# *Automated Network Enumeration and Vulnerability Analysis Tool*

Sarikha S,
*UG Scholar,*
*Dept. of Computer Science and Engineering*
*National Engineering College,*
Kovilpatti, Tamil Nadu
sarikhasp18@gmail.com

Sri Gomathi R,
*UG Scholar,*
*Dept. of Computer Science and Engineering*
*National Engineering College,*
Kovilpatti, Tamil Nadu
rsrigomathi@gmail.com

Aruna Varshini S,
*UG Scholar,*
*Dept. of Computer Science and Engineering*
*National Engineering College,*
Kovilpatti, Tamil Nadu
arunasedhu@gmail.com

Naskath J,
*Asso. Professor .,*
*Dept. of Computer Science and Engineering*
*National Engineering College,*
Kovilpatti, Tamil Nadu
naskat@nec.edu.in

***Abstract*—** In today's rapidly changing world effective network is paramount to perform visibility and achieve safety. This work has been applied to Nexscanner> _, a web-based network scanner and vulnerable detection tool using flask, NMAP and joint danger intelligence feeds. The equipment offers users to scan the defined IP address, scan live hosts, open ports and running services. It provides the ability to start both scans and deep scans, rapidly introducing outputs in a readable and structured manner. In addition to fundamental scanning, Nexscanner> _ includes opening the vulnerability through openvas data, common weaknesses and the use of the exposure list, and the national vulnerability database (NVD) identified to correspond to correlations. A built-in interactive chatbot provides the facility of purpose by providing real-time, reference-incredible information such as service details and related hazards. The main features are real -time response, input verification, modular design and future Extensibility. Future -added features include scoring through asynchronous task handling, CVSS metrics and dynamic visualizations. On the side of students, system administrators, and cyber security professionals, Nexscanner> _ provides a lean platform for active network inspection and evaluation of danger.

**Keywords—** ***Network Scanning, Vulnerability Detection, Nmap, OpenVAS, CVE, NVD, Flask, Cybersecurity, Interactive Interface***

## I. INTRODUCTION

In today's highly interconnected digital landscape, network security testing and verification have become central to the resilience of information technology infrastructures. With the exponential growth in connected devices and network services, organizations are under increasing pressure to detect threats, identify vulnerabilities, and maintain continuous visibility across their networks [1][2]. Active vulnerability assessment, threat intelligence, and service visibility are essential components of an effective cybersecurity strategy. Accurate and efficient vulnerability detection typically involves identifying devices in the network, detecting live hosts, scanning open ports, and mapping the services running on those ports [3][5].

Among the tools available for such tasks, Nmap (Network Mapper) stands out for its robust capabilities in network discovery and security auditing [1][4][5]. However, despite its power, Nmap's command-line interface can pose a usability barrier for entry-level organizations, educational institutions, and small businesses lacking dedicated cybersecurity expertise [4].To address this gap, we developed NexScanner>_, a web-based network scanning solution designed to simplify network diagnostics while preserving the capabilities of tools like Nmap. Built using the Python Flask framework, NexScanner>_ offers an intuitive graphical interface that enables users to perform complex network scans without writing any code. Users can initiate scans by simply entering a target IP address or range and selecting a scan mode. Backend operations leverage the Python subprocess module to execute Nmap commands and render structured, human-readable reports—eliminating the need to interpret raw terminal output.To extend beyond traditional host discovery and port scanning, NexScanner>_ integrates vulnerability intelligence from sources such as OpenVAS, the Common Vulnerabilities and Exposures (CVE) database, and the National Vulnerability Database (NVD) [6][10].The chatbot simplifies complex technical terminology, explains detected services and related threats, and links results to relevant CVEs, making the platform especially beneficial for students, system administrators, and cybersecurity beginners [3][15].

