Automated Network Enumeration and Vulnerability Analysis Tool: A Web-Based Approach with AI-Assisted Vulnerability Assessment

[1]Sarikha S, [2]Naskath J, [3]Sri Gomathi R, [4]Aruna Varshini S
[1,3,4] UG – Computer Science and Engineering, [2]Associate Professor – Artificial Intelligence and Data Science
[1]Department of Computer Science and Engineering,
[1]National Engineering College, Kovilpatti, Tamil Nadu, India
[1]sarikhasp18@gmail.com, [2]naskat@nec.edu,[3]rsrigomathi@gmail.com,[4]arunasedhu@gmail.com

Abstract— Robust network security requires accurate host enumeration and timely vulnerability detection. To simplify these tasks for both novice and professional users, this paper presents an automated web-based vulnerability assessment tool that integrates advanced Nmap scanning with real-time analysis and artificial intelligence assistance. Developed using the Flask framework with an SQLite backend, the system provides guest quick-scan access and authenticated deep-scan functionality, supports multiple scan execution with history management, and correlates findings with the National Vulnerability Database. The platform also incorporates DNS-over-HTTPS resolution and an AI-powered chatbot that interprets scan results in plain language and recommends remediation steps. Experimental evaluation demonstrates scan completion within five seconds for quick scans and between fifteen and thirty seconds for deep scans. The system achieves significant improvement in usability compared with traditional command-line tools while maintaining strong detection performance. This scalable platform bridges the accessibility gap in network security and provides a practical solution for ethical hacking, network auditing, and cybersecurity education.

Index Terms— Network security, vulnerability assessment, Nmap, CVE mapping, artificial intelligence assistant, web-based scanning, cybersecurity education.

1. INTRODUCTION

Network security has become a critical priority as modern organizations rely heavily on complex, interconnected digital environments. Enterprise networks now extend far beyond traditional office boundaries, encompassing remote workers, mobile devices, cloud services, and Internet of Things (IoT) endpoints. Each new connection increases the potential attack surface, giving malicious actors more opportunities to exploit weaknesses. At the same time, cyberattacks are growing in sophistication, employing advanced tactics such as zero-day exploits, stealthy lateral movement, and automated scanning to identify vulnerable systems. In this climate, discovering and addressing security gaps before they are exploited is essential to maintaining the confidentiality, integrity, and availability of information assets [1].

A fundamental starting point for any security assessment is network enumeration and vulnerability analysis. Network enumeration involves systematically identifying active hosts within a network, detecting open or filtered ports, and cataloging the services running on those ports. Vulnerability analysis builds on this information by highlighting misconfigurations, outdated software, or weak authentication mechanisms that could be leveraged by attackers. Together, these processes provide administrators with a detailed map of the network’s topology and its potential weak spots, enabling proactive remediation before an adversary can gain a foothold. Among the various tools available for this purpose, Nmap (Network Mapper) has earned a reputation as the industry standard. Nmap offers a rich feature set, including comprehensive TCP and UDP scanning, operating system fingerprinting, service version detection, and flexible scripting capabilities for deeper analysis. Security professionals and researchers value its reliability and extensibility, making it a cornerstone of penetration testing and routine network audits. However, Nmap’s text-based, command-line interface can be daunting for newcomers or those without strong technical expertise [2]. Understanding its wide array of options, flags, and scripting possibilities often requires a steep learning curve. As a result, cybersecurity students, educators, and even some working professionals may find the tool challenging to adopt in day-to-day practice, limiting its use in classrooms, training programs, and smaller organizations with limited security staff [3].

This usability gap creates a significant obstacle. Educational institutions seek practical, hands-on experiences for students, but a complex tool can discourage engagement and slow learning. Likewise, small and medium-sized businesses—often the most vulnerable to cyberattacks—may postpone or avoid regular network assessments simply because the available tools appear too technical or resource-intensive. Moreover, modern security assessments demand more than raw scan results. Administrators and analysts need contextual interpretation, automated risk scoring, and clear, actionable recommendations to prioritize remediation efforts effectively [4]. Without these enhancements, valuable data from scans can remain underutilized, reducing the impact of security initiatives. To address these challenges, this paper presents a web-based network enumeration platform that integrates Nmap's powerful scanning capabilities with an intuitive graphical interface. The system incorporates database management for storing and tracking scan results over time, enabling trend analysis and historical comparisons. In addition, an AI-powered assistant offers guidance on interpreting findings, assessing risk levels, and recommending next steps for remediation. This approach lowers the barrier to entry for beginners while providing experienced professionals with advanced features and analytical depth. By bridging the gap between sophisticated scanning technology and user-friendly design, the proposed platform supports both education and enterprise needs, fostering more frequent and effective security assessments across a wide range of organizations.

2. LITERATURE REVIEW

Modern networks are no longer static, isolated systems—they are dynamic ecosystems of cloud servers, on-premises hardware, mobile devices, and countless connected sensors. As this web of devices grows denser and attackers employ more sophisticated techniques, the job of protecting critical data and services has become significantly more challenging. Recent academic and industry research shows that vulnerability assessment itself is evolving to match these new realities, moving away from simple checklists and numeric risk ratings toward context-aware, intelligent approaches that can adapt to changing environments.

Traditional scoring models such as the Common Vulnerability Scoring System (CVSS) remain an important foundation, but researchers have recognized that static scores alone cannot capture the complexity of modern networks. Wang and colleagues [1], for example, proposed a graph-based model that represents vulnerability contexts using heterogeneous information networks. By incorporating details such as network topology and the relationships between different services, their method ranks risks in a way that reflects how an attacker might actually move through a large infrastructure. This shift from one-dimensional scoring to a more structural view of risk provides security teams with a clearer understanding of which weaknesses pose the greatest real-world threat. The explosive growth of the Internet of Things (IoT) has further reshaped the security landscape. Billions of small, often poorly secured devices—from smart thermostats to industrial sensors—are now permanently online. Gupta et al. [3] demonstrated how these endpoints can be recruited into massive "botnets of things," allowing attackers to compromise entire network segments through a single vulnerable device. Verma et al. [4] addressed the unique challenge of scanning such dynamic wireless environments, where devices frequently join and leave the network. Their adaptive scanning technique achieved over 90 percent accuracy in identifying the operational state of devices, a critical step toward defending networks in which endpoints are constantly in flux. Comprehensive protection requires multiple layers of testing rather than a single assessment method. Cruz and collaborators [5] compared a suite of open-source security tools, evaluating software composition analysis (SCA), dynamic application security testing (DAST), and static application security testing (SAST). Their framework shows that combining these approaches throughout the software development lifecycle yields far better coverage, catching issues that might slip past any single technique. In sensitive fields such as healthcare, Kandasamy et al. [6] went further by mapping detected vulnerabilities to established NIST cybersecurity frameworks, allowing organizations to create mitigation strategies that are both technically precise and aligned with regulatory requirements. Similarly, Aslam et al. [7] focused on Industrial Control Systems—vital to energy, water, and manufacturing sectors—cataloging common architectural flaws and analyzing real incidents to highlight how even small weaknesses can lead to large-scale disruption.

Artificial intelligence is emerging as a transformative force in this domain. Tamberg and co-authors [9] demonstrated that large language models (LLMs) can uncover more potential vulnerabilities than traditional static analysis tools, achieving higher recall rates despite producing a greater number of false positives. This trade-off suggests that AI can serve as a powerful first-line detector, surfacing subtle issues that conventional scanners may overlook, while human analysts refine the final results. Beyond simple detection, ongoing monitoring is equally vital. Jafarian et al. [10] created inline detection systems that correlate network flow data with DNS queries to identify scanning activity in real time, enabling defenders to spot reconnaissance attempts before they escalate. Complementing this, Lyu et al. [12] focused on large enterprises, developing distributed attack-detection and behavioral monitoring techniques that can recognize suspicious patterns across a sprawling organizational network. Taken together, these studies reveal a clear direction for the future of vulnerability assessment. Security tools are moving toward intelligent, context-aware platforms that can learn from past scans, integrate diverse data sources, and adapt to a variety of network environments. Rather than relying solely on raw output, next-generation systems will combine automated scanning, AI-driven analysis, and sector-specific expertise to deliver actionable insights. This integrated approach lays the groundwork for security solutions capable of keeping pace with the rapidly evolving threat landscape, providing defenders with the agility and depth needed to protect complex, interconnected infrastructures.

