WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Vulnerability Scanner by Using Kali Linux Tools

[1] Ganapathi Kinjarapu, [2] G. Jayaram, [3] Kolla Siva Sai, [4] H. Nani, [5] Mr.S.Chanti

[1] Computer Science and Engineering GMR Institute of Technology Rajam,India, kinjarapuganapathi9493@gmail.com
[2] Computer Science and Engineering GMR Institute of Technology Rajam,India, 22341a0569@gmrit.edu.in
[3] Computer Science and Engineering GMR Institute of Technology Rajam,India 22341a0587@gmrit.edu.in
[4] Computer Science and Engineering GMR Institute of Technology Rajam,India, 22341a0571@gmrit.edu.in
[5] Computer Science and Engineering GMR Institute of Technology Rajam,India Chanti.s@gmrit.edu.in

ABSTRACT:

A vulnerability scanner is an essential cybersecurity tool used to scan, analyze, and prioritize vulnerabilities in systems, networks, and applications. This tool automatically checks for known misconfigurations and security flaws exploitable by adversaries. It can be seen in how scanners are programmed to recognize the system components against a threat database or based on algorithms, which could create a hole for malicious users to penetrate the network. They are generally used for scanning network devices, servers, web applications, cloud environments, and containers. There are various types of vulnerability scanners, including network scanners, web application scanners, host-based scanners, and cloud security tools, each focused on different layers of infrastructure. These tools help reduce the burden on security teams by automating vulnerability assessments so that they can focus on remediation and mitigation. Modern vulnerability scanners have features like automated scanning, risk prioritization, detailed reporting, and integration with other security tools. Nevertheless, they do have limitations as they may throw false positives or fail to catch newly discovered vulnerabilities. Regular use of vulnerability scanners, along with a comprehensive vulnerability management program, is essential to maintaining the security posture of an organization and to defend against cyber threats

Keywords— Vulnerability Scanner, Cybersecurity, Network Security, Web Application Security, Risk Assessment, Threat Detection, Security Flaws, Vulnerability Management, Automated Scanning, Compliance Checks

Introduction

To prevent such threats, vulnerability scanning has become a critical part of modern cybersecurity. It helps identify flaws before they are exploited by attackers, allowing organizations to fix them in time and strengthen their defenses. In this project, we focused specifically on two types of scanning: web application scanning and network scanning.

Web application scanning looks for vulnerabilities in websites, such as exposed admin panels, insecure plugins, SQL injection, or cross-site scripting (XSS). Network scanning, on the other hand, focuses on discovering open ports, active services, and misconfigurations that could be exploited through the network. Both are essential for understanding and securing the digital environment of any organization.

To achieve this, we used several well-known tools available in Kali Linux, each with its own strengths. Nmap was used for network scanning to identify open ports and services. Nikto helped scan web servers for outdated software and insecure configurations. Nuclei was included for fast, template-based vulnerability scanning across various web technologies. Wappalyzer and WhatWeb were used for fingerprinting web applications and detecting the underlying technologies and platforms used by a website.

These tools are powerful but usually require terminal knowledge to operate. To make them easier to use, we developed a web-based interface that allows users to simply enter a URL or IP address and run a scan with a single click. The system then runs the appropriate tool in the background and displays the results in a clean, user-friendly format.

In today's digital age, web applications and network systems are the backbone of most organizations. While they enable fast communication, data sharing, and remote operations, they also open up opportunities for cyberattacks. Attackers often target weaknesses such as outdated software, open ports, or misconfigured services. If these security gaps are not detected and fixed early, they can lead to serious problems like data breaches, service downtime, or unauthorized access.

The goal of this project is to make web and network vulnerability scanning more accessible for students, developers, and system administrators. By combining professional cybersecurity tools with a simple interface, we aim to make regular security checks easier to perform and encourage good security practices, even for those without deep technical expertise.