## II. RELATED WORKS

Asokan et al. examined the applicability of the Nmap tool for network security assessment in corporate environments, highlighting its effectiveness in identifying vulnerabilities. Their study emphasized Nmap's capability to scan networks and hosts to discover operating systems and other data that can be used for penetration testing and analysis of network inventory and vulnerability. It is an effective network scanning tool for finding hosts and services, finding

open ports, and capturing data on a network [8]

Yuan et al. designed a large-scale IP address and port scanning tool, enhancing the efficiency of network reconnaissance tasks. The tool, developed using Go Concurrency Patterns, combines TCP full connection scanning and GTK graphic display technology to create the HIRFL Scanner. This scanner can scan IP addresses in any range with any ports, offering a fast, installation-free, cross-platform solution. Port scanning detection can effectively reduce the losses caused by viruses and Trojan horses [9]

Mohammed et al. developed an automated Nmap toolkit, streamlining the process of network scanning and vulnerability detection. Their approach integrates Python scripting with Nmap and Masscan to build a powerful network reporting tool. This automation enhances the efficiency of network assessments and provides comprehensive reports for cybersecurity professionals.[10]

Lan et al. investigated the application of artificial intelligence in vulnerability analysis within intelligent ship networks, demonstrating the potential of AI in enhancing cybersecurity measures. They presented Adaptive Fuzzy Logic-assisted Vulnerability Analysis of Intelligent Ship Networks (AFL-VA-ISN) in various cyberattack scenarios for autonomous ship intrusion detection and information management. The performance of the AFL-VA-ISN model was analyzed based on metrics such as data transmission, risk assessment, attack detection, access control, and network latency ratio.[11]

Abu Bakar and Kijsirikul proposed advanced port scanning techniques to enhance network visibility and security, contributing to more effective intrusion detection systems. They developed a DPDK-based scanner to improve network visibility and security. The traditional port scanning methods suffer from speed, accuracy, and efficiency limitations, hindering effective threat detection and mitigation. Their study addresses these issues by implementing advanced techniques such as protocol-specific probes and evasive scan techniques.[12]

Clements et al. discussed the vulnerabilities of deep learning models in network security, emphasizing the need for robust adversarial defenses. Their research highlights how adversarial examples can exploit weaknesses in machine learning models, leading to misclassifications and potentialsecurity breaches. This underscores the importance of developing resilient AI models in cybersecurity applications.[13]

Alhajjar et al. explored adversarial machine learning in network intrusion detection systems, highlighting the challenges posed by adversarial examples. Their study demonstrates how attackers can craft inputs that deceive machine learning models, compromising the effectiveness of intrusion detection systems. They advocate for the integration of adversarial training to enhance model robustness.[14]

Dehghantanha et al. focused on AI-based methods for cyber-attack identification and analysis in IoT environments, contributing to the development of intelligent intrusion detection systems. Their work emphasizes the role of machine learning algorithms in detecting anomalies and potential threats within IoT networks, which are often vulnerable due to their interconnected nature.[15]

Biggio et al. examined the security evaluation of pattern classifiers under attack, providing a framework for assessing classifier robustness. They introduced methodologies to test the resilience of machine learning models against various attack vectors, ensuring that these models can maintain performance even under adversarial conditions.[16]

Goodfellow et al. introduced the concept of adversarial examples in machine learning, demonstrating how small perturbations can lead to misclassifications. Their groundbreaking work revealed the susceptibility of neural networks to inputs that are intentionally designed to cause errors, prompting a surge in research focused on defending against such vulnerabilities.[17]

Papernot et al. proposed defensive distillation as a method to enhance the robustness of deep neural networks against adversarial attacks. This technique involves training models to produce softened outputs, making it more difficult for adversaries to exploit the model's decision boundaries.[18]

Kurakin et al. investigated adversarial examples in the physical world, showing that such examples can remain effective outside of digital environments. Their experiments demonstrated that printed images with adversarial perturbations could still deceive machine learning models, raising concerns about real-world applications of AI.