3. METHODOLOGY

The Automated Network Scanner is designed to modernize and simplify the process of network-vulnerability assessment by transforming the traditionally command-line-driven Nmap utility into an intuitive, browser-based platform. The methodology is guided by three central objectives: accessibility, intelligent vulnerability detection, and actionable reporting. These objectives are realized through a set of coordinated modules that together create a seamless scanning, interpretation, and remediation workflow. Each module operates independently while exchanging data with the others to maintain accuracy and consistency.

3.1 Web-Based Scanning Module

The first module forms the operational core of the system. It replaces Nmap’s steep command-line learning curve with a responsive web interface that allows users to configure and launch scans using only a browser. By abstracting the syntax of Nmap into point-and-click controls, the module makes professional-grade scanning accessible to a much broader audience—including students, small-business owners, and security analysts—without sacrificing the full capabilities of the underlying engine.

3.1.1 Quick Scan Mode: Quick Scan is designed for situations where immediate situational awareness is more important than exhaustive analysis. This mode rapidly checks for active hosts, commonly exploited open ports, and a small set of high-impact misconfigurations. Because it runs with a lightweight configuration and requires minimal input, it is ideal for classroom demonstrations, preliminary health checks, or small organizations that need fast feedback on basic network hygiene.

```
{
  "url": "http://127.0.0.1:5000/scan",
  "method": "POST",
  "inputs": [
            { "name": "ip", "type": "text" },
            { "name": "scanType", "type": "select" }
  ]
}
```

3.1.2 Deep Scan Mode: Deep Scan provides a far more comprehensive examination of the target environment. It performs extensive TCP/UDP probing, service fingerprinting, operating-system detection, and version enumeration to identify vulnerable services and weak configurations. This level of analysis is tailored to IT administrators, penetration testers, and security professionals who require a detailed vulnerability map and precise data for incident response or compliance reporting.

```
{
  "url": "http://localhost:3000/api/scan",
  "method": "POST",
"inputs": [
        { "name": "target", "type": "text" },
        { "name": "scanType", "type": "select" },
        { "name": "command", "type": "hidden" }
 ]
}
```

3.1.3 Structured Data Output: Regardless of the scan type, all results are normalized and stored in a structured format such as JSON. The output includes key metadata—target IP range, timestamps, discovered hosts, open or filtered ports, detected services, and preliminary risk scores. Storing the data in a standardized schema enables longitudinal analysis, automated report generation, and integration with external tools such as SIEM platforms or vulnerability-management dashboards.

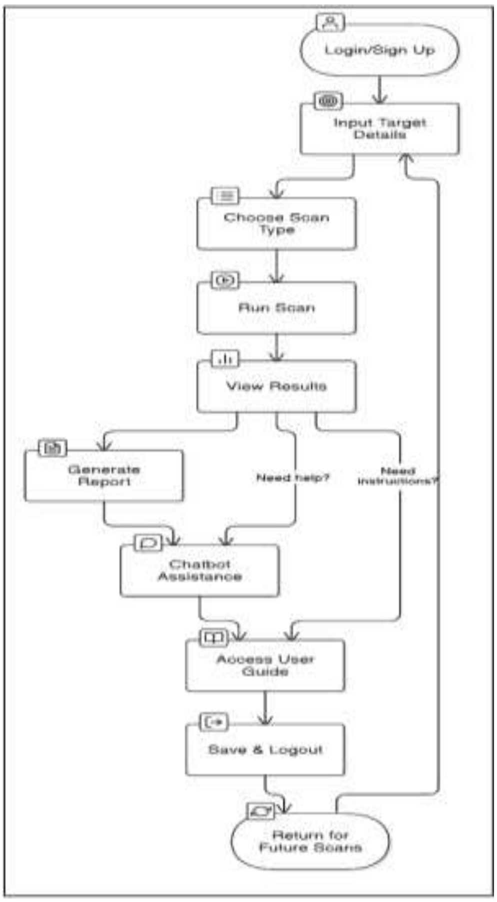


Figure 1. General Methodology

3.2 AI-Powered Vulnerability Interpretation

While Nmap traditionally produces raw technical logs, the second major module focuses on translating those results into **clear, actionable intelligence**. This interpretation layer is critical for non-experts and greatly reduces the time security teams spend parsing command-line output.

3.2.1 Plain-Language Explanations: Each finding is automatically converted into concise, human-readable text. For example, instead of presenting a cryptic message such as “Port 22 open,” the system provides an explanation like: “The server is accepting remote connections over SSH. If not properly configured or patched, this service may allow unauthorized access.” Such phrasing allows decision-makers with limited technical backgrounds to understand the implications immediately.

3.2.2 Real-Time Vulnerability Correlation: The platform cross-references scan results with up-to-date threat intelligence feeds and public vulnerability databases including the National Vulnerability Database (NVD) and the Common Vulnerabilities and Exposures (CVE) repository. This live correlation ensures that severity ratings and suggested countermeasures reflect the latest advisories and emerging attack trends, an essential capability in today’s fast-moving threat landscape.

3.2.3 Remediation Guidance: Beyond simply identifying weaknesses, the system provides concrete mitigation steps. If an outdated service version is detected, the interface recommends specific actions such as applying vendor patches, disabling unused ports, or enforcing stronger authentication mechanisms. The guidance is prioritized by risk level, enabling administrators to address the most critical vulnerabilities first and allocate resources effectively.

3.3 Automated Reporting and Historical Analysis

The modules described above are connected by a unified workflow. Users begin by selecting the appropriate scan type, monitor real-time progress through the browser interface, and then receive a comprehensive report that includes both technical details and high-level recommendations. Historical scan data is archived in the backend database, allowing organizations to track changes in their security posture over time and demonstrate compliance with internal or regulatory standards.

3.3.1 PDF and JSON Report Generation: At the conclusion of each scan, the platform automatically produces reports in two complementary formats—PDF and JSON. The PDF report offers a polished, reader-friendly summary that highlights key findings, severity ratings, and recommended remediation steps, making it ideal for presentations to management or compliance audits. The JSON report, by contrast, retains the full technical detail and structured data required by IT teams and automated security workflows. This dual-report strategy ensures that both non-technical decision-makers and technical staff receive the information in a form they can readily understand and act upon.

3.3.2 Historical Trend Tracking: Unlike traditional scanning tools that provide only a snapshot of the current network state, the Automated Network Scanner archives every set of results in a secure backend database. This feature allows users to compare present findings with previous scans, identify recurring vulnerabilities, and track the success of remediation efforts over time. Trend analysis helps organizations demonstrate measurable improvements in security posture, verify that applied fixes remain effective, and comply with regulatory requirements that demand evidence of continuous monitoring.

3.3.3 Mobile-Friendly Access: Recognizing the need for flexibility in modern security operations, the platform provides a responsive interface accessible from desktops, tablets, and mobile devices. Administrators and security teams can review reports, check scan histories, and even initiate follow-up scans while away from their primary workstations. This mobility ensures that critical information remains available to stakeholders anytime and anywhere, supporting rapid decision-making in the event of emerging threats.

3.4 Interactive User Support and Guidance

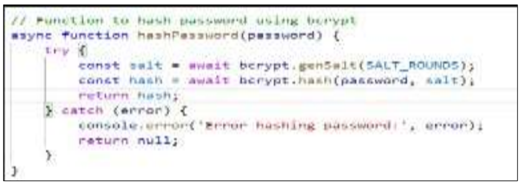
The final module is dedicated to user experience, transforming the system from a simple scanning tool into an interactive learning and advisory environment. Its goal is to empower a wide range of users—from novices to experienced professionals—by providing context-aware assistance throughout the scanning process.