LITERATURE REVIEW

Odion et al. (2023) proposed VulScan**,** a web-based vulnerability scanner integrating multiple scanning tools under a unified interface. The tool aims to simplify vulnerability detection in web applications by automating scan processes and consolidating reports. VulScan includes modules that assess both frontend and backend vulnerabilities, offering support for technologies such as PHP, JavaScript, and Python-based frameworks. The system emphasizes scalability, allowing organizations to deploy it across large environments without performance degradation. The researchers validated the effectiveness of VulScan through practical implementation and comparison with existing tools like Nikto and OWASP ZAP.The purpose of this article is to compare passive and active vulnerability scanning methodologies. It explains the strengths and weaknesses of each approach, providing guidance on their appropriate use cases. The study helps organizations choose the right scanning strategy to improve cybersecurity while balancing intrusiveness and effectiveness [1]. RiskOptics (2022) differentiated between passive and active scanning**,** highlighting their respective roles in cybersecurity. Passive scanning monitors traffic without injecting data, making it ideal for sensitive environments like medical or industrial systems. Active scanning, on the other hand, simulates real- world attack behavior to probe for vulnerabilities, providing deeper visibility. The article recommends using passive scans continuously, and supplementing with active scans periodically to avoid blind spots. It also highlights the importance of context-aware scanning that adapts to evolving infrastructure and threats [2]. The National Cyber Security Centre (NCSC, 2021) provided official guidance on vulnerability scanning tools and services, with a focus on organizational readiness. The guide outlines key features organizations should consider, such as real-time alerts, scalability, compliance reporting, and vendor support. NCSC also categorizes scanning services into internal, external, and third-party offerings, aiding stakeholders in choosing the right approach. Moreover, it stresses the significance of combining automated scans with manual penetration testing to identify business logic flaws. Their work encourages the use of scanning as a proactive, routine component of an organization's risk management strategy [3]. Pandey and Chaudhary (2022) provided a detailed explanation of vulnerability scanning concepts and tools, tailored for both academia and practitioners. They discussed the architecture of scanners, typically involving a vulnerability database, scanning engine, and report generator. Their work evaluates tools like Nessus, OpenVAS, and Acunetix in terms of detection accuracy and usability. The paper also introduces the concept of hybrid scanning that leverages both signature- based and behavioral analysis. It concludes with a roadmap for improving scanner accuracy using AI and machine learning [4]. RSI Security (2023) identified seven major types of vulnerability scanners, helping organizations to understand the specific roles each type plays in the security lifecycle. For instance, network-based scanners detect open ports and weak configurations, while application scanners assess input validation flaws. The article breaks down when and where to deploy each scanner, advocating a layered approach for robust coverage. It also notes the importance of integrating scanning into CI/CD pipelines to catch vulnerabilities during development. RSI emphasizes that no single scanner is sufficient, recommending the use of specialized tools based on system criticality [5]. Basan (2023) described twelve types of vulnerability scans, extending RSI's classification with additional practical insights. He includes scans like authenticated vs unauthenticated, stealth scans, and compliance-specific scans (e.g., for PCI-DSS or HIPAA). The article highlights how timing of scans—real-time, scheduled, or event-driven—can impact detection efficiency. Basan provides use-case examples for each type, such as using port scans during onboarding of new devices and configuration scans after policy updates. The work also discusses how scanning results can inform incident response planning and long-term hardening strategies [6]. Deeptha et al. (2022) presented a comprehensive study on web application vulnerabilities and the various methods used to detect and prevent them. They examined common attack vectors such as cross-site scripting (XSS), cross-site request forgery (CSRF), and directory traversal, analyzing how they exploit web application logic. The study covers both static code analysis (pre-deployment) and dynamic analysis (post- deployment) approaches. It further recommends the adoption of secure coding practices and regular code reviews to reduce vulnerability risk. The authors also propose a layered security model that includes firewalls, WAFs, and regular scanning as core defenses [7]. Tenable (2023) developed the Tenable Vulnerability Management FedRAMP Moderate User Guide, which outlines how their scanning platform complies with federal risk and authorization standards. The document describes procedures for configuring scans, managing discovered vulnerabilities, and generating compliance reports for cloud environments. It emphasizes automation, scalability, and centralized risk tracking for large enterprises and government agencies. The guide also outlines integrations with asset management systems, incident response workflows, and ticketing platforms. This resource provides practical insights into how vulnerability management aligns with regulatory requirements [8]. Wang et al. (2020) discussed advances in web application security, focusing on both detection and prevention of vulnerabilities. The study categorizes approaches into static, dynamic, and hybrid techniques, analyzing their pros and cons. It also addresses the integration of AI and machine learning in improving detection accuracy and reducing false positives. The authors stress the importance of securing both client-side and server-side code to prevent injection and authentication attacks. Their findings provide a solid foundation for next- generation security frameworks that incorporate predictive and adaptive capabilities [9]**.** Rathod et al. (2021) proposed an automatic vulnerability scanner enhanced with firewall techniques for web applications. Their system detects suspicious patterns and automatically updates firewall rules to block potential threats in real time. The paper describes a feedback loop where scan results inform firewall configuration, improving system resilience. This integration reduces manual intervention and provides a proactive security layer. The research supports the idea of merging scanning with live protection mechanisms to mitigate threats dynamically [10]. Yulimanton et al. (2021) analyzed web security vulnerabilities through the lens of common attack types such as SQL injection and XSS. Their study explains how poor coding practices and weak input validation contribute to these vulnerabilities. They emphasize the need for continuous testing throughout the software development lifecycle (SDLC). The authors also advocate for integrating security education into developer training to reduce vulnerabilities at the source. Their work provides a practical perspective on the human and technical aspects of web application security [11]. Alazmi & Conte de Leon (2020) provided a comparative analysis of web application vulnerability scanners, evaluating their performance, accuracy, and scanning depth. The study tested several popular scanners on a standardized

