STREAMLINED SECURITY: MERGING NETWORK SCANNING, SQL INJECTION AND XSS VULNERABILITY DETECTION

[1]Sravan A Surendran, [2]Jeevan Paul Chitran, [3]Krishnapriya Pradeep, [4]Nurjahan V A
[1]Student, [2]Student, [3]Student, [4]Professor
Department Of Artificial Intelligence and Cyber Security
Ilahia College Of Engineering and Technology, Muvattupuzha, Kerala, India.

Abstract - To counter increasing sophistication in cyber threats, this research proposes a lightweight security tool incorporating network scanning, SQL Injection (SQLi) detection, and Cross-Site Scripting (XSS) vulnerability analysis. Conventional security analysis approaches work well but tend to be highly expertise- and time-consuming-oriented. The new tool detects vulnerabilities automatically, utilizes Python with Tkinter for intuitive interaction, and incorporates the Nmap utility for effective network scanning. Through integrating these capabilities, the tool enables security experts and beginners alike to perform detailed assessments with enhanced accuracy and less effort. Experimental findings identify the efficiency of the tool in detecting vulnerabilities and enhancing assessment effectiveness. This study emphasizes the importance of integrated security tools in increasing cyber resilience.

Index Terms: **Vulnerability Detection, Network Scanning, SQL Injection, XSS, Python, Tkinter, Nmap.**

I. INTRODUCTION (HEADING 1)

The quick growth of digital networks and internet services has delivered a variety of advantages to individuals and organizations. Nonetheless, this added connectivity has also raised the level of vulnerability to cyberattacks, which makes effective security solutions indispensable. Cyber attackers use weaknesses in systems to acquire unauthorized access, pilfer confidential information, or disable business operations. Therefore, vulnerability assessment and penetration testing have emerged as indispensable components in the protection of digital assets.Cyberattacks tend to hit vulnerabilities like older software, servers with incorrect configurations, or buggy web application code. These can serve as the entry point for attackers to breach systems. Attacks like SQL Injection (SQLi) and Cross-Site Scripting (XSS) are regularly used, which pose substantial threats to data integrity and the privacy of users. Organizations need to detect such vulnerabilities early on to control risks effectively. Classic penetration testing entails the use of manual examination of systems, a technique that requires qualified staff and a significant amount of time commitment. Although manual approaches can reveal intricate vulnerabilities, they tend to be less effective for low-resource organizations. Automated security solutions have appeared to fill this gap by providing quicker and more scalable options.

This study proposes a novel security tool that integrates three vital security auditing techniques: network scanning, SQL Injection (SQLi) discovery, and Cross-Site Scripting (XSS) vulnerability detection. Through the use of Python for its extensibility and Tkinter for its friendly graphical user interface, the tool presents an easy-to-use interface for both seasoned experts and beginners in the field of cybersecurity. The integration of Nmap strengthens network scanning capabilities, with the provision of extensive information about open ports, active services, and possible security loopholes. Custom algorithms for SQLi and XSS detection strengthen the tool further in detecting vulnerabilities within web applications with greater efficiency.The envisioned tool is created to reduce human intervention, enhancing speed and accuracy in assessment. Through the automation of these security processes, organizations are able to actively detect and eliminate risks before they are compromised. This study underscores the necessity of integrating automation with ease-of-use features in developing an efficient security solution to address various organizational requirements.

II. RELATED WORKS

A number of tools are already developed to cover some parts of vulnerability detection and penetration testing. Nmap, Nikto, and OpenVAS are popularly known tools that have very strong scanning capacity.

Nmap is a robust network scanner that effectively scans open ports, services, and system information. Although Nmap is best at network mapping and reconnaissance, it does not have inherent capabilities to exploit vulnerabilities directly, confining its function to early-stage assessment activities.

Nikto is effective in detecting and reporting vulnerabilities in web servers, specifically in identifying outdated software, misconfigurations, and security vulnerabilities. Nike's main limitation is that it cannot execute high-level exploitation tasks, and it needs to be used manually for thorough security testing..