3.4.1 Interactive User Guide: A built-in, step-by-step guide walks users through each stage of configuration and execution. This interactive guide minimizes the need for external manuals or formal training sessions and reduces the risk of misconfiguration that could lead to incomplete scans or inaccurate results. Pop-up tips and contextual explanations appear as users navigate the interface, making the process intuitive even for those encountering vulnerability scanning for the first time.

3.4.2 AI Chatbot Assistance: The platform integrates a conversational chatbot that allows users to ask questions in natural language and receive contextual, real-time answers. Queries such as “What is a critical vulnerability?” or “How do I secure port 443?” trigger concise explanations and actionable advice. The chatbot draws on both system knowledge and up-to-date security references, enabling users to clarify technical concepts or find remediation steps without leaving the application.

3.4.3 Scalability for Different Users: Recognizing that security expertise varies widely, the system adapts its interface and available options to the skill level of the user. Beginners can rely on the Quick Scan mode and simplified explanations, while advanced users can customize scan parameters, script their own checks, and conduct in-depth analyses. This scalability makes the platform equally valuable in educational settings, small and medium-sized enterprises, and professional security operations centers.

3.4.4 Security of the Web Application: Because the system itself is part of the attack surface, dedicated measures are built into the architecture. Role-Based Access Control (RBAC) enforces separation of privileges between users and administrators. Sandboxed Execution of Nmap commands ensures scans run in isolated environments, minimizing risks of system compromise. Input Sanitization and Logging protect against injection attacks and enable forensic auditing of user actions. This ensures that the platform maintains credibility as a security tool by avoiding the vulnerabilities it is designed to detect.



```
// Function to hash password using bcrypt
async function hashPassword(password) {
    try {
        const salt = await bcrypt.genSalt(SALT_ROUNDS);
        const hash = await bcrypt.hash(password, salt);
        return hash;
    } catch (error) {
        console.error('Error hashing password:', error);
        return null;
    }
}
```

Figure 2. JWT Authentication and Hashing Process

3.5 Database Design and Data Flow

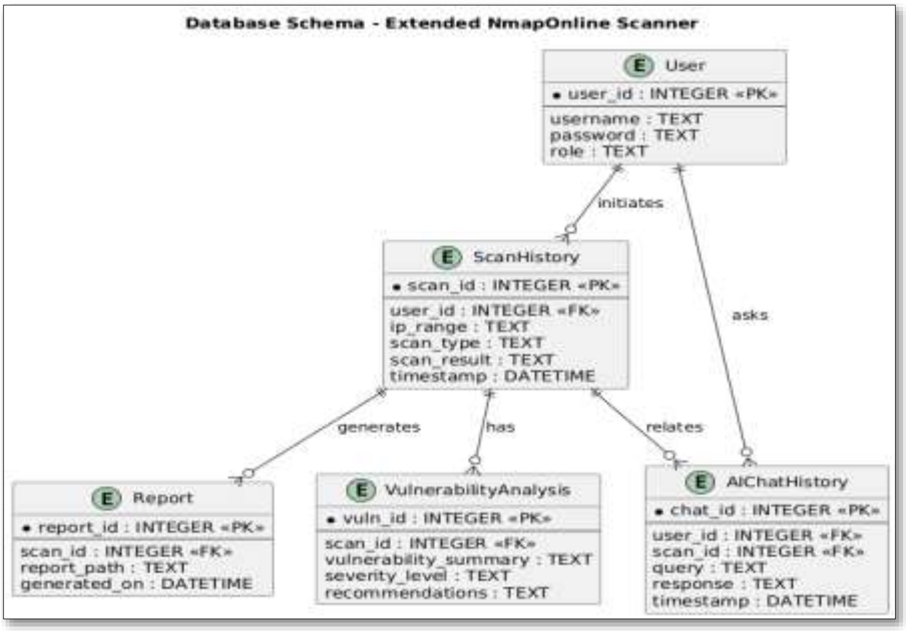


Figure 3. Database Schema

Fig 3 shows the database design which supports core functionalities such as user authentication, scan tracking, vulnerability storage, and AI-driven chat history. A relational schema using SQLite is chosen for its lightweight nature and easy integration with Flask. The structure is kept simple yet efficient, ensuring secure handling of user data, proper linkage of scan results, and streamlined report generation.

3.6 Comparative Analysis Framework

To establish scientific rigor, ANS is benchmarked against both baseline tools (raw Nmap) and enterprise-grade platforms (OpenVAS, Nessus, and Qualys).

1. Metrics for Comparison include Detection Accuracy (ratio of correctly identified vulnerabilities to total ground truth CVEs). False Positive Rate (FPR), quantifying over-reporting tendencies. Scan Efficiency (setup time, execution time, report generation time). Resource Utilization (CPU, memory, and storage footprint). Usability Scores collected from penetration testers and students in controlled trials.

Table 1. Feature Comparison Matrix

Feature	Traditional Nmap	Proposed Platform
UI	Runs only in command line	Easy web-based interface
Installation	Must install on every system	Works directly in browser
Learning Curve	Hard to learn for beginners	Simple and beginner-friendly
Vulnerability Check	User checks vulnerabilities manually	System checks automatically
Report Generation	Reports created by hand	Reports generated instantly
AI Assistances	No help or guidance	AI explains and guides users
Compatibility	Works on limited platforms	Works on any device with a browser

Table 1 highlights the key differences between traditional Nmap and the proposed platform. Unlike Nmap's command-line interface and manual processes, the platform offers a web-based GUI, automated reporting, and built-in AI assistance. This makes scanning simpler, faster, and more user-friendly across different systems. The bar chart shows that the Automated Network Scanner improves usability, accessibility, input handling, and reporting compared to traditional Nmap. Unlike Nmap's manual setup, it provides a browser-based, automated, and user-friendly experience for faster and easier scanning.

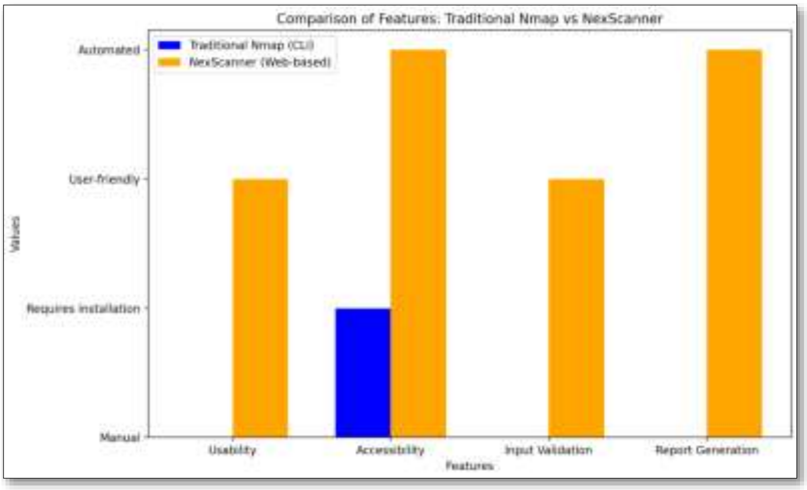


Figure 4. Comparative Analysis

4. IMPLEMENTATION

The Automated Network Scanner was implemented as a secure, browser-based platform that brings the full capabilities of Nmap to a wider audience. The system allows two primary modes of operation: quick scans for guests who need an immediate overview and deep scans for authenticated users who require detailed vulnerability analysis. A lightweight terminal-style interface streams results in real time, so users can watch the progress of a scan as it unfolds. Behind the interface, the application enforces strict input validation, manages user sessions, and applies role-based access control to prevent misuse or unauthorized access. These safeguards ensure that scans are performed safely while preserving the flexibility needed by both novice users and experienced security professionals.