testbed to measure detection rates and false positives. It identified key features that enhance scanner effectiveness, such as customizable rulesets, plugin support, and automation APIs. The researchers also discussed how scan results can be used in broader risk assessment processes. Their contribution helps users make informed choices when selecting scanning tools for different environments [12]. Bairwa, Mewara & Gajrani (2021) discussed vulnerability scanners as proactive security tools, emphasizing their role in early threat detection. They highlighted how routine scanning helps identify configuration flaws, unpatched software, and exposed services before they are exploited. The paper categorizes scanners by scope—network, host-based, application-level— and compares open-source vs commercial tools. Their work supports the integration of scanning into DevSecOps pipelines for continuous monitoring. They also underscore the importance of contextual analysis to prioritize remediation efforts [13]. WAVSEP Project (2021) is an open- source evaluation project designed to test the accuracy and reliability of web application vulnerability scanners. It offers a standardized environment with known vulnerabilities for benchmarking scanners. The project provides metrics such as detection rate, false positive ratio, and coverage per vulnerability type. WAVSEP has become a de facto standard in academic and professional research for evaluating scanner performance. Its continuous updates ensure compatibility with emerging vulnerability classes [14]. Chen (2020) led the Web Application Vulnerability Scanner Evaluation Project, an initiative to independently test and compare commercial and open-source scanning tools. His work includes in-depth reports on tools like Acunetix, Burp Suite, Netsparker, and more. Evaluation criteria include ease of use, detection capability, update frequency, and scan customization. Chen's database serves as a reference point for researchers, auditors, and organizations seeking to deploy effective scanning solutions. The project promotes transparency and helps improve vendor accountability [15].

