BindHost: Comprehensive Vulnerability ScannerTool

Yogesh Handge
Department of Computer Engineering
SCTR's Pune Institute of Computer Technology
Pune, India
yahandge@pict.edu

Jaee Bawdekar
Department of Computer Engineering
SCTR's Pune Institute of Computer Technology
Pune, India
jaeebawdekar2511@gmail.com

***Abstract*—The increasing dependence on web applications amplifies the importance of protecting them against cyber threats. Vulnerability scanners are of utmost importance in this process, as they periodically detect vulnerabilities within network infrastructures. This tool automates vulnerability detection, eliminating the need for manual intervention. This paper introduces a vulnerability scanner tool that determines probable threats to a web application by utilizing technologies such as header analysis, port scanning, and comparison against a database of known vulnerabilities(CVEs),thereby providing a robust solution**

***Index Terms*—Vulnerability Scanning, Header Analysis, Port Scanning, CVEs, Web Crawling,Socket Programming,Automation**

I. Vulnerability Scanning

Vulnerability scanning plays an important role in cybersecurity protocols, for examining networks,machines and applications for potential vulnerabilities and security breaches. Automated software tools take the lead in this process, thoroughly inspecting an organization's infrastructure to pinpoint known weaknesses, without the need for human intervention. These scans operate in a periodic manner, ensuring that emerging vulnerabilities are promptly detected and updated. After the detection of vulnerabilities, they are assessed based on severity ratings, ranging from critical to low, aiding organizations in prioritizing remediation efforts. Furthermore, compliance requirements, such as those outlined in PCI DSS, mandate regular vulnerability scanning to uphold data security standards, particularly for entities handling sensitive information like credit card data. Through this systematic approach, vulnerability scanning plays a pivotal role in fortifying organizational defenses against potential cyber threats. Utilizing vulnerability scanners before product deployment becomes paramount to identify and rectify potential security vulnerabilities, ensuring a robust and secure environment for both the organization and its users.

II. Developing Tool

A. Header Analysis

The initial phase involves analyzing the response headers. Python library Requests can be used to obtain these headers. It includes HTTP headers that are frequently neglected but are pivotal mechanisms for enhancing web security.

1) X-Frame-Options: It enhances protection against clickjacking attacks by restricting web pages from being rendered within frames. If X-Frame-Options headers are absent, web applications may be vulnerable to clickjacking attacks, enabling attackers to trick users into interacting with hidden or unwanted elements on compromised web pages.
2) Strict-Transport-Security (HSTS): This ensures secure connection via HTTPS. It instructs the browser to only interact with websites over HTTPS. This helps prevent attackers from downgrading the connection to HTTP, which is susceptible to interception and manipulation(Man-In-The-Middle threats)
3) X-Content-Type-Options: MIME Sniffing allows attackers to intercept server response by modifying content-type and injecting malicious code. To mitigate this risk, web servers can specify the X-Content-Type-Options header in the response with the value "nosniff" which instructs the browser to strictly adhere to specific content type.
4) X-XSS: It prevents the execution of malicious Javascript code. If enabled,the browser will stop loading the site after detection of Cross Site scripting.

B. Port Scanning

Port scanning is a critical aspect of vulnerability assessment and information gathering. It serves multiple purposes, including verifying the functionality of servers and identifying potential vulnerabilities that could be exploited by malicious actors..It is essential to scan your own network to spot potential security threats.Common standardized ports include 80 for HTTP, 443 for HTTPS, 21 for FTP, 22 for SSH, and
25 for SMTP

1) Implementation of Port Scanner:

Utilizing Python's socket library, a port scanner can be implemented to initiate connections with specified IP addresses across selected ports. This process involves the creation and use of network sockets, enabling the program to interact with other devices across a network. Multithreading is incorporated to boost efficiency, enabling simultaneous handling of multiple connection attempts by the scanner thereby scanning multiple ports simultaneously. Each thread is dedicated to scanning a distinct range of ports.If a connection is successful, the port is marked as open, indicating potential accessibility. Conversely, if the connection attempt fails, the port is considered closed. Through this iterative process, the port scanner identifies open ports, providing valuable insights into the target system's network configuration and potential vulnerabilities.