4.1 Technology Stack Selection

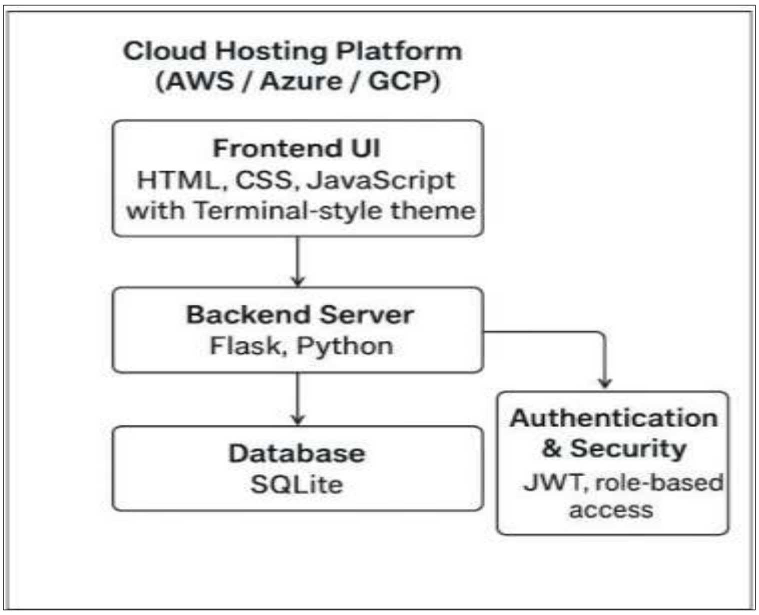


Figure 5. Technology stack

Automated Network scanner's technology stack is built for ease, security, and scalability. The frontend uses HTML5, CSS3, and JavaScript with optional Bootstrap for responsive design. The backend runs on Python with Flask to manage routing, sessions, and secure scan execution. SQLite stores scan history, roles, and reports. Nmap works as the core scanning engine, while LangChain with GPT-based models is used for analyzing results and improving detection ability. This setup makes the system simple, reliable, and powerful for both beginners and experts.

4.2 Class Diagram and System Architecture

The system adheres to the traditional three-tier architecture, with the Presentation Layer, Application Layer, and Data Layer. A modular approach ensures improved separation of concerns, maintenance simplicity, and future scalability. The user interface is built with HTML5, CSS3, and JavaScript, following a terminal-inspired layout that will be familiar to command-line tool users. Users can specify an IP address or range, select between Quick or Deep scans, and see real-time updates of scan activity. Real-time interactivity like animations, status messages, and error messages are handled through JavaScript. Optionally, Bootstrap can be incorporated for added responsiveness and visual appearance. Developed using Python 3.2.2 and the Flask web framework, the backend has the application's core functionality. It regulates routing, session management, form validation, and securely builds Nmap commands from user input. Python's subprocess module is utilized to run Nmap scans, while Flask provides authenticated access to enhanced scan modes. This layer applies business logic like user role verification, input sanitizing, and result formatting.

The system uses SQLite as its database solution to handle user accounts, scan history, and report metadata. This lightweight relational database is used to store information like login credentials (with permission levels based on roles), scan settings, and URLs for downloadable reports. It is an important component in supporting the authenticated Deep Scan capability and provides users with the facility to track and revisit past scan results.

Fig 5 depicts the system's three-tier architecture, demonstrating how each layer works together to offer a secure, user-friendly, and effective network scanning platform.

The AI assistant bridges the gap between raw scan data and meaningful security insight. Plain-Language Summaries are generated so that technical results such as "Port 443 open, TLS 1.0 detected" are converted into actionable statements like "The server supports deprecated TLS version, which may allow downgrade attacks."

Vulnerability Correlation Engine automatically maps service versions and OS fingerprints to CVE and NVD entries. This is achieved using text embedding–based similarity between scan fingerprints and CVE metadata, ensuring robust and flexible matching. We compute a match score between a detected service s and CVE c:

$$\text{match}(s,c)=\alpha\cdot\text{prod\_match}(s,c)+\beta\cdot\text{version\_match}(s,c)+\gamma\cdot\text{keyword\_sim}(s,c) \qquad (1)$$

Where, prod_match ∈ {0,1} if product name matches known affected product (fuzzy match allowed), version_match ∈ [0,1] measures semantic version overlap (0 if mismatch, 1 if exact), keyword_sim ∈ [0,1] is cosine similarity (or token overlap) between service/version text and CVE description. Weights α, β, γ sum to 1 (default α=0.5, β=0.3, γ=0.2; configurable). A CVE is considered correlated if match(s,c) ≥ τ (threshold τ, e.g., 0.6). We then compute a risk score for the detected issue:

$$\text{risk}(s,c)=\lambda\cdot 10\text{CVSSI}+(1-\lambda)\cdot \text{match}(s,c) \qquad (2)$$

where CVSS ∈ [0,10] and λ ∈ [0,1] controls emphasis on vendor severity vs. observed evidence (default λ=0.7).

1. Evaluation Metrics are applied to validate the AI- generated outputs: Interpretation Accuracy (percentage of AI summaries aligned with expert validation). False Hallucination Rate (FHR):

$$\text{FHR} = \frac{(\text{AI}-\text{generated invalid CVEs})}{(\text{Total CVEs reported})} \qquad (3)$$

2. Precision, Recall, and F1-score measured on a labelled dataset of 500 vulnerabilities.

This allows the AI to not only assist users but also provide quantifiable guarantees of reliability

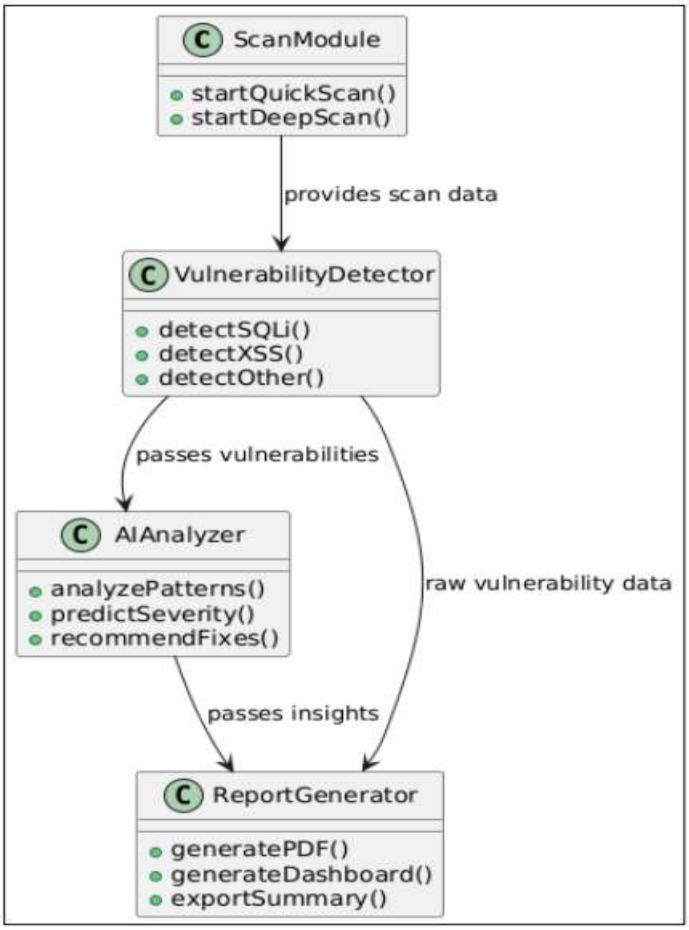
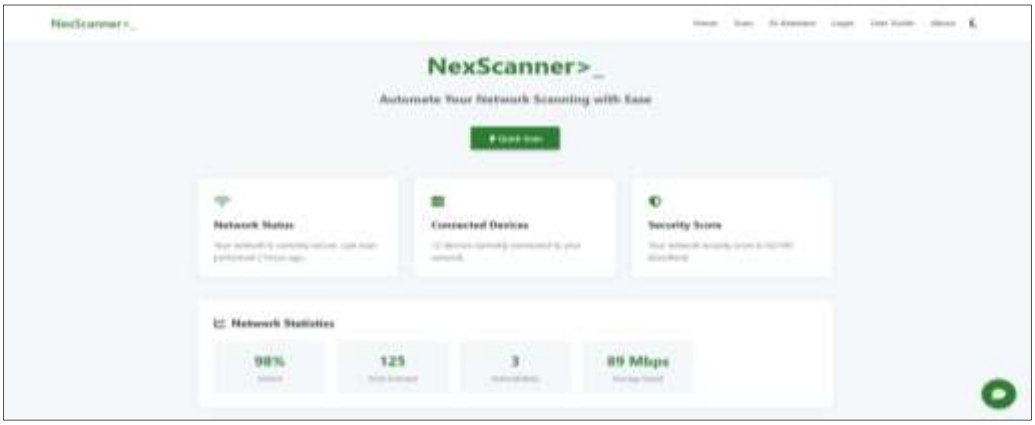


Figure 6. Automated Network Scanner Class Diagram

Fig 6 shows how the automated network scanner works. The Scan Module gathers inputs and runs scans. The Vulnerability Detection Module finds possible security issues. The AI Module (LLM Engine) analyzes results for severity and insights. Finally, the Reporting Module creates clear reports for remediation.