EXISTING MODELS

Hybrid Active-Passive Scanning with Threat Intelligence Integration

Feature Extraction:

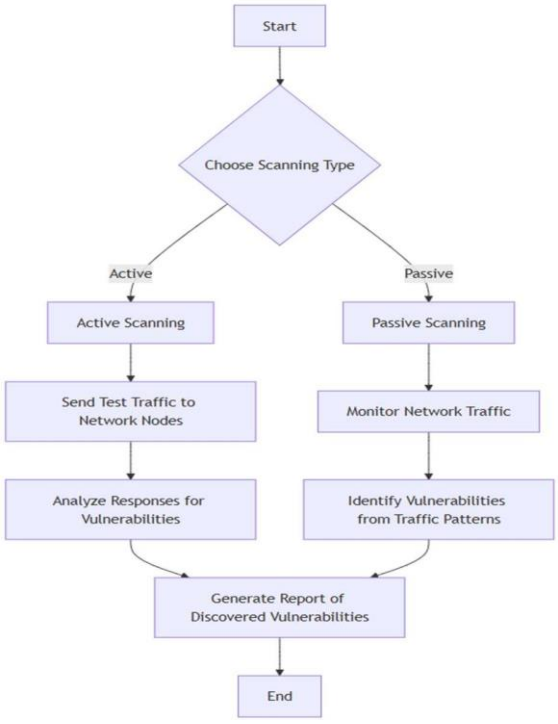
- This approach combines active scanning, which directly probes systems for vulnerabilities, with passive scanning, which monitors network traffic for suspicious patterns.
- External threat intelligence feeds are integrated to enhance real-time risk assessment by identifying known exploits from global databases.

Classification:

- Uses machine learning models to correlate findings from active and passive scans.
- Employs anomaly detection techniques to recognize zero-day vulnerabilities by analyzing deviations from normal traffic behavior.

Evaluation:

- Performance is assessed based on metrics like detection accuracy, false positive rate, and scan efficiency.
- Correlation analysis is used to improve accuracy by cross-referencing vulnerabilities with known threat intelligence sources.



Context-Aware Risk-Based Scanning

Risk Prioritization:

- Vulnerabilities are assessed based on exploitability, business impact, and system context, ensuring a targeted security approach.
- Unlike traditional scanning, this method does not treat all vulnerabilities equally but focuses on high- risk threats.

Threat Intelligence Integration:

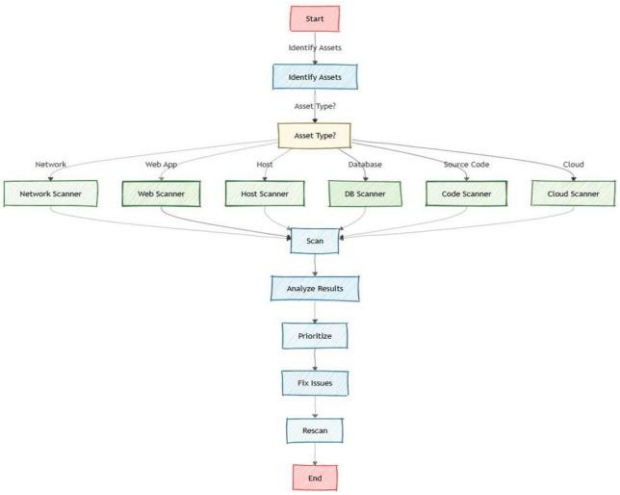
- Utilizes the Common Vulnerability Scoring System (CVSS) and advanced threat modeling to analyze potential risks.
- External threat intelligence sources are leveraged to identify and correlate known exploits in real time.

Decision-Making and Optimization:

- Reduces alert fatigue by filtering out low-risk vulnerabilities, allowing security teams to focus on critical issues.
- Provides contextual risk assessments to enhance security decision-making and optimize resource allocation.

Evaluation:

- Security effectiveness is measured using exploitability scores, severity ratings, and business impact metrics.
- Continuous updates ensure adaptation to evolving cybersecurity threats and emerging attack vectors.



