E-ISSN No : 2581-6195 | Volume : 9 | Special Issue | May 2026

AUTOMATED VULNAREBILITY SCANNER USING PYTHON

Dr. Basanti Ghanti[1], Sakshikadaganchi[2], Sakshisappani[2], Ranjeetapawar[2], Priya Goudanguru[2]

ABSTRACT

The Automated Vulnerability Scanner is a cybersecurity tool designed to identify security weaknesses in web applications, servers, and network systems through automated scanning techniques. The project aims to reduce manual Fortin ulnar ability assessment by integrating reconnaissance, port scanning, service detection, and basic vulnerability analysis into a single automated framework. The scanner is developed using Python and utilizes networking and web security libraries to detect common vulnerabilities such as open ports, weak SSL/TLS configurations, missing security headers, SQL Injection indicators, and Cross-Site Scripting (XSS) issues. The system works by accepting a target URL or IP address, performing automated scans, analyzing responses, and generating structured security reports with severity classifications. The tool helps security professionals and system administrators quickly identify potential risks and improve overall system security. The project emphasizes ethical usage and is intended strictly for authorized security testing environments. Future enhancements may include machine learning-based vulnerability analysis, CVE integration, real-time dashboards, and cloud-based deployment for scalable security assessments

KEYWORDS: Automated Vulnerability Scanner, Cybersecurity, Vulnerability Assessment, Penetration Testing, Network Security, Web Application Security, Port Scanning, Service Detection, SQL Injection, Cross-Site Scripting (XSS), SSL/TLS Analysis, Security Automation, Risk Classification, Security Reporting, Ethical Hacking.

1. INTRODUCTION

With the rapid growth of internet-based services and digital infrastructure, cybersecurity has become a critical concern for organizations and individuals. Web applications, servers, and network systems are constantly exposed to cyber threats such as unauthorized access, malware attacks, data breaches, SQL Injection, Cross-Site Scripting (XSS), and other security vulnerabilities. Identifying these vulnerabilities manually is time-consuming, complex, and prone to human error, especially in large-scale environments. An Automated Vulnerability Scanner is a security tool developed to simplify and accelerate the process of vulnerability assessment. The scanner automatically examines target systems, detects security weaknesses, and generates detailed reports for analysis. By automating repetitive security testing tasks, the tool improves efficiency, consistency, and accuracy in identifying potential threats. This project focuses on developing an automated vulnerability scanner using Python and various security

[1] Professor, Electronics and Communication Engineering, Faculty of Engineering and Technology Exclusively for Women, Sharnbasva University, Kalaburagi, Karnataka, India.
[2] UG Students, Electronics and Communication Engineering, Faculty of Engineering and Technology Exclusively for Women

How to Cite:
Dr. Basanti Ghanti, Sakshikadaganchi, Sakshisappani, Ranjeetapawar & Priya Goudanguru (2026), Automated Vulnarebility Scanner using Python, International Educational Journal of Science & Engineering (IEJSE),Vol: 9, Special Issue, May 2026 115-118

Research Paper

testing libraries. The system performs functions such as port scanning, Vulnerabilities can be exploited by attackers.

2. LITERATURE SURVEY

The field of vulnerability assessment and penetration testing has evolved significantly with the advancement of cybersecurity technologies. Various researchers and organizations have developed automated tools and techniques to identify vulnerabilities in computer networks, servers, and web applications. Automated vulnerability scanners help security professionals reduce manual effort, improve detection accuracy, and perform continuous monitoring of systems.

One of the most widely used open-source vulnerability scanning tools is Nmap, developed by Insecure.Org. Nmap is primarily used for network discovery and port scanning. It helps identify open ports, running services, operating systems, and potential security weaknesses in target systems. Many modern scanners integrate Nmap for reconnaissance and network enumeration.

Another popular vulnerability assessment tool is Open VAS, which provides comprehensive vulnerability detection

Capabilities for enterprise environments. Open VAS uses a Regularly updated vulnerability data base to detect known security issues and generate detailed assessment reports. It has been widely adopted due to its open-source nature and extensive plugin support.

3. BACKGROUND

Cyber security has become one of the most important aspects of modern information technology due to the increasing dependence on digital systems, online services, and internet-connected devices. Organizations store and process large amounts of sensitive information such as financial records, personal data, and confidential business information on computer networks and web applications. As technology continues to evolve, cyber attackers constantly search for weaknesses in these systems to gain unauthorized access, steal data, or disrupt services.

A vulnerability is a weakness or flaw in software, hardware, or network configurations that can be exploited by attackers to compromise system security. Common vulnerabilities include weak passwords, outdated software, improper input validation, in secure network services, and misconfigured security settings. Cyberattacks exploiting these vulnerabilities can result in financial loss, reputational damage, and data breaches.

Traditionally, security professionals performed vulnerability assessments manually by inspecting systems, analyzing configurations, and testing applications. However, manual testing requires significant time, expertise, and resources, especially in large-scale environments. The growing complexity of networks and web applications has increased the need for automated security assessment tools that can quickly and efficiently identify vulnerabilities.

4. METHODOLOGY

The methodology of the Automated Vulnerability Scanner follows a structured and systematic approach to identify security weaknesses in target systems. The process begins with requirement analysis, where the functional goals of the system are defined, such as scanning IP addresses or URLs, detecting vulnerabilities, and generating detailed reports. After this, the system design phase is carried out, where the overall architecture is divided into modules like input handling, scanning engine, vulnerability detection, and reporting. This modular design ensures better organization and easy maintenance of the system.