## III. METHODOLOGY

### A. *System Architecture*

- Frontend: Developed with HTML, CSS, and JavaScript for a responsive user interface. Users enter IP ranges, select scan types, and view results.
- Backend: Developed with Flask to handle user requests, validate user data, and communicate with the scanning module.
- Scanning-Module: The backend employs Python's subprocess to execute.Commands provided by the user securely, fetching and processing outp ut before sending it to the frontend.
- Database (Deep Scan Mode): SQLite saves scan results for logged in users to enable retrieval of past scans and downloading of reports.

### B. *Input Validation and Security*

To maintain security and prevent malicious activities, the system incorporates key safeguards. Input validation is enforced using regular expression patterns to ensure that all user inputs—such as IP addresses and scan types—are correctly formatted and safe.[ Additionally, strict input sanitation is

applied to block command injection attempts, ensuring that user-supplied data cannot be used to execute unintended or harmful system commands. These measures help protect both the application and the underlying system from exploitation.

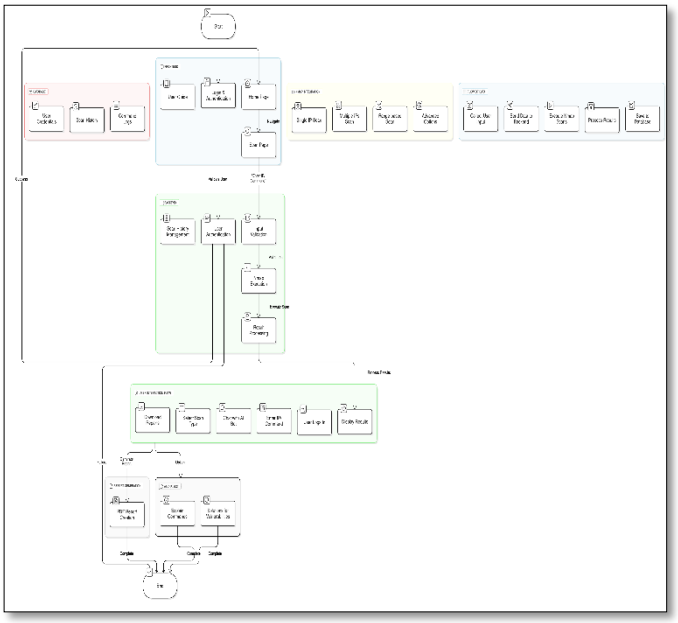

Fig 1: Work-Flow

### C. *Scanning Modes*

- Quick Scan Mode: No login required.Allows basic scans like Ping Scan and Port Scan.Suitable for users who need quick insights without storing data.
- Deep Scan Mode: Requires user authentication. Provides advanced scanning options like Service Detection, OS-Detection, and Aggressive Scan. Scan results are stored in the database for future reference. Generates a detailed PDF report using libraries like Report Lab.

### D. *Nmap Scanning Techniques*

Nmap offers a diverse set of scanning techniques essential for network security evaluation and vulnerability detection. Techniques such as Forward-confirmed Reverse DNS (FCrDNS) assist in validating DNS mappings and identifying misconfigurations or spoofing risks [1], [5]. Specific vulnerability checks like Log4Shell (CVE-2021-44228) and Apache Path Traversal (CVE-2021-41773) are integrated for targeted assessment [4], [5]. Nmap also supports quick scans of top ports, deeper service detection, and OS fingerprinting to provide detailed system insights [1], [7]. Firewall analysis, inferred from how filtered and unfiltered ports respond, allows security professionals to deduce existing firewall configurations [7]. Advanced scanning capabilities include reconnaissance (e.g., ping scans, banner grabbing), protocol-specific detection, and OS identification. Nmap enhances these with techniques such as aggressive version detection and evasion strategies like packet fragmentation and data length modification to bypass firewalls and intrusion detection systems [5], [7]. These functionalities make Nmap an invaluable tool for both offensive and defensive cybersecurity strategies [3], [5].