NexScanner>_

Automate Your Network Scanning with Ease

Network Status

Connected Devices

Security Score

Network Statistics

98%

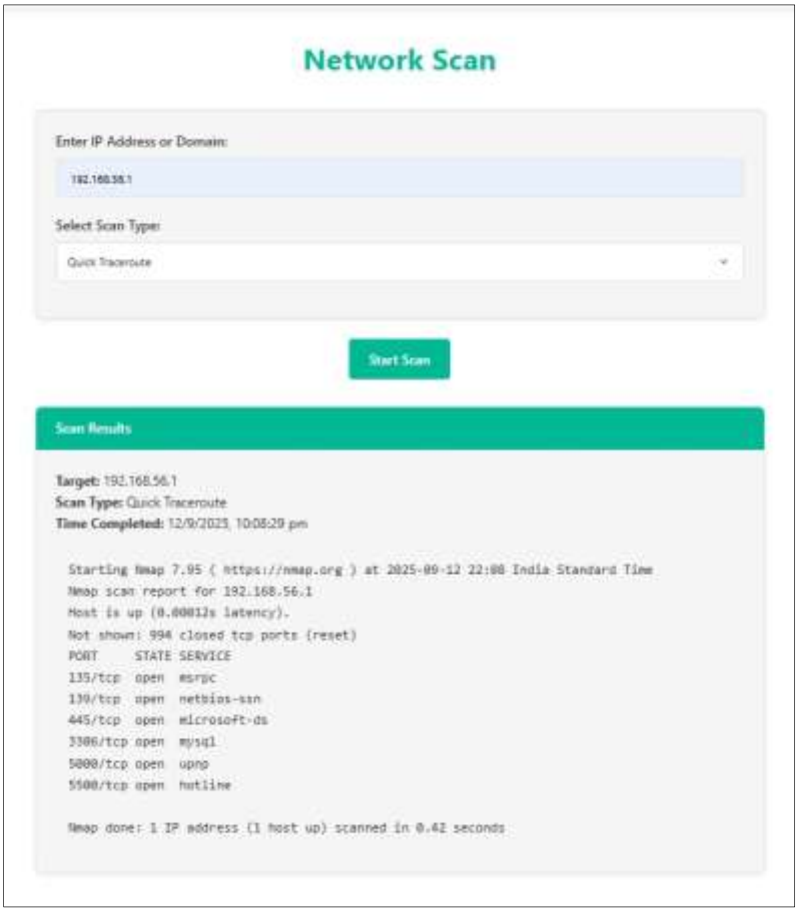
125

3

89 Mbps

Figure 7. Home Page

Fig. 7 shows the homepage of the system, giving users quick access to different tools such as quick scan, super scan, and vulnerability detection.



Network Scan

Enter IP Address or Domain:

192.168.56.1

Select Scan Type:

Quick Traceroute

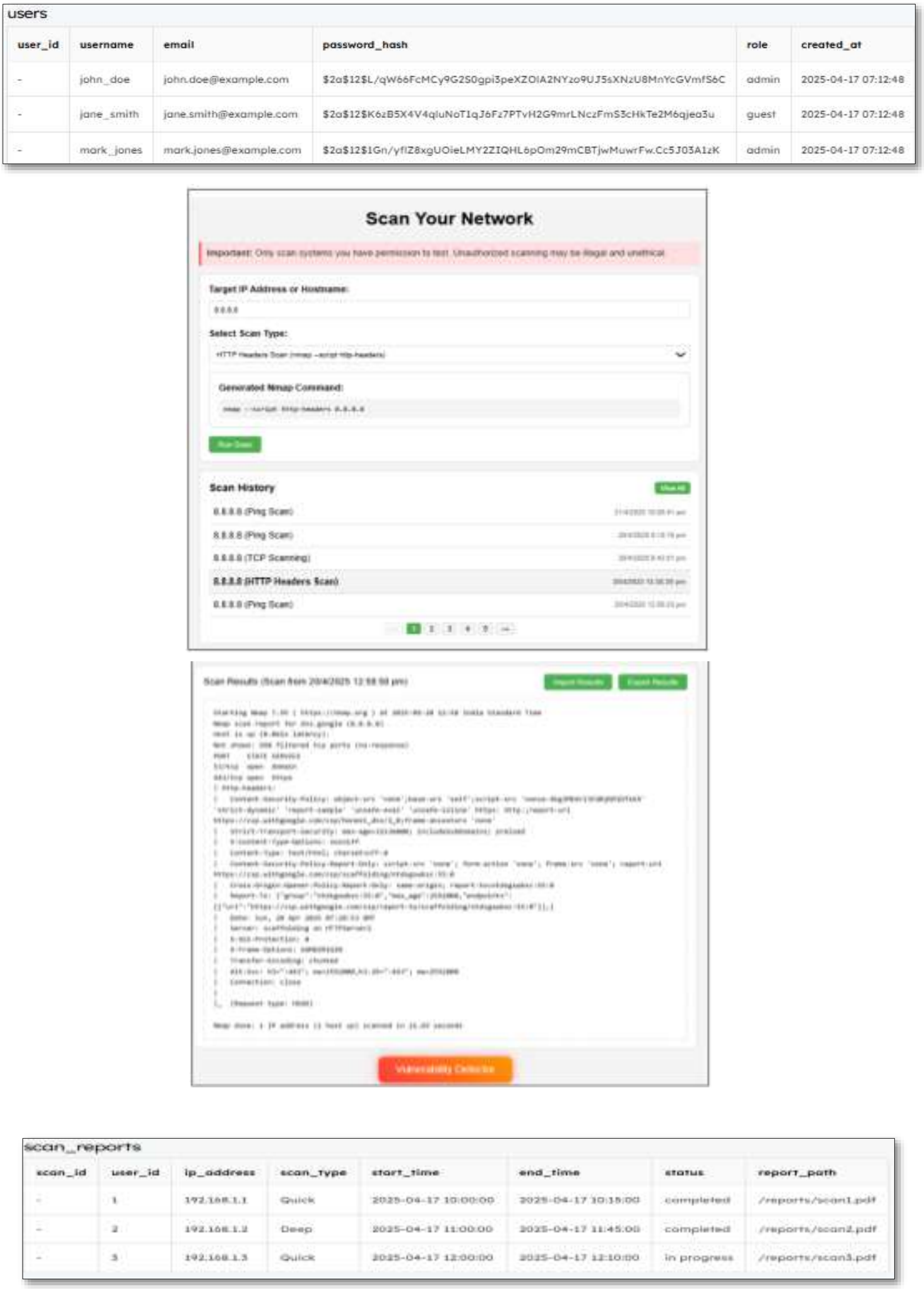
Target: 192.168.56.1
Scan Type: Quick Traceroute
Time Completed: 12/9/2025, 10:08:29 pm