OpenVAS (Open Vulnerability Assessment Scanner) is a robust vulnerability scanning tool that detects security vulnerabilities on networks and systems. Even though OpenVAS has a very strong scanning capability, it can be heavy on resources and may generate a large number of false positives, requiring further filtering to effectively segregate actual threats.

Metasploit, the industry benchmark penetration testing framework, contains full-featured exploit modules and delivery mechanisms for payloads. Although very effective, Metasploit's complexity tends to be challenging for users with little cybersecurity expertise, necessitating extensive experience to optimize.

SQLmap is an expert tool intended for automatic SQL Injection detection and exploitation. Though SQLmap is very effective in detecting SQLi vulnerabilities, it is not capable of giving wide coverage for network scanning or other web-related threats, thus restraining its usability for extensive security auditing.

OWASP ZAP (Zed Attack Proxy) is a widely used web application security tool that offers automated scanning and manual testing capabilities. While OWASP ZAP is good at identifying common vulnerabilities, it might need extra configuration to customize scans for sophisticated applications, which makes it less user-friendly for beginners.

Although these tools are excellent in their respective fields, they usually work independently, and security experts have to depend on numerous tools for thorough security analysis. Fragmentation enhances complexity, time requirements, and chances of oversight. The tool in question remedies this shortcoming by combining network scanning, SQL Injection detection, and XSS vulnerability detection into one comprehensive framework. Through the integration of these features, the tool intends to increase the efficiency of security assessment, eliminate manual labor, and enhance vulnerability identification accuracy for varied environments.

III. PROPOSED SYSTEM

The system, as proposed, seeks to provide an end-to-end security scan solution by bringing together network scanning, SQL Injection vulnerability detection, and XSS vulnerability detection within one automated framework. This combined process is intended to overcome the deficiencies seen in standalone security tools so that efficiency and accuracy in identifying vulnerabilities are enhanced.

The system uses Nmap for network scanning to detect active hosts, open ports, and service information. Moreover, SQLmap is integrated to automate detection of SQL Injection vulnerabilities to ensure proper scanning of web applications for database security vulnerabilities. For Cross-Site Scripting (XSS) vulnerability detection, custom detection methods are integrated to detect script injection threats that may jeopardize web-based applications.

The platform is developed using Python with Tkinter for the graphical user interface, promoting ease of use and usability. The combined platform reduces the dependency on manual setup through the option of automated scanning, live analysis reports, and actionable guidance for reducing found threats. By reducing the security scanning process complexity, the system provides increased usability even for individuals lacking extensive technical background.

Major characteristics of the proposed system are:

- Automation: Minimizes human intervention by scanning and detecting vulnerabilities automatically.
- Detailed Analysis: Combines various security methods to deliver detailed analyses.
- User Interface: Provides a straightforward and easy-to-use GUI for easy operation.
- Real-Time Reporting: Automatically provides immediate feedback and recommended remedies for detected threats.

This combined solution is designed to enhance cybersecurity preparedness by streamlining the assessment process, minimizing the necessity for multiple standalone tools, and providing actionable information for enhancing system defenses.

IV. SOLUTION METHODOLOGY

The suggested automated penetration testing tool is based on a systematic system architecture that is intended to effectively detect vulnerabilities, scan security threats, and produce detailed reports. The system consists of a number of important components, each with a vital function in the penetration testing process:

1. User Interface (UI) The UI is the point of entry for users to access the system. Users may enter key parameters like target URLs, attack types (SQL Injection, Cross-Site Scripting), and scanning options. The interface provides ease of navigation for both technical cybersecurity professionals and those with minimal technical skills.

2. Integration of Nmap for Network Scanning: Nmap is used by the tool for automated network scanning. It uses this element to effectively discover open ports, active services, and possible vulnerabilities in the target system. Using Nmap ensures the system efficiently maps the network environment, allowing for more penetration testing accuracy.

3. Automated SQL Injection and XSS Detection Module: This module actively replicates attack scenarios against SQL Injection and Cross-Site Scripting vulnerabilities. By generating and sending targeted payloads, the software replicates real-world penetration testing environments. The automation allows for better vulnerability detection with less manual effort.