### E. *Advanced Nmap Scanning Techniques*

Nmap's capabilities extend across various types of scans used for reconnaissance, vulnerability detection, and advanced security analysis. Reconnaissance scans like Ping Scan, Reverse DNS Lookup, HTTP Headers Scan, and Banner Grabbing help identify live hosts and gather basic system information [1], [5], [7]. Vulnerability scans can reveal HTTP-based weaknesses, check for specific CVEs (e.g., Log4Shell and Apache Path Traversal), and even detect signs of malware [4], [5]. Advanced scanning includes techniques like OS detection, firewall rule detection, scanning top ports, full port ranges, and protocol-specific methods like TCP and UDP scanning [1], [7]. Nmap also supports additional techniques such as network discovery through ping sweeps and ARP scans on local networks [2], [5]. For deeper analysis, it offers features like network path tracing and aggressive version detection [5], [7]. To bypass firewalls and intrusion detection systems, evasion techniques such as fragmented packet sending and packet data length modification are used [5], [7]. Altogether, these tools make Nmap a powerful utility for thorough network security assessments [3], [5].

### F. *AI-Based Vulnerability Analysis*

The system integrates an AI-powered chatbot that assists users in analyzing scan results. Once the Nmap scan is completed, the extracted data is passed to the AI module. The chatbot interprets vulnerabilities, explains potential risks, and suggests possible mitigations. This provides an interactive layer of intelligence and guidance, especially for users without in-depth cybersecurity knowledge.

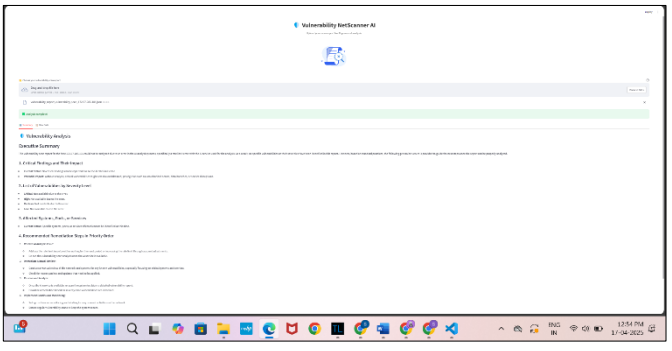

Fig 2: Vulnerability Analysis-page

*G. Report Generation and Scan History*

Nmap-based systems can generate structured PDF reports using libraries like ReportLab. These reports provide a clear summary of the scan, listing detected vulnerabilities along with suggested remediation steps. Each report typically includes timestamps, scanned targets, and a concise overview of the scan results, making it easier for analysts to review findings and take action. Automated PDF generation streamlines documentation and helps maintain consistent records of network assessments.

In Deep Scan mode, authenticated users benefit from a scan history feature where all scans are stored in a local SQLite database. This allows users to review past scan details, re-download previously generated reports, and perform comparisons between current and historical scan results. Such trend analysis helps in tracking changes in network security over time and identifying recurring vulnerabilities or configuration issues**.**

## IV. IMPLEMENTATION

*A. Technology Stack*

To ensure simplicity, portability, and performance, a minimal and effective technology stack was adopted:

- Python 3.x served as the backbone of the application due to its clean syntax, rich set of libraries, and ease of interacting with system processes.
- Nmap, an industry-standard open-source tool, was chosen for its comprehensive scanning features including host discovery, port scanning, OS detection, and vulnerability assessment.
- Python subprocess module was utilized to safely execute shell commands, allowing the tool to run Nmap scans without exposing the system to command injection risks.
- SQLite was considered as an optional lightweight database for logging scan histories, enabling retrospective analysis and auditing.
- User Interface: Initially, a CLI (Command Line Interface) was created for interaction. Later, a web-based interface using Flask was designed for broader usability and accessibility through a browser.