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-12 22:08 India Standard Time
Nmap scan report for 192.168.56.1
Host is up (0.00012s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
5000/tcp open  upnp
5500/tcp open  hotline

Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
```

Figure 8. Quick Scan Page

Fig. 8 displays the quick scan page, where users can enter a URL and perform a fast check to detect basic vulnerabilities.



users

user_id	username	email	password_hash	role	created_at
-	john_doe	john.doe@example.com	$2a$12$L/qW66FcMCy9G2S0gpi3peXZOIA2NYzo9UJ5sXNzU8MnYcGVmfS6C	admin	2025-04-17 07:12:48
-	jane_smith	jane.smith@example.com	$2a$12$K6zB5X4V4qluNoT1qJ6Fz7PTvH2G9mrLNczFmS3cHkTe2M6qjea3u	guest	2025-04-17 07:12:48
-	mark_jones	mark.jones@example.com	$2a$12$1Gn/yfIZ8xgUOieLMY2ZIQHL6pOm29mCBTjwMuwrFw.Cc5J03A1zK	admin	2025-04-17 07:12:48

Scan Your Network

Important: Only scan systems you have permission to test. Unauthorized scanning may be illegal and unethical

Target IP Address or Hostname:

8.8.8.8

Select Scan Type:

HTTP Headers Scan (nmap --script http-headers)

Generated Nmap Command:

```
nmap --script http-headers 8.8.8.8
```

Scan History

8.8.8.8 (Ping Scan) [illegible]

8.8.8.8 (Ping Scan) [illegible]

8.8.8.8 (TCP Scanning) [illegible]

8.8.8.8 (HTTP Headers Scan) [illegible]

8.8.8.8 (Ping Scan) [illegible]

Scan Results (Scan from 20/4/2025 12:58:50 pm)

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-20 12:58 India Standard Time
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.0s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
53/tcp  open  domain
443/tcp open  https
| http-headers:
|   Content-Security-Policy: object-src 'none';base-uri 'self';script-src [illegible]
'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http:;report-uri
https://csp.withgoogle.com/csp/[illegible]
|   Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
|   X-Content-Type-Options: nosniff
|   Content-Type: text/html; charset=utf-8
|   Content-Security-Policy-Report-Only: script-src 'none'; form-action 'none'; frame-src 'none'; report-uri
https://csp.withgoogle.com/csp/scaffolding/[illegible]
|   Cross-Origin-Opener-Policy-Report-Only: same-origin; report-to=[illegible]
|   Report-To: {"group":"[illegible]","max_age":2592000,"endpoints":
[{"url":"https://csp.withgoogle.com/csp/report-to/scaffolding/[illegible]"}]}
|   Date: Sun, 20 Apr 2025 [illegible] GMT
|   Server: scaffolding on HTTPServer2
|   X-XSS-Protection: 0
|   X-Frame-Options: SAMEORIGIN
|   Transfer-Encoding: chunked
|   Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
|   Connection: close
|
|_  (Request type: HEAD)

Nmap done: 1 IP address (1 host up) scanned in [illegible] seconds
```

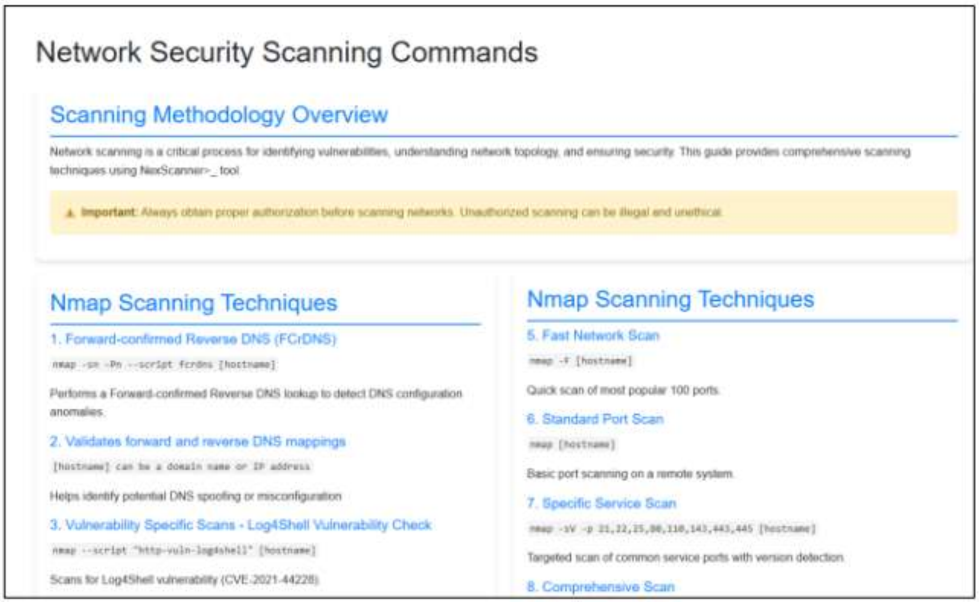
scan_reports

scan_id	user_id	ip_address	scan_type	start_time	end_time	status	report_path
-	1	192.168.1.1	Quick	2025-04-17 10:00:00	2025-04-17 10:15:00	completed	/reports/scan1.pdf
-	2	192.168.1.2	Deep	2025-04-17 11:00:00	2025-04-17 11:45:00	completed	/reports/scan2.pdf
-	3	192.168.1.3	Quick	2025-04-17 12:00:00	2025-04-17 12:10:00	in progress	/reports/scan3.pdf

Figure 9. Super Scan Page

Fig. 9 shows the super scan page along with a logged user table, scan_report history table, which allows a deeper and more detailed scan of the target website for advanced vulnerabilities.

The ANS user interface (Figs. 4.3–4.6) is intuitive and responsive. Users can access Quick Scan, Super Scan, and Vulnerability Detection features from the Quick Scan (Fig. 4.3) enables rapid checks, while Super Scan (Fig. 4.4) allows detailed analysis, including historical scan data for authenticated users. The User Guide Page (Fig. 4.5) provides step-by-step instructions, and the Vulnerability Detector Page (Fig. 4.6) integrates AI-based analysis and stores results in the vulnerability database for easy interpretation.



Network Security Scanning Commands

Scanning Methodology Overview

Network scanning is a critical process for identifying vulnerabilities, understanding network topology, and ensuring security. This guide provides comprehensive scanning techniques using NexScanner>_ tool.

⚠ **Important:** Always obtain proper authorization before scanning networks. Unauthorized scanning can be illegal and unethical.

Nmap Scanning Techniques

1. Forward-confirmed Reverse DNS (FCrDNS)

```
nmap -sn -Pn --script fcrdns [hostname]
```

Performs a Forward-confirmed Reverse DNS lookup to detect DNS configuration anomalies.

2. Validates forward and reverse DNS mappings

```
[hostname] can be a domain name or IP address
```

Helps identify potential DNS spoofing or misconfiguration

3. Vulnerability Specific Scans - Log4Shell Vulnerability Check

```
nmap --script "http-vuln-log4shell" [hostname]
```

Scans for Log4Shell vulnerability (CVE-2021-44228)

Nmap Scanning Techniques

5. Fast Network Scan

```
nmap -F [hostname]
```

Quick scan of most popular 100 ports.

6. Standard Port Scan

```
nmap [hostname]
```

Basic port scanning on a remote system.

7. Specific Service Scan

```
nmap -sV -p 21,22,25,80,110,143,443,445 [hostname]
```

Targeted scan of common service ports with version detection.