4. Vulnerability Analysis Engine This engine computes data gathered from Nmap scans and results from injection attacks. Through pattern matching and rule-based scanning, it identifies vulnerabilities with ease, rates their severity, and prioritizes them for their correction. The reduction of false positives by the system guarantees the accuracy of issues identified.
5. Report Generation Module To enhance the ease of use of the results of the assessment, the software includes a report generator that brings together discovered vulnerabilities, their effects, and proposed solutions. The reports are formatted to offer explicit insights to technical teams as well as non-technical stakeholders in order to communicate actionable information effectively.
6. Technology Stack The system is built on Python, utilizing libraries for network scanning, data processing, and security testing. The graphical user interface is implemented with Tkinter for better usability. The tool also incorporates OWASP security guidelines to ensure compliance with industry best practices for effective penetration testing.

With its blend of automation, effective scan techniques, and extensive reporting, this solution maximizes penetration test efficiency with ease while providing ease of use accessible to a diversity of skill sets.

The overall system design is shown in Fig 4.1, which shows the interaction between each of the components to provide an effective penetration testing solution.

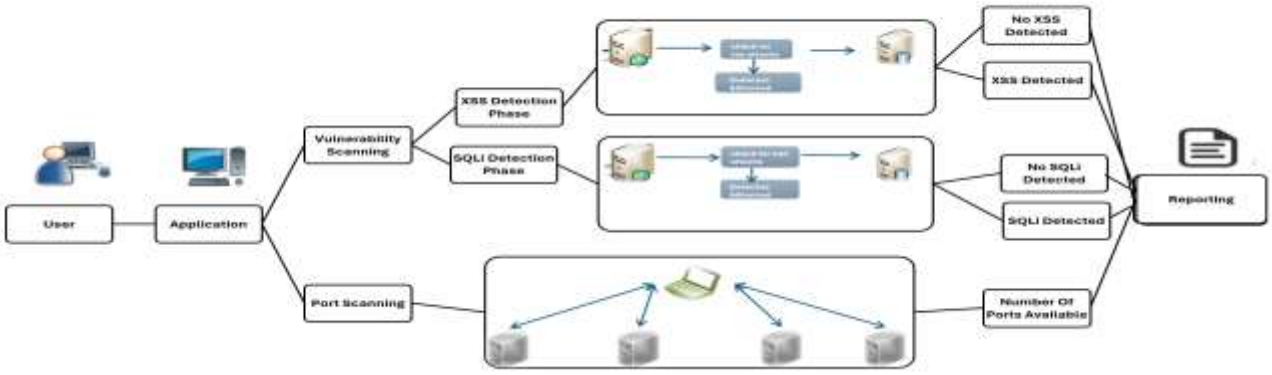

Fig. 4.1: System Architecture

A. Algorithm

BEGIN

Step 1: Initialization

- Initialize system components
- Load Nmap integration module
- Load SQL Injection and XSS detection module
- Load report generation engine

Step 2: User Input Collection

- DISPLAY "Enter Target URL:"
- READ target_url
- DISPLAY "Enter Attack Types (e.g., SQL Injection, XSS):"
- READ attack_types
- DISPLAY "Enter Custom Payloads (optional, comma-separated):"
- READ custom_payloads

Step 3: Network Scanning

- EXECUTE Nmap scan on target_url
- STORE scan results in scan_data

Step 4: Attack Simulation FOR EACH attack_type IN attack_types DO FOR EACH payload IN custom_payloads DO response = SEND_HTTP_REQUEST(target_url, attack_type, payload) STORE response in response_list END FOR END FOR

Step 5: Vulnerability Analysis

- vulnerabilities = [] FOR EACH entry IN scan_data DO IF IDENTIFY_WEAKNESS(entry) THEN ADD entry TO vulnerabilities END IF END FOR FOR EACH response IN response_list DO IF DETECT_VULNERABILITY(response) THEN ADD response TO vulnerabilities END IF END FOR