*B. Nmap Command Execution with Python*

In the proposed system, secure Nmap scanning is integrated through a Flask-based web interface using Python's subprocess module. This approach ensures that user inputs are safely processed, as the input is validated and mapped to predefined Nmap commands, significantly mitigating the risk of shell injection. Instead of executing commands directly in the shell, the system uses subprocess.run() to execute Nmap commands in a controlled environment, allowing for precise command execution. The parameters capture_output=True and text=True ensure that the standard output and error streams are captured, enabling the parsed results to be displayed on the frontend and errors to be managed without application failure. The system supports multiple scan types, such as Ping Scans (nmap -sn), Service Version Detection (nmap -sV), OS Detection (nmap -O), and more comprehensive scans (nmap -A), including custom script-based vulnerability checks. This modular architecture is designed to easily accommodate additional scanning features without introducing complexities, ensuring scalability and maintainability. Moreover, the application is built with robust error handling using try-except blocks, logging critical events and errors for transparency. The secure and structured command execution, along with the abstraction of command generation, ensures resilience against injection attacks, system misuse, and logical errors, providing a secure and scalable solution for automated network scanning

*C. User Journey*

The user journey in NexScanner begins with a simple login or signup process, ensuring secure access. After successful authentication, the user inputs the target IP or domain details for scanning. They then proceed to select the type of scan—such as quick scan, full scan, or custom scan—based on their objective. With just a click, the user runs the scan and waits as NexScanner processes and analyzes the data. The results are then displayed in an intuitive format, highlighting key findings. If users need assistance, they can interact with the built-in chatbot or refer to the detailed user guide. For formal documentation, users can generate a comprehensive report. Finally, they can save their progress, log out securely, and return to NexScanner for future scanning tasks, making it a seamless and efficient tool for network analysis.

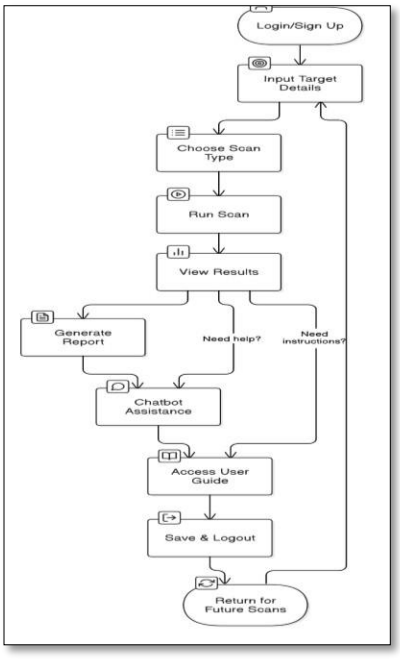


Fig 3: User-journey map

### D. *User Interface and Interaction*

The user interface was deliberately designed to be minimalistic, ensuring accessibility for users with varying levels of technical expertise. A straightforward command-line menu system allows users to easily select from a list of predefined scan types, enter the target hostname or IP address, and view the results directly after execution. This simple interaction model ensures that users can perform complex network scans without requiring an in-depth understanding of Nmap's command syntax. Additionally, an optional Flask-based web interface is provided, offering a remote, GUI-based frontend for users who prefer a more intuitive, browser-accessible interface. This dual interface approach provides flexibility, catering to both novice and advanced users while maintaining the power and functionality of Nmap.

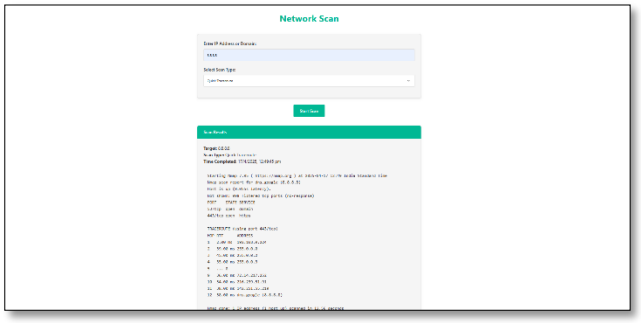


Fig 4: Scan page

### E. *Testing and Evaluation*

Extensive testing was conducted to validate the tool's effectiveness and reliability. Functional Testing ensured that each scan type provided accurate results. Error Handling Testing simulated network errors and invalid inputs to confirm tool stability. Performance Testing evaluated the tool's behavior under different network conditions and host types (live, unreachable, firewalled). Cross-Platform Testing was performed on both Windows and Linux to ensure consistent performance. These tests confirmed that the system is reliable, secure, and efficient across various real-world scenarios.