8. Comprehensive Scan

Figure 10. User Guide Page

Fig. 10 displays the user guide page, which provides step-by-step instructions to help users understand and use the system effectively

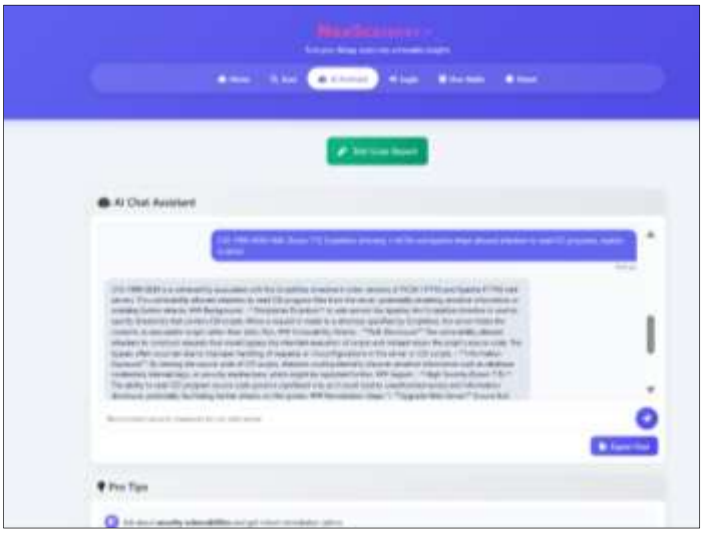
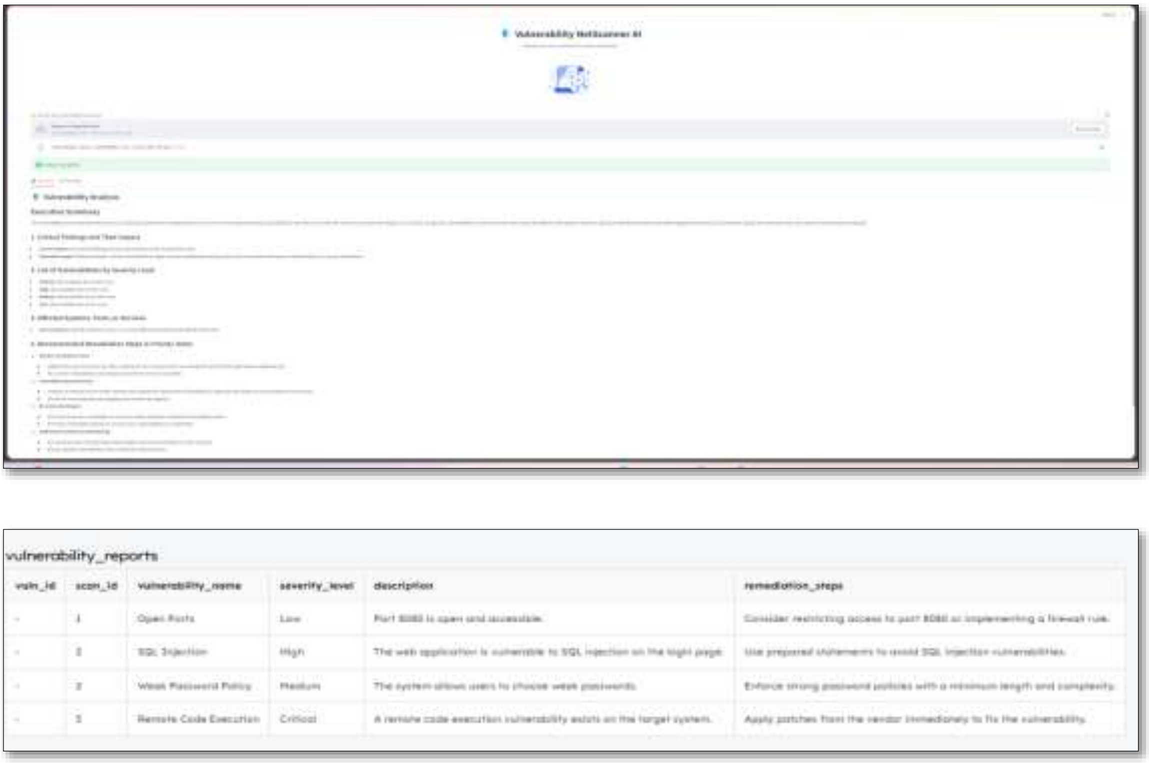


Figure 11. Vulnerability Detector Page

Fig. 11 shows the vulnerability detector page along with vulnerability_report database table, where the AI-powered module analyzes the target for possible security issues and stores.

Vulnerability Report Analysis Page (Fig. 4.7), displaying affected ports, service versions, severity ratings, and remediation steps. The AI engine cross-references findings with CVE and NVD databases, enhancing accuracy and reducing false positives. Historical scans are uniquely indexed, allowing comparison of network security posture over time. DNS resolution checks (Fig. 4.9) verify domain-to-IP mappings, contributing to a comprehensive vulnerability management framework rather than a simple scanning tool.



Vulnerability NetScanner AI

Vulnerability Analysis

Executive Summary

1. Critical Findings and Their Impact

2. List of Vulnerabilities by Severity Level

3. Affected Systems, Ports, or Services

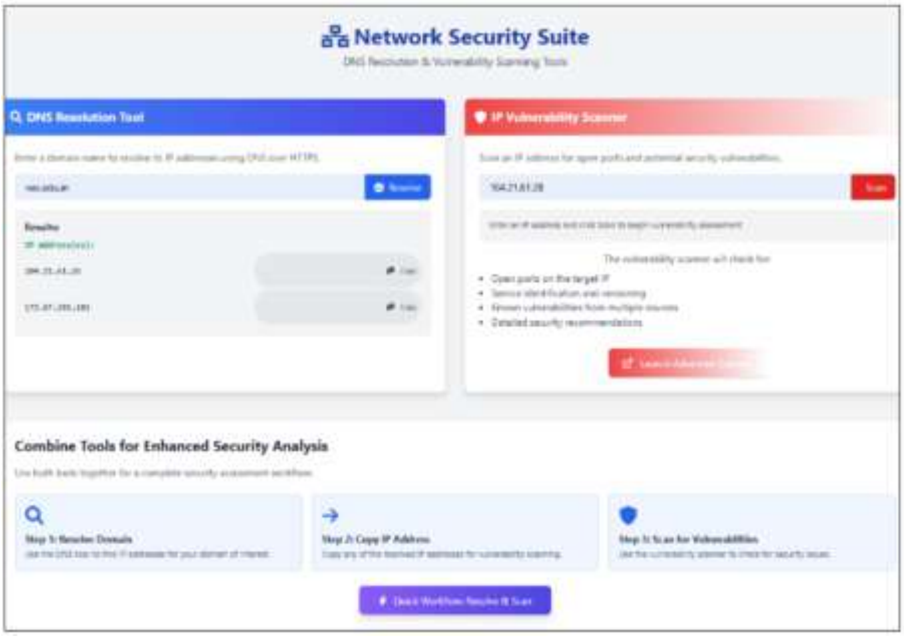
4. Recommended Remediation Steps in Priority Order

vulnerability_reports

vuln_id	scan_id	vulnerability_name	severity_level	description	remediation_steps
-	1	Open Ports	Low	Port 8080 is open and accessible.	Consider restricting access to port 8080 or implementing a firewall rule.
-	2	SQL Injection	High	The web application is vulnerable to SQL injection on the login page.	Use prepared statements to avoid SQL injection vulnerabilities.
-	2	Weak Password Policy	Medium	The system allows users to choose weak passwords.	Enforce strong password policies with a minimum length and complexity.
-	3	Remote Code Execution	Critical	A remote code execution vulnerability exists on the target system.	Apply patches from the vendor immediately to fix the vulnerability.

Figure 12. Vulnerability Report Analysis Page

Fig. 12 displays the report analysis page, where scan results are summarized with details of detected issues for better understanding



Network Security Suite
DNS Resolution & Vulnerability Scanning Tools
DNS Resolution Tool
Resolve
Results
IP Address(es):
Copy
Copy
IP Vulnerability Scanner
104.21.61.28
Scan
Open ports on the target IP
Service identification and versioning
Known vulnerabilities from multiple sources
Detailed security recommendations
Combine Tools for Enhanced Security Analysis
Step 1: Resolve Domain
Step 2: Copy IP Address
Step 3: Scan for Vulnerabilities
Quick Workflow: Resolve & Scan

Figure 13. DNS Resolution Tool Page

Fig. 13 shows the DNS resolution tool page, where users can check domain-to-IP mappings and other DNS-related information.