Rule-Based Detection and Machine Learning Classification Models

Input Web Application:

- The user provides the web application's URL or other relevant input to start the scan.

Pre-Scanning Phase:

- This phase consists of two parallel processes: Network Traffic Monitoring:
- Observing network traffic for suspicious activity. Firewall Configuration Check:
- Analyzing firewall rules and settings to identify potential security misconfigurations.

Active Vulnerability Scanning:

- The scanner actively probes the web application to detect security flaws, such as SQL injection, XSS, and outdated software.

Scan Results Analysis:

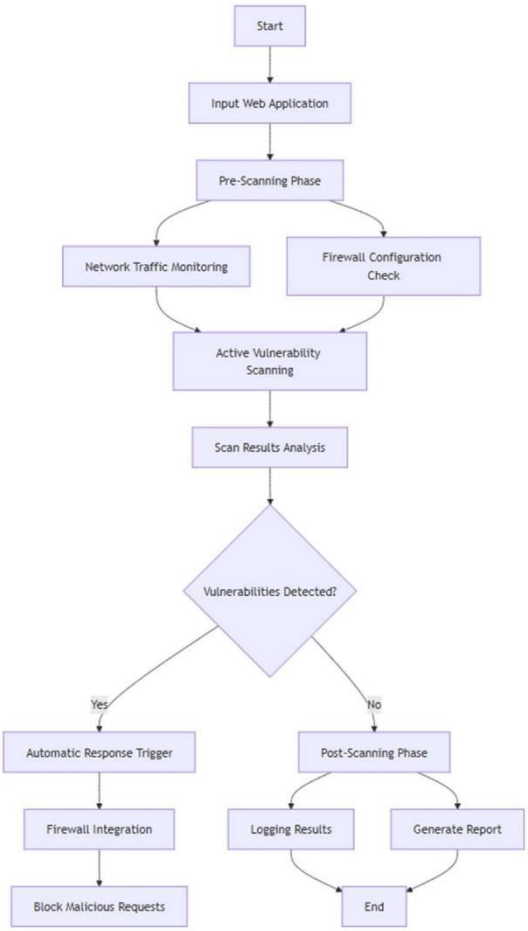
- The results from the vulnerability scanning are analyzed to determine if security risks exist.

Vulnerabilities Detected? (Decision Point):

- If vulnerabilities are found, the system takes an automatic security response.
- If no vulnerabilities are found, the process moves to post-scanning activities.

If Vulnerabilities Are Detected (Yes Branch):

- Automatic Response Trigger: Initiates countermeasures to mitigate the detected risks.
- Firewall Integration: Adjusts firewall rules dynamically to prevent attacks.
- Block Malicious Requests: Blocks potentially harmful traffic from exploiting vulnerabilities.
- If No Vulnerabilities Are Detected (No Branch - Post-Scanning Phase)
- Logging Results: Stores scan data for future reference and auditing.
- Generate Report: Creates a detailed report of the scan findings.



Dev Sec Ops-Integrated Continuous Vulnerability Scanning

Risk Reduction:

- Embeds automated security scanning within DevOps pipelines to catch vulnerabilities early.
- Scans code, dependencies, and configurations before deployment.

Threat Intelligence Integration:

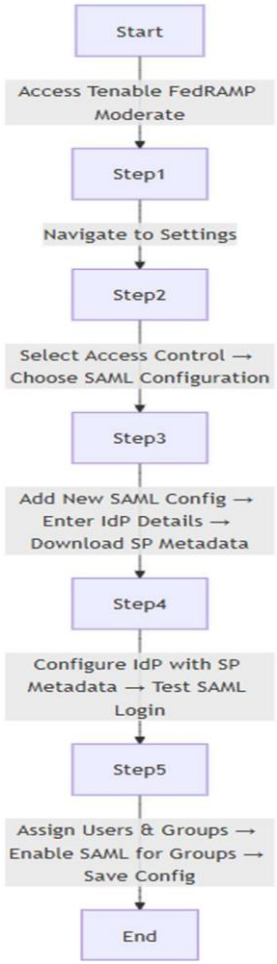
- Uses real-time scanning to detect security flaws in application code.
- Ensures secure software development lifecycle (SDLC) compliance.