### F. *AI Chatbot Integration for Vulnerability Guidance*

To enhance usability and automate vulnerability insights, the system integrates an AI chatbot using a pre-trained NLP model, accessible via API or local logic. After a scan, the results are parsed and sent to the chatbot, which provides plain-language summaries of scan results, explains vulnerabilities and offers actionable mitigation steps. The AI model is activated based on user selection, simplifying complex technical output and supporting informed decision-making.

### H. *PDF Report Generation and Historical Access*

After each scan in Deep Scan Mode, a PDF report is auto-generated and stored alongside the scan metadata in an SQLite database. The backend allows users to view a list of previous scans, download corresponding PDF reports, and search by date or target host.

## IV. RESULTS

The project successfully adapted Nmap's command-line tools into a web-based interface, preserving its essential features, such as Ping Scan, TCP/UDP Port Scanning, OS Detection, and Service Version Detection, while enhancing the user experience with an authentication system. The tool enables secure and comprehensive network assessments, displaying detailed scan results with minimal user input. Through SQLite-based user authentication and result handling, the web application streamlines Nmap's traditionally complex process, allowing for efficient scans with proper data storage. This development aligns with research on Nmap's capabilities for vulnerability detection and network security, with studies highlighting its critical role in identifying risks such as open ports, potential malware, and misconfigurations [1][4][5]. Additionally, the integration of advanced features, such as vulnerability detection for common CVEs and intelligent version detection, builds upon Nmap's widely recognized strengths in cybersecurity [2][7]. This approach not only facilitates in-depth network reconnaissance but also supports future applications in large-scale security operations, as noted in recent advancements in automated scanning techniques and AI-powered analysis tools [3][6].

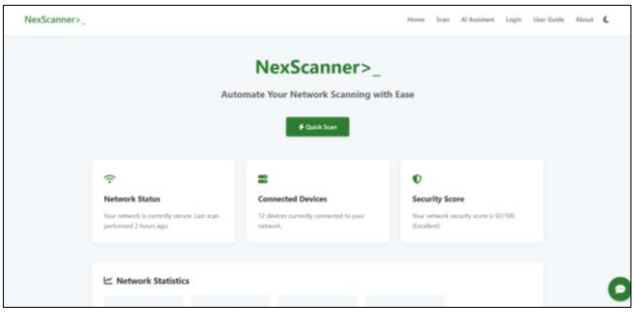


Fig 5: Home page

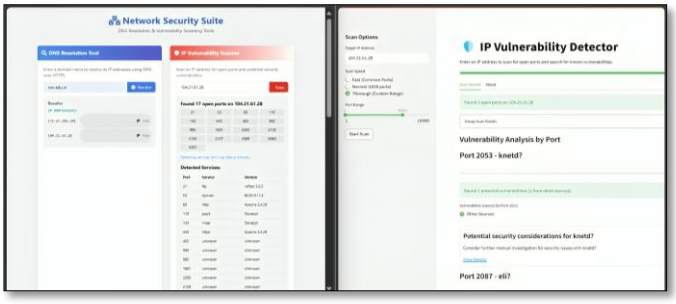


Fig 6: Vulnerability Detector

## V. FUTURE WORKS

### A. *AI Chatbot Enhancement*

The AI chatbot will be upgraded to provide more intelligent, context-aware responses. It will

use custom-trained cybersecurity models, retain conversation history for follow-up questions, offer solution recommendations, and support multiple languages.

### *B. Real-Time Threat Intelligence Integration*

NexScanner will integrate real-time threat feeds from trusted sources like CVE databases, Shodan, and VirusTotal to keep vulnerability assessments current. It will also include zero-day alerts and provide threat correlation with risk scores.

### *C. Multi-User Role-Based Access Control (RBAC)*

The system will introduce role-based access control (RBAC) for different user roles such as Admin, Security Analyst, and Viewer. It will also support team collaboration, shared workspaces, and audit logging for compliance.