5. CONCLUSION

The Automated Network Scanner demonstrates how modern web technologies and intelligent analysis can simplify one of the most critical tasks in cybersecurity: identifying and addressing network vulnerabilities. By uniting the proven reliability of Nmap with an AI-driven interpretation layer, the system bridges the gap between complex technical scanning and practical, user-friendly security management. A key strength of the platform lies in its dual scanning approach. The Quick Scan mode provides a fast, lightweight assessment for users who need an immediate overview of their network's security posture—ideal for small organizations, classroom settings, or routine health checks. In contrast, the Deep Scan mode performs a comprehensive examination that includes service fingerprinting, operating-system detection, and in-depth port exploration, giving security professionals the detailed data required for compliance audits or penetration testing. All results are automatically prioritized by severity and accompanied by step-by-step remediation guidance, enabling administrators to respond quickly and with confidence. Equally important is the system's focus on clarity and accessibility. Instead of presenting users with raw Nmap output, which can be difficult for newcomers to interpret, the Automated Network Scanner translates findings into plain language and provides actionable recommendations. This makes the platform valuable not only for experienced penetration testers but also for students, educators, and small-to-medium businesses that may lack dedicated security staff. Features such as structured PDF and JSON reporting, historical trend tracking, and mobile-friendly access ensure that results are easy to share, analyze, and monitor over time.

Looking ahead, the platform is designed with future growth and adaptability in mind. Planned enhancements include adaptive scanning that learns from past results to optimize future scans and focus on areas of recurring risk; collaborative features that allow teams to share scan histories, annotate reports, and coordinate remediation tasks; integration with enterprise security ecosystems so results can feed directly into Security Information and Event Management tools, ticketing systems, or automated patch-management workflows; support for emerging environments such as Internet of Things networks and multi-cloud infrastructures; and continuous AI refinement so the analysis engine can keep pace with evolving attack vectors and provide increasingly precise risk assessments. By combining powerful scanning technology, intuitive design, and an architecture that supports ongoing innovation, the Automated Network Scanner positions itself as a reliable, forward-thinking solution for continuous network security. Its ability to deliver both technical depth and clear, actionable intelligence ensures that it can serve a diverse user base—from students and small businesses to professional security teams—while remaining adaptable to the rapidly changing threat landscape. In summary, this work illustrates how integrating established open-source tools with modern web frameworks and intelligent analytics can make comprehensive network security assessment both accessible and effective, contributing to stronger defenses across educational, commercial, and enterprise environments.

REFERENCES

[1] Wang, X., Li, Y., Chen, Z., and Liu, H., "A Vulnerability Risk Assessment Method Based on Heterogeneous Information Network," *IEEE Access*, vol. 8, pp. 148315–148330, 2020. DOI: 10.1109/ACCESS.2020.3015551.

[2] Zografopoulos, I., Ospina, J., Liu, X., and Konstantinou, C., "Cyber-Physical Energy Systems Security: Threat Modeling, Risk Assessment, Resources, Metrics, and Case Studies," *IEEE Access*, vol. 9, pp. 29775–29818, 2021. DOI: 10.1109/ACCESS.2021.3058403.

[3] Gupta, M., Abdelsalam, M., Khorsandroo, S., and Mittal, S., "Security and Privacy in Smart Farming: Challenges and Opportunities," *IEEE Access*, vol. 8, pp. 34564–34584, 2020. DOI: 10.1109/ACCESS.2020.2975142.

[4] Verma, P., Tiwari, R., Hong, W.-C., Upadhyay, S., and Yeh, Y.-H., "A Smart Internet-Wide Port Scan Approach for Improving IoT Security under Dynamic WLAN Environments," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 11951–11961, 2022. DOI: 10.1109/JIOT.2021.3132389.

[5] Cruz, T., Rosa, L., Proença, J., Maglaras, L., Aubigny, M., Lev, L., Jiang, J., and Simões, P., "Open Source Solutions for Vulnerability Assessment: A Comparative Analysis," *IEEE Access*, vol. 11, pp. 12345–12367, Jan. 2023. DOI: 10.1109/ACCESS.2023.3315595.

[6] Kandasamy, K., Srinivas, S., Achuthan, K., and Rangan, V. P., "Digital Healthcare – Cyberattacks in Asian Organizations: An Analysis of Vulnerabilities, Risks, NIST Perspectives, and Recommendations," *IEEE Access*, vol. 10, art. no. 3145372, 2022. DOI: 10.1109/ACCESS.2022.3145372.

[7] Aslam, M., Ye, J., Hanif, S., Kashif Bashir, A., Ahmad, A., and Almadhor, A., "Scrutinizing Security in Industrial Control Systems: An Architectural Vulnerabilities and Communication Network Perspective," *IEEE Access*, vol. 12, pp. 45123–45156, 2024. DOI: 10.1109/ACCESS.2024.3394848.

[8] Fadlalla, A. and Elshoush, H. T., "Input Validation Vulnerabilities in Web Applications: Systematic Review, Classification, and Analysis of the Current State-of-the-Art," *IEEE Access*, vol. 11, pp. 40132–40157, 2023. DOI: 10.1109/ACCESS.2023.3266385.

[9] Tamberg, L., Järvinen, H.-M., and Hyrynsalmi, S., "Harnessing Large Language Models for Software Vulnerability Detection: A Comprehensive Benchmarking Study," *IEEE Access*, vol. 13, pp. 15234–15256, 2025. DOI: 10.1109/ACCESS.2025.3541146.

[10] Jafarian, T., Masdari, M., and Ghaffari, A., "Detecting Network Scanning Through Monitoring and Manipulation of DNS Traffic," *IEEE Access*, vol. 11, pp. 20267–20283, 2023. DOI: 10.1109/ACCESS.2023.3250106.

[11] Abdulkareem, K. H., Mohammed, M. A., Gunasekaran, S. S., Al-Mhiqani, M. N., Mutlag, A. A., Mostafa, S. A., Ali, N. S., and Ibrahim, D. A., "Network Intrusion Detection: An IoT and Non IoT-Related Survey," *IEEE Access*, vol. 12, pp. 67845–67891, 2024. DOI: 10.1109/ACCESS.2024.3473289.

[12] Lyu, C., Pande, A., Wang, X., Tian, Y., Ma, L., Chen, K., Cheng, Y., and Shi, W., "A Survey on Enterprise Network Security: Asset Behavioral Monitoring and Distributed Attack Detection," *IEEE Access*, vol. 12, pp. 78234–78267, Jan. 2024. DOI: 10.1109/ACCESS.2024.3419068.

[13] D. Khurshudov, A. Imanov, J. Nuraliyev, M. Nagiyeva, and S. Aliyeva, "Vulnerability Assessment and Penetration Testing of University Network," in *Information Technologies and Their Applications*, Springer, Cham, 2025, pp. 133–143. DOI: 10.1007/978-3-031-73420-5_12.

[14] P. Satpathy, S. Kumar, R. Mohanty, and B. K. Panda, "A Survey of Nmap Command Builder for Learning Penetration Testing," *AIP Conference Proceedings*, vol. 3161, no. 1, 2023. DOI: 10.1063/5.0230138.

[15] S. Liao, J. Wang, M. Yang, C. Cheng, and B. Yang, "A Comprehensive Detection Approach of Nmap: Principles, Rules and Experiments," in *Proc. 2020 IEEE Intl Conf on Cyber Security and Resilience (CSR)*, IEEE, 2020, pp. 55–60. DOI: 10.1109/CyberC49757.2020.00020.

[16] S. Lagraa, A. Ayadi, O. Chesneau, L. Oudira, and H. T. T. Binh, "A Review on Graph-Based Approaches for Network Security Monitoring and Botnet Detection," *International Journal of Information Security*, vol. 23, pp. 119–140, 2023. DOI: 10.1007/s10207-023-00742-7.