Step 6: Report Generation IF vulnerabilities IS NOT EMPTY THEN report = GENERATE_REPORT(vulnerabilities) ELSE report = "No vulnerabilities detected." END IF

Step 7: Report Formatting and Saving

- formatted_report = FORMAT_REPORT(report)
- SAVE formatted_report AS "Pentest_Report.pdf"

END

V. RESULTS AND DISCUSSIONS

The outcomes of results from the implemented tests are worthy insights into efficiency and performance and effectiveness of all cybersecurity tools involved. Every method of scanning was examined in light of its reach, effectiveness, and precision while further tests were done comparing easiness of installments, users' interface, and detection sensitivity. The findings presented below have significant points based on strengths and benefits of this suggested system relative to traditional methods.

1. Quick Scan Analysis

The Quick Scan picture depicts a quick scanning procedure intended to determine open ports in a target system within short time. This process prioritizes scanning most heavily used ports to ensure that key vulnerabilities are quickly detected. The effectiveness of the scan allows it to be suitable for initial assessments where speed is paramount. Although it might not offer comprehensive information such as with deeper scans, its main benefit is in locating major security loopholes quickly. Results from the scan can be used to help security analysts in prioritizing main areas for detailed examination.

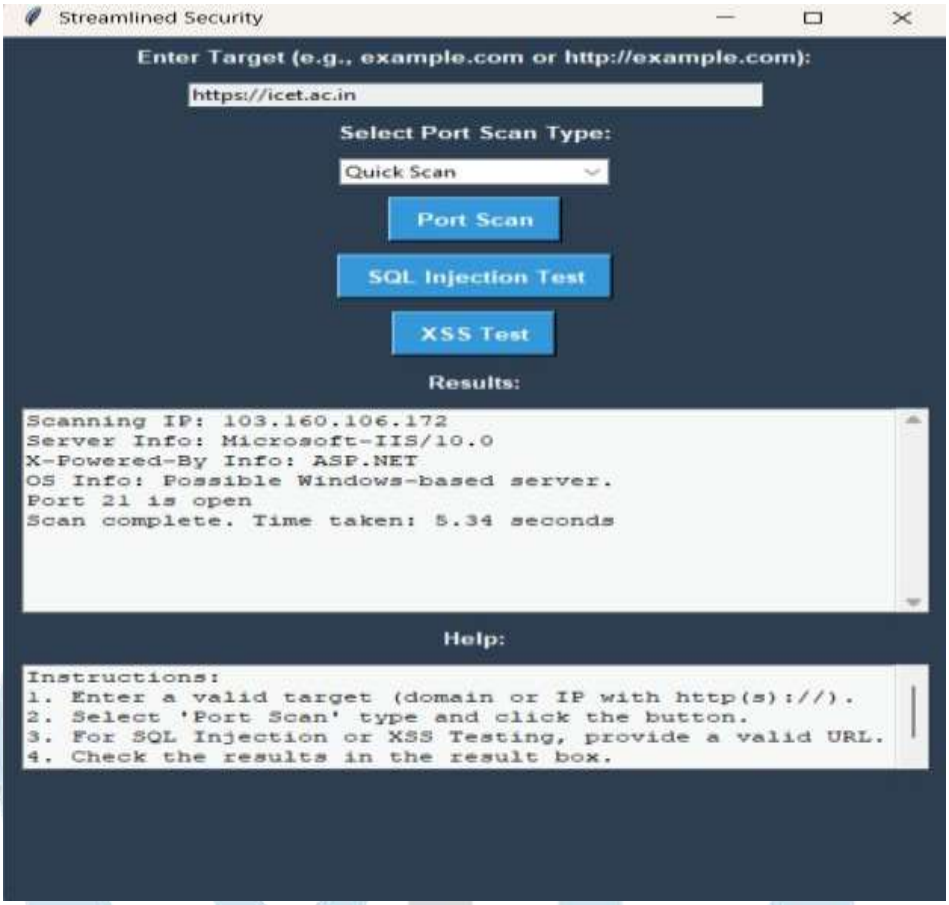

Fig. 5.1: Quick Scan

2. Common Scan Analysis

The Common Scan image is a scanning technique aimed at commonly exploited ports. The method presents a balance between speed and depth by targeting ports most likely to be attacked. By focusing on these vulnerable entry points, the Common Scan effectively probes for possible vulnerabilities without the time demands of full-scale scanning. It assists in the discovery of weaknesses in critical services like HTTP (port 80), HTTPS (port 443), and SSH (port 22) that are frequently exploited in actual attacks. The approach is effective for proactive security scans as well as for ongoing system monitoring.