Decision-Making and Optimization:

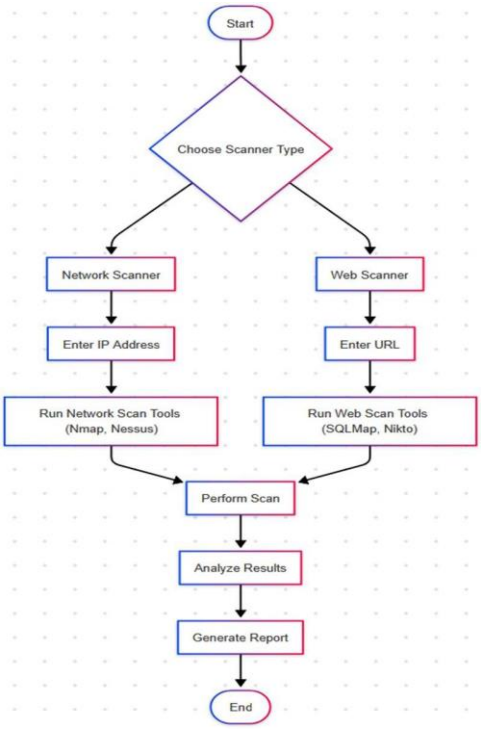
- Identifies security risks in development stages to reduce production vulnerabilities.
- Automates security checks to maintain fast and secure CI/CD processes.

Evaluation:

- Provides real-time feedback to developers for secure coding practices.
- Ensures compliance with industry frameworks like OWASP, NIST, and ISO 27001.


Start
Access Tenable FedRAMP Moderate
Step1
Navigate to Settings
Step2
Select Access Control → Choose SAML Configuration
Step3
Add New SAML Config → Enter IdP Details → Download SP Metadata
Step4
Configure IdP with SP Metadata → Test SAML Login
Step5
Assign Users & Groups → Enable SAML for Groups → Save Config
End

DESIGN


Start
Choose Scanner Type
Network Scanner
Web Scanner
Enter IP Address
Enter URL
Run Network Scan Tools (Nmap, Nessus)
Run Web Scan Tools (SQLMap, Nikto)
Perform Scan
Analyze Results
Generate Report
End

Start:

The process begins when the user launches the vulnerability scanning application. This is the initial phase where the user gets ready to interact with the system to perform a scan.

Choose Scanner Type:

At this stage, the user is presented with a choice between two types of scanning options Web Scanner or Network Scanner. This decision helps direct the tool to apply the correct set of scanners and techniques based on whether the target is a website or a network system.

Web Scanner Path:

If the user selects the Web Scanner option, the system prompts them to enter the URL of the web application they want to scan. This initiates a series of scans using a powerful

Combination of tools:

- Nikto is used to scan web servers for common vulnerabilities and outdated software.
- SQLMap helps in detecting SQL injection vulnerabilities.
- Nuclei runs fast, template-based scans to uncover a wide range of known issues in web apps.
- Wappalyzer identifies technologies and platforms used by the website (such as CMSs, libraries, or plugins).
- WhatWeb performs web server fingerprinting to detect backend technologies and software versions.

These tools work together to scan the website for security misconfigurations, vulnerabilities, and exposed components.

Network Scanner Path:

If the user selects the Network Scanner, they are asked to input the IP address of the system or network they want to check. Once the IP is entered, the scanner runs a series of tools to evaluate the network's security:

- Nmap is used to discover open ports, active services, and detect vulnerable software or configurations.
- Nessus (if implemented or mentioned) adds advanced vulnerability scanning to detect deeper system flaws and misconfigurations.