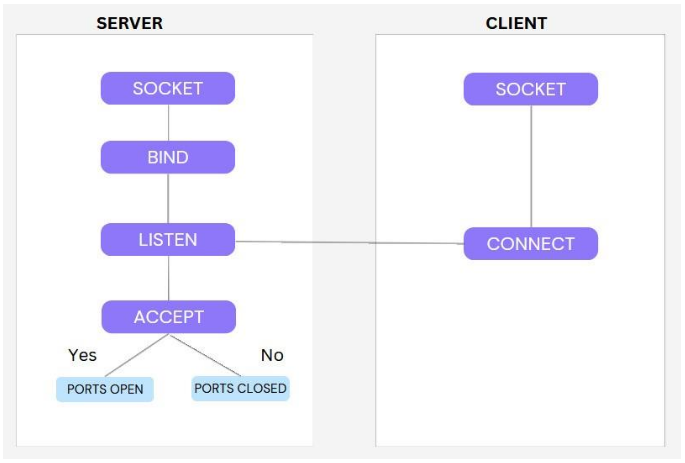

Fig. 1. Port Scanning using Socket Programmimg

C. *Web Crawling*

The goal of Web crawling is to map out a website, identifying all the possible entry points (like form fields, URL parameters, and cookies) that could be susceptible to security breaches.

1) Mapping a Website and identifying potential input fields: Crawler starts with the specified URL ,identifies all the hyperlinks on the page and adds them to the list of URLs to visit next. This recursive process allows it to discover content systematically. During the crawling process, the crawler examines the content of each page to understand its structure, content, and how it interacts with users. This involves analyzing HTML code, JavaScript interactions and form structures. The critical Part of this process is identifying points where the user input is accepted. This includes not only visible form fields but also hidden fields, cookies that track user sessions, and URL parameters.

2) Testing For vulnerabilities: The key to testing input fields for vulnerabilities lies in crafting malicious inputs, known as payloads. For SQL Injection, payloads might attempt to manipulate database queries which can lead to gaining access over database, while for Cross-Site Scripting (XSS), they aim to execute scripts in the context of the user's browser. Injection of these payloads into identified entry points plays pivotal role in the testing which might trigger specific vulnerabilities. After payload injection, the scanner analyzes the responses from the web application. An unexpected response, such as an error message, a different output than normally seen, or even a successful execution of the payload's intent (e.g. unauthorized data retrieval or alert box appearance for XSS), can indicate a vulnerability. To ensure thorough coverage, the web crawler must execute JavaScript and parse dynamically loaded content to identify additional input points hidden within dynamically generated pages.

D. *Comparison with CVE*

1) Incorporating a database of known CVE (Common Vulnerabilities and Exposures): This database contains detailed information about vulnerabilities discovered in various software and hardware products. During a vulnerability scan, the scanner compares the versions of installed software on the target system against the CVE database to identify any matches. If a software version is associated with a CVE entry in the database, it indicates a potential vulnerability. Each CVE entry provides a standardized identifier along with detailed information about the vulnerability, including its severity, impact, affected software versions, and possible mitigations.The severity of vulnerability can be predicted easily considering CVEs.

III. Conclusion

In the realm of cybersecurity, BindHost stand as indispensable tool for organizations seeking to fortify their digital defenses against an ever-expanding array of threats. Through the integration of advanced techniques such as security header analysis, port scanning using socket programming and multithreading, web crawling, and comparison with Common Vulnerabilities and Exposures (CVE), BindHost provides a comprehensive approach to identifying and mitigating security vulnerabilities within IT infrastructures thereby enhancing their resilience to cyber threats and safeguard sensitive, confidential data.

References

[1] Yudha, Fietyata Panji, Andi Ar, Laksono Ramadhani, Erika. (2019). Web Crawling Technique for Vulnerability Assessment on Web.
[2] provides a standardized identifier along with detailed information about the vulnerability, including its severity, impact, affected software versions, and possible mitigations.
[3] Goel, Jai Mehtre, Babu. (2015). Vulnerability Assessment Penetration Testing as a Cyber Defence Technology. Procedia Computer Science. 57. 710-715. 10.1016/j.procs.2015.07.458. .
[4] Wang, Liwei Abbas, Robert Almansour, Fahad Gaba, Gurjot Alroobaea, Roobaea Masud, Mehedi. (2021). An empirical study on vulnerability assessment and penetration detection for highly sensitive networks. Journal of Intelligent Systems. 30. 592-603. 10.1515/jisys-2020-0145.
[5] Petkova, Lilyana. (2019). HTTP SECURITY HEADERS.
[6] Vivo, Marco Ke, Le Isern, Germinal Vivo, Gabriela. (1999). A review of port scanning techniques. Computer Communication Review. 29. 41-48. 10.1145/505733.505737.