Fig. 5.2: Common Scan

3. ***Full Scan Analysis***

The Full Scan image shows a thorough scanning process that scans all 65,535 TCP ports of a particular system. This intensive approach is essential to detect less prevalent vulnerabilities that are not picked up by quicker scans. Even though this approach requires substantially more resources and time, it guarantees a more in-depth study, unearthing concealed security threats. Full scans are especially effective in advanced security scans where hidden threats may result in critical breaches. The comprehensive information derived from full scans is precious in developing effective security measures.

Fig. 5.3: Full Scan

4. SQL Injection Vulnerability Analysis

The SQL Injection Test screenshot displays an application that would determine database vulnerability due to wrong input validation of web applications. The interface draws attention to the format of URL under test for identifying possible points of SQL injection. This testing aims at pinpointing loopholes where malicious SQL queries can get inserted to play with or draw out sensitive data from the database. SQL injection attacks have been one of the most common web application threats, and this test is important in countering risks by detecting potential database entry points for exploitation. User guidance in the interface increases usability such that testers can carry out the assessment easily.

Fig. 5.4: SQL Injection Test

5. Cross-Site Scripting (XSS) Vulnerability Analysis

The XSS Test image captures a focused device to detect cross-site scripting threats. The design points out tested URL parameters under potential XSS vulnerabilities. XSS attack is caused when there are malicious scripts injected in legitimate sites to tamper with user data and session tokens. The tool features will help security professionals and developers find insecure web input points ahead of time and create preventive mitigations. The addition of a well-defined interface along with step-by-step instructions makes the test more accessible to both novice and experienced security analysts.

Fig. 5.5: XSS Test

Analysis

The comparison assesses the performance of different cybersecurity tools like Nmap, Burp Suite, and SQLmap with the Proposed System. The assessment is done considering major performance factors like User Interface, Feature Integration, Installation Ease, Scanning Speed, and Detection Accuracy. From these parameters, the assessment identifies the advantages and betterment provided by the Proposed System in improving overall security assessment mechanisms.

The Line Graph indicates that the Proposed System is better in User Interface design and Feature Integration. Since it has a higher score for UI design, the Proposed System provides an easier and more intuitive experience, which minimizes complexity for security professionals. Better feature integration provides greater security coverage through the integration of multiple scanning techniques, which makes the Proposed System more efficient and flexible.

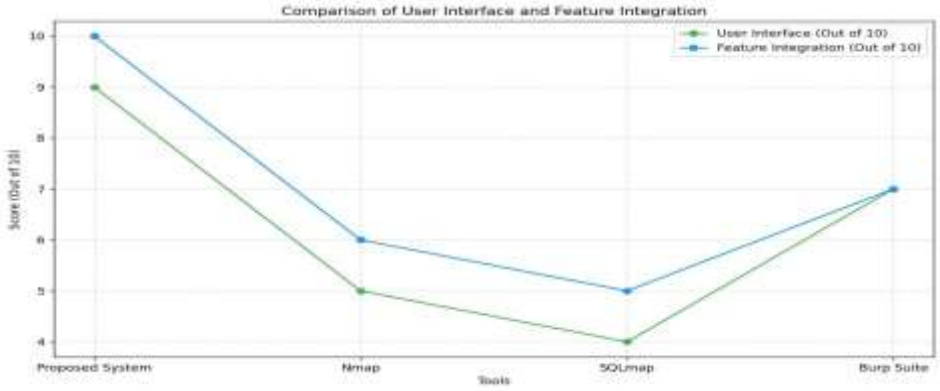

Fig. 5.6: Performance Graph Line

Pie Chart indicates the simplicity of installation with respect to different tools. The Proposed System reveals a simpler method of installation with less technical intricacy and speedy deployment. The improvement increases ease of use, particularly for those with less technical experience, hence increasing the flexibility of the tool to fit any security environment.