These tools are effective in mapping out the network surface and identifying weaknesses that could be exploited by attackers.

Perform Scan:

After the target (URL or IP) is provided and the relevant tools are chosen, the system automatically executes the selected scanning tools. These tools gather data on vulnerabilities, configurations, and system behavior, based on the user's input.

Analyze Results:

Once scanning is complete, the system analyzes the results. It filters out unnecessary data and highlights key security issues found during the scan. This stage focuses on making the output easy to understand, even for users who may not be security professionals.

Generate Report:

After analysis, a detailed report is generated. This report includes a summary of discovered vulnerabilities, categorized by severity (low, medium, high), along with descriptions and suggestions for fixing or mitigating each issue. This helps the user take informed action to improve their system's security.

End:

The process ends once the report is generated. The user can then review the results, apply recommended fixes, or start a new scan. This final phase marks the completion of one full scan cycle.

CASE STUDIES

Enhancing Cybersecurity with Active and Passive Vulnerability Scanning:

A financial institution implemented a dual-layer approach using both active and passive vulnerability scanning to secure its network and protect sensitive customer data. Active scanning conducted scheduled probes to identify misconfigurations and outdated software, while passive

scanning monitored traffic continuously for anomalies, ensuring real-time threat detection.

Outcome:

- The institution successfully minimized security gaps by detecting and addressing vulnerabilities promptly.
- Continuous monitoring reduced the risk of undetected threats without disrupting operations.
- A proactive approach bolstered trust among stakeholders, ensuring regulatory compliance and enhancing customer confidence in the organization's cybersecurity measures.

Ensuring Network Security with Multiple Vulnerability Scanners:

A global e-commerce company adopted a multi-layered cybersecurity framework through seven types of vulnerability scanners. Each scanner was strategically deployed to identify and mitigate security risks across different aspects of its infrastructure, from web applications to databases.

Outcome:

- The company significantly reduced the risk of breaches by addressing vulnerabilities at every level of its infrastructure.
- Real-time scanning and analysis improved the detection of emerging threats without disrupting day-to-day operations.
- The diverse scanning approach enhanced overall network resilience and boosted customer trust by ensuring secure online transactions.

Securing Web Applications with Automated Vulnerability Scanners:

A technology enterprise deployed an advanced automated vulnerability scanner integrated with firewall techniques to safeguard its web applications. The scanner was configured to perform real-time threat analysis, identify vulnerabilities in web application code, and automatically implement firewall rules to prevent exploitation.

Outcome:

- The organization successfully mitigated common vulnerabilities, such as SQL injection and cross-site scripting (XSS), reducing the risk of cyberattacks.
- The automated system ensured faster detection and resolution of security flaws without manual intervention.
- Enhanced application security resulted in improved customer trust and compliance with industry regulations.

Strengthening Government Security with Tenable Vulnerability Management:

A federal agency deployed Tenable Vulnerability Management (FedRAMP Moderate) to strengthen the security of its framework. The tool presented a centralized dashboard that could track vulnerabilities throughout its information technology infrastructure advanced analytics could prioritize critical threats while also providing compliance for strict federal regulations.

Outcome:

- The agency achieved real-time visibility into its security posture, enabling faster and more effective mitigation of risks.
- Automation of compliance audits reduced manual effort and ensured continuous alignment with government standards.
- The deployment built confidence among stakeholders and improved the agency's ability to respond to evolving cybersecurity threats.

RESULT & DISCUSSION

Discussion

This project was all about making vulnerability scanning easier for everyone by creating a simple web app that uses powerful tools from Kali Linux behind the scenes. Instead of typing long commands in the terminal, users can just open the website, enter a URL, IP address and the system takes care of the scanning.