Once the design is finalized, the target information gathering process begins, where basic details such as IP resolution, domain information, and server response headers are collected. The next step involves port scanning, where the system checks for open ports

Key Steps and Components

1. Target Identification

The first step is to collect information about the target system. The user provides:

- IP address
- Domain name
- Website URL

The scanner verifies the target and prepares it for analysis.

2. Information Gathering

The scanner gathers basic information about the target, including:

- IP resolution
- Open ports
- Running services
- Server details
- HTTP headers

This phase helps understand the target environment before vulnerability testing begins.

3. Port Scanning

The system scans commonly used network ports to identify active services such as:

- HTTP (80)
- HTTPS (443)
- FTP (21)
- SSH (22)

Open ports may indicate exposed services that could contain vulnerabilities.

4. Service and Banner Detection

The scanner identifies:

- Service versions
- Server software
- Protocol information

Banner grabbing techniques are used to collect this information for vulnerability analysis.

5. Vulnerability Analysis

The scanner checks for common security vulnerabilities, including:

- SQL Injection
- Cross-Site Scripting (XSS)
- Weak SSL/TLS configurations
- Missing security headers
- In secure services

The system compares collected data against predefined security rules and patterns.

6. Risk Classification

Detected vulnerabilities are categorized according to severity:

- Critical
- High
- Medium
- Low

This helps prioritize security fixes.

4.2 Key Advantages of Computing for Edge AI Neuromorphic

1. User Interface Module

This module allows users to:

- Enter target details
- Start scans
- View scan progress
- Access reports

The interface may be command-line based or graphical.

2. Scanning Engine

The scanning engine is the core component responsible for:

- Port scanning
- Service detection
- Vulnerability checks
- Data collection

It coordinates all scanning activities.

3. Vulnerability Detection Module

This module analyzes collected data and identifies security weaknesses using:

- Signature-based detection
- Pattern matching
- Security rules

4. SSL/TLS Analysis Module

This component checks:

- SSL certificate validity
- TLS protocol versions
- Encryption strength
- Secure communication settings

Weak configurations are flagged as security risks.

5. Reporting Module

The reporting module generates structured output such as:

- PDF reports
- JSON reports
- Text summaries

It presents vulnerabilities in are ad able format for administrators.

6. Database or Log Storage

This component stores:

- Scan history
- Vulnerability records
- Logs and results

Stored data can be used for future comparison and monitoring.

5. RESULTS

The implementation of the Automated Vulnerability Scanner demonstrates effective identification and reporting of common security vulnerabilities in target systems. The system successfully performs automated scanning of IP addresses and web applications, providing structured and meaningful security insights with minimal manual intervention. During testing in a controlled environment, the scanner was able to detect key security issues such as open ports, exposed services, missing HTTP security headers, and weak SSL/TLS configurations. It also identified potential indicators of common web vulnerabilities like SQL Injection and Cross-Site Scripting (XSS) based on response behavior and pattern analysis. The results generated by the system were organized into a clear and readable report format. Each detected vulnerability was classified according to severity levels such as Critical, High, Medium, and Low. This classification helped in prioritizing security risks and understanding their potential impact on the system. The scanning process showed that automated techniques significantly reduce the time required for vulnerability assessment compared to manual testing methods. The system provided consistent and repeatable results across multiple test cases, improving reliability insecurity evaluation.

6. CONCLUSION

The Automated Vulnerability Scanner project successfully demonstrates an efficient approach to identifying security weaknesses in web applications and network systems. By automating key security testing processes such as port scanning, service detection, and vulnerability analysis, the system reduces manual effort while improving the speed and consistency of security assessments. The tool is capable of detecting common vulnerabilities including open ports, misconfigured services, weak SSL/TLS configurations, and basic web application flaws such as SQL Injection and Cross-Site Scripting indicators. It also generates structured reports that help security professionals and system administrators understand risks and take appropriate corrective actions. From the results obtained, it is evident that automation significantly enhances the effectiveness of vulnerability assessment, especially in environments that require continuous monitoring

However, the system also has limitations, such as the possibility of false positives and the inability to detect complex logical vulnerabilities that require manual penetration testing. Overall, the project achieves its objective of providing a simple, scalable, and automated solution for basic vulnerability detection. It also serves as a strong foundation for further enhancements, such as integration with machine learning models, real-time monitoring systems, and advanced threat intelligence databases. The project emphasizes the importance of ethical usage and is intended strictly for authorized cybersecurity testing and educational purposes.

REFERENCE

1. Nmap Gordon Lyon (Fyodor), Nmap Network Mapper Documentation. Used for network discovery, port scanning, and service detection in security assessments.
2. OpenVAS Green bone Networks, OpenVAS Official Documentation. Provides vulnerability scanning framework and continuously updated vulnerability feeds.
3. OWASP ZAP OWASP Foundation, OWASPZAP User Guide. Used for Automated web application security testing and vulnerability detection.
4. Stallings, William. Network Security Essentials: Applications and Standards. Pearson Education. Covers fundamental concepts of network security, encryption, and vulnerability management.
5. Whitman, Michael E., and Herbert J. Mattord. Principles of Information Security. Cengage Learning. Provides foundational knowledge of information security principles and risk assessment.
6. Aho, Alfred V., et al. Compilers and Security Analysis Concepts. (Reference concepts Applied in pattern matching and detection techniques).