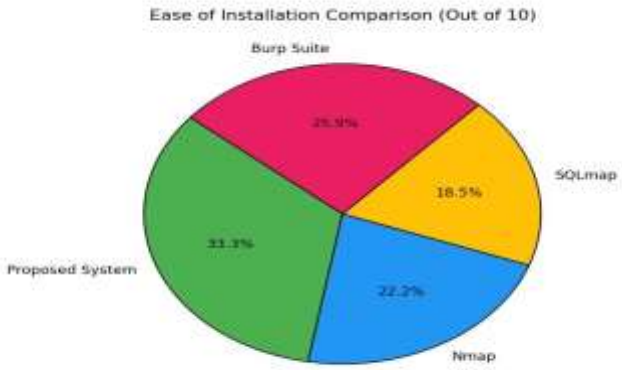

Fig. 5.7: Performance Pie Chart

The Bar Chart indicates that the Proposed System surpasses traditional tools in scanning speed while still having a high rate of detection accuracy. This is important in cybersecurity because fast assessments are essential to effectively counter threats. Although tools such as Nmap and Burp Suite have high accuracy, the Proposed System has the same level of precision but with enhanced efficiency, which makes it ideal for time-critical security assessments.

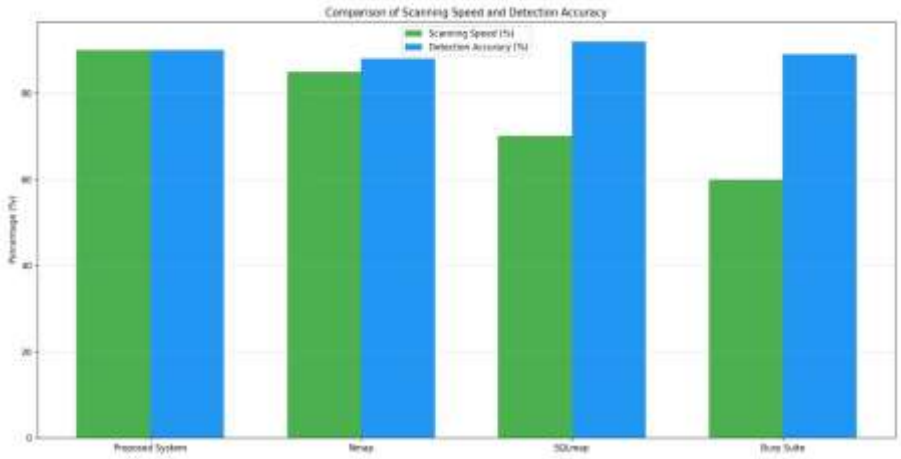

Fig. 5.8: Performance Bar Chart

VI. CONCLUSION AND FUTURE SCOPE

The designed penetration testing tool has shown remarkable performance in detecting web application vulnerabilities with great accuracy and efficiency. Its upgraded user interface, efficient navigation, and better reporting facilities make it easier to use, rendering it appropriate for both seasoned security professionals and users with minimal technical competence. The scalability of the tool ensures the stable functioning even under heavy workload conditions, making it even more reliable within enterprise ecosystems.

The inclusion of editable attack templates brings flexibility in terms of users being able to customize their security testing to suit certain testing needs. The fact that the tool can produce detailed reports also helps ensure proper documentation of findings, allowing security teams to effectively prioritize and fix vulnerabilities.

In the future, there are various prospects for upgrading the tool's functionality. Subsequent updates might involve adding advanced evasion mechanisms to evade current security defenses, e.g., Web Application Firewalls (WAFs). Adding real-time monitoring and alert systems would enhance the tool's reactivity further by allowing instantaneous threat detection and blocking. Extending support for cloud-based infrastructure security audits and compatibility with upcoming technologies will enhance the tool's reach and usability.
Through the incorporation of these enhancements, the envisioned tool can become a full-fledged security solution that can address the dynamic nature of contemporary cybersecurity environments.