Each scan works with tools that are trusted by cybersecurity professionals:

- Website scans use tools like Nikto and WPScan to check for things like outdated software or common web security issues.
- Network scans use Nmap to see which ports are open on a server or device, which could be potential entry points for attackers

One challenge we faced was running these tools from a web interface safely, since they usually require special permissions and can give a lot of raw output. We solved this by using backend scripts that run the tools and clean up the results to make them easy to read.

This project helped us understand how to take command-line tools and turn them into something more user-friendly. It was also a great way to learn more about web security and how different types of scans work in real situations.

7.2 Result

The web-based vulnerability scanner we built was successfully able to perform three types of scans:

Web Application Scanning:

When users entered a website URL, tools like Nikto and WPScan were used to check for issues like outdated software, insecure configurations, or known vulnerabilities in web applications.

Network Scanning:

When an IP address was entered, the scanner used Nmap to find open ports and detect what services were running, which helps in identifying possible entry points for attackers.

All the scan results were displayed clearly on the webpage and also saved for future reference. The system was tested using known vulnerable environments and it successfully detected multiple types of security issues, proving that the tool works as intended.

The overall performance of the scanner was smooth, and the results were reliable and accurate. The interface made it easier for users to perform scans without needing to run complex terminal commands, fulfilling the goal of making security tools more accessible.

CONCLUSION

This mini project, "Vulnerability Scanner Using Kali Linux Tools," shows how we can take powerful security tools and make them easier for anyone to use by building a simple web- based platform. Instead of using complex terminal commands, users can just open the web app and scan a website (URL)**, a** network (IP address)**,** or even upload a file to check for security issues. The scanner runs trusted tools like Nmap, Nikto and WPScan behind the scenes, but the user doesn't need to know any commands. This makes it perfect for students, developers, or anyone who wants to check for vulnerabilities without being a cybersecurity expert.In the end, the project achieved what we set out to do—build a basic but useful security scanner that's both functional and user-friendly. It also leaves room for future improvements like adding better reports, login features, or more advanced scanning options. This tool makes vulnerability scanning simpler, more accessible, and a lot less intimidating.

REFERENCES

[1] 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.

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

[3] 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.

[4] Pandey, S., Chaudhary, A. (2022). "Vulnerability Scanning." techrxiv. DOI: 10.36227/techrxiv.20317194Science(2021): 2582-5208.

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

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

[7] Deeptha R, K.Sujatha, D.Sasireka, R.Neelaveni , R.Pavithra Guru (2022). A Study on Web Application Vulnerabilities and Detection Techniques.

[8] Tenable. (2023). Tenable Vulnerability Management FedRAMP Moderate User Guide. Retrieved from https://docs.tenable.com/vulnerabilitymanagement/FedRAMP/ Content/PDF/ VM_FedRAMP_User_Guide.pdf.

[9] Bin Wang, Lu Liu, Feng Li, Jianye Zhang, Tao Chen & Zhenwan Zou(2020). Advances in Web Application Security: Vulnerability Detection and Prevention Approaches.

[10] Rathod, S.K. Jagtap, J.R. Satpute, A. P. Shikhare4, K.A. Pujari, A.S. Pandit (2021). An Automatic scanner for vulnerabilities Web Applications with Firewall Techniques.

[11] H Yulimanton 1,2*, H L H S Warnars1, B Soewtiol, F L Gaol1 And E Abdurachaman (2021). Web security and Vulnerabilities.

[12] Suliman Alazmi (Member, IEEE), and daniel conte de leon (Member, IEEE) (2020). An organized knowledge on the features and performance of web application vulnerability scanners .

[13] Sheetal Bairwa, Bhawna Mewara & Jyoti Gajrani (2021).Vulnerability scanners: a proactive Approach to assess security.

[14] Web Application Vulnerability Scanner Evaluation Project (Vulnerability Scanner Evaluation Project) (2021) . http://code.google.com/p/wavsep.

[15] Shay Chen. (2020). The Web Application vulnerability Scanner. Evaluation Project [EB] http://www.sectoolmarket.com.