurance throughout the development and maintenance phases. It plays an integral role in the entire software lifecycle. During testing, the main objective is to verify that the specifications have been accurately and fully implemented into the design, while also confirming the correctness of the design itself.For instance, it's essential to ensure that the design is free from logic faults before the coding phase begins. Addressing faults in the design stage is significantly more cost-effective compared to rectifying them later in the development process. Detection of design faults can be achieved through methods such as inspection and walkthroughs. These processes help identify any potential issues early on, allowing for prompt resolution and a smoother development process overall.

UNIT TESTING

Unit testing is a critical phase in software development where the functional performance of individual modular components is verified. It centers on examining the smallest units of the software design, known as modules. During unit testing, the focus lies on thoroughly assessing the functionality of each module.To achieve comprehensive testing, whitebox testing techniques are extensively utilized. These techniques involve scrutinizing the internal logic and structure of the software components to ensure they perform as expected. By employing whitebox testing methods, developers can gain insights into the intricate details of the modules, enabling them to identify and rectify any potential issues early in the development process.

FUNCIONAL TESTING

Functional test cases entail executing the code with nominal input values for which the expected results are already known. Additionally, these test cases also involve assessing boundary values and special values, such as logically-related inputs, files containing identical elements, and empty files. There are three primary types of tests within functional testing:Positive testing involves validating the software's behavior when provided with valid input data. It aims to confirm that the system functions correctly under normal operating conditions.Negative testing examines the software's response to invalid or unexpected input data. It seeks to identify how the system handles errors, exceptions, and edge cases, ensuring robustness and resilience in adverse scenarios.Boundary testing focuses on evaluating the software's behavior at the boundaries of input ranges. It aims to uncover any issues related to boundary conditions, such as off-by-one errors or boundary-related failures, by testing inputs at or near the edges of acceptable ranges.By conducting these three types of tests comprehensively, functional testing helps ensure that the software meets its intended requirements, functions correctly under various conditions, and delivers a reliable user experience.

WHITE BOX TESTING

White box testing, also known as Glass box testing, focuses on evaluating the internal structure and logic of a software application. In this testing approach, testers have knowledge of the internal workings of the system, including its code, architecture, and implementation details.By understanding the specific functions that a product is designed to perform, testers can create test cases that aim to demonstrate the functionality of each function while also searching for errors or faults within them. White box testing involves examining the control flow and data flow within the application to ensure that all paths and conditions are tested thoroughly.Basis path testing is indeed a white box testing technique. It involves analyzing the control flow graph of the program to identify linearly independent paths, known as basis paths. Test cases are then designed to execute each basis path, ensuring that all statements and branches within the program are exercised during testing. This method helps uncover errors in the control flow and logic of the software, leading to improved reliability and quality.In summary, white box testing, also referred to as Glass box testing, involves testing the internal structure and logic of a software application to ensure its correctness and robustness. Basis path testing is one of the techniques used within white box testing to achieve thorough test coverage of the program's control flow.

CONCLUSION

Our comparative evaluation of various scanners once again demonstrated that scanners exhibit varying performance across different categories. Consequently, no scanner can be deemed universally effective in scanning all web vulnerabilities. The proposed scanner outlined above is particularly well-suited for beginners who may not be familiar with the intricate steps involved in scanning.Vulnerability scanning serves the crucial function of identifying security vulnerabilities within an organization. This process enables organizations to gain awareness of potential risks associated with their operating environment and take appropriate measures to mitigate them. Utilizing a vulnerability scanner offers the advantage of detecting known security weaknesses before

malicious attackers exploit them.In essence, vulnerability scanning and assessment provide organizations with valuable insights into their security posture, empowering them to proactively address vulnerabilities and enhance their overall cybersecurity defenses

REFERENCES

[1] Assem I. Mohaidat1, Dr Adnan Al-Helali2,"Web Vulnerability Scanning Tools: A Comprehensive Overview,Selection Guidance, and Cyber Security Recommendations"2024.

[2] A Subhangani*, B Anita Chaudhary,"Examination of Vulnerability Scanning Technologies"2022.

[3] Basan, M. (2023). "12 Types of Vulnerability Scans & When to Run Each." eSecurityPlanet. Retrieved from https://www.esecurityplanet.com/networks/types-of-vulnerability-scans/#port

[4] Binny George1, Jenu Maria Scaria1, Jobin B1, Praseetha VM2,"Web Application Security Scanner for Prevention and Protection against Vulnerabilities"2020.

[5] 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.

[6] 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.

[7] 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

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

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

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

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

[12] Sheetal Bairwa1, Bhawna Mewara2and Jyoti Gajrani3,"vulnerability scanners: a proactive approach to assess web application security"2020.

[13] Suliman Alazmi and Daniel Conte De Leon,A Systematic Literature Review on the Characteristics and Effectiveness of Web Application Vulnerability Scanners, 2022.