References

[1] Joshi Padma N., Dr. N. Ravishankar, Dr. M. B. Raju, and N.CH. Ravi, “ContemplatingSecurity of HTTP from SQL Injection and Cross Script,” 2017.
[2] Shreya Chowdhury, Aakansh Nandi, Miran Ahmad, Aadish Jain, and Mohandas Pawar,“A Comprehensive Survey for Detection and Prevention of SQL Injection,” 2021.
[3] Limei Ma, Dongmei Zhao, Yijun Gao, and Chen Zhao, “Research on SQL InjectionAttack and Prevention Technology Based on Web,” 2020.
[4] Bin Wang, Lu Liu, Feng Li, Jianye Zhang, Tao Chen, and Zhenwan Zou, “Research onWeb Application Security Vulnerability Scanning Technology,” 2018.
[5] Joshi Padma N., Dr. N. Ravishankar, Dr. M. B. Raju, and N.CH. Ravi, “EncounteringSQL Injection in Web Applications,” 2019.
[6] Beaulah A. Navamani, Chuan Yue, and Xiaobo Zhou, “An Analysis of Open Ports andPort Pairs in EC2 Instances,” 2022.
[7] Emad Eldin Mohamed, Adel Ben Mnaoue, and Ezedin Barka, “PSCAN: A Port ScanningNetwork Covert Channel,” 2020.
[8] Muath Obaidat, Joseph Brown, and Abdullah Al Hayajneh, “Web Browser ExtensionUser- Script XSS Vulnerabilities,” 2019.
[9] Mehr-u-Nisaa and Kashif Kifayat, “Detection of Slow Port Scanning Attacks,” 2021.
[10] Chengcheng Lv, Long Zhang, Fanping Zeng, and Jian Zhang, “Adaptive RandomTesting for XSS Vulnerability,” 2018.
[11] Agung Wijayanto, Ema Utami, and Agung Budi Prasetio, “Analysis of VulnerabilityWebserver Office Management of Information and Documentation Diskominfo UsingOWASP Scanner,” 2017.
[12] Xin-Yu Hou, Xiao-Lin Zhao, Mei-Jing Wu, Rui Ma, and Yu-Peng Chen, “A DynamicDetection Technique for XSS Vulnerabilities,” 2020.
[13] Keyur Patel, “A Survey on Vulnerability Assessment & Penetration Testing for SecureCommunication,” 2019.
[14] Dimitris E. Simos, Bernhard Garn, Jovan Zivanovic, and Manuel Leithner, “PracticalCombinatorial Testing for XSS Detection Using Locally Optimized Attack Models,” 2018.
[15] Ajjarapu Kusuma Priyanka and Siddemsetty Sai Smruthi, “Web ApplicationVulnerabilities: Exploitation and Prevention,” 2020.
[16] Mujahid Shah, Sheeraz Ahmed, Khalid Saeed, Muhammad Junaid, Hamayun Khan,and Ata-ur- Rehman, “Penetration Testing Active Reconnaissance Phase – Optimized PortScanning with Nmap Tool,” 2021.
[17] Santi Pattanavichai, “Comparison for Network Security Scanner Tools Between GFILanGuard and Microsoft Baseline Security Analyzer (MBSA),” 2018.
[18] Akira Tanaka, Chansu Han, and Takeshi Takahashi, “Detecting Coordinated Internet-Wide Scanning by TCP/IP Header Fingerprint,” 2019.
[19] Bruce Hartpence and Andres Kwasinski, “Combating TCP Port Scan Attacks UsingSequential Neural Networks,” 2020.
[20] Rodney R. Rohrmann, Vincent J. Ercolani, and Dr. Mark W. Patton, “Large Scale PortScanning Through Tor Using Parallel Nmap Scans to Scan Large Portions of the IPv4Range,” 2022.