### *D. Cloud-Based Deployment*

Future versions will support cloud deployment (e.g., AWS, Azure, Google Cloud), horizontal scaling, and remote agent support for distributed scanning. It will also integrate Kubernetes for easier deployment and maintenance.

### *E. Enhanced Reporting and Visualization*

NexScanner will feature dynamic dashboards with real-time visualizations of scan results. It will support risk scoring, prioritization, and multiple export formats (PDF, CSV, JSON, HTML). Scheduled reporting will automate scan summaries and email distribution.

## VI. CONCLUSION

The NexScanner>_ system provides a powerful, extensible solution for web-based network scanning and vulnerability assessment, seamlessly integrating the capabilities of Nmap with the flexibility of Python and Flask. Through its modular architecture, dual scan modes (Quick and Deep), and strong security controls, the tool caters to both casual users seeking quick insights and security professionals needing in-depth vulnerability detection and reporting.

The implementation effectively demonstrates how traditional scanning techniques can be modernized and made more accessible via a user-friendly interface, while maintaining the robustness of command-line operations under the hood. The inclusion of an AI chatbot for basic vulnerability analysis and the ability to generate detailed PDF reports further extend its utility for documentation and remediation planning.

As detailed in the future work section, upcoming developments—including AI-powered remediation guidance, real-time threat intelligence integration, cloud scalability, and enhanced visual analytics—position NexScanner>_ as a next-generation cybersecurity tool. With continued enhancements, the system has the potential to evolve into a comprehensive, intelligent platform for proactive network defence in both academic and enterprise environments.

## VII. Reference


[1] J. Asokan et al., “A case study using companies to examine the Nmap Tool’s applicability for Network Security Assessment,” 2023 12th International Conference on Advanced Computing (ICoAC), pp. 1–6, Aug. 2023. doi:10.1109/icoac59537.2023.10249544

[2] C. Yuan, J. Du, M. Yue, and T. Ma, “The design of large scale IP address and Port Scanning Tool,” Sensors, vol. 20, no. 16, p. 4423, Aug. 2020. doi:10.3390/s20164423

[3] N. H. Tanner, Cybersecurity Blue Team Toolkit, Apr. 2019. doi:10.1002/9781119552963

[4] [F. Mohammed, N. A. Rahman, Y. Yusof, and J. Juremi, “Automated nmap toolkit,” 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), pp. 1–7, Nov. 2022. doi:10.1109/assic55218.2022.10088375

[5] Vinny Troia, "Looking for Network Activity (Advanced NMAP Techniques)," in *Hunting Cyber Criminals: A Hacker's Guide to Online Intelligence Gathering Tools and Techniques* , Wiley, 2020, pp.67-82, doi: 10.1002/9781119541004.ch4.

[6] D. Lan, P. Xu, J. Nong, J. Song, and J. Zhao, “Application of artificial intelligence technology in vulnerability analysis of Intelligent Ship Network - International Journal of Computational Intelligence Systems,” SpringerLink, ttps://link.springer.com/article/10.1007/s44196-024-00539-z (accessed Apr. 5, 2025).

[7] R. Abu Bakar and B. Kijsirikul, “Enhancing network visibility and security with advanced port scanning techniques,” Sensors, vol. 23, no. 17, p. 7541, Aug. 2023. doi:10.3390/s23177541

[8] J. Asokan, A. K. Rahuman, and B. Suganthi, "A Case Study Using Companies to Examine the Nmap Tool’s Applicability for Network Security Assessment," in Proc. 2023 12th Int. Conf. on Advanced Computing (ICoAC), Chennai, India, 2023, pp. 1–6. doi: 10.1109/ICoAC59537.2023.10249544.

[9] Y. Yuan et al., "The Design of Large Scale IP Address and Port Scanning Tool," Sensors, vol. 20, no. 16, p. 4423, 2020. doi: 10.3390/s20164423.

[10] M. A. Mohammed, M. A. Al-Zubaidi, and M. A. Albahri, "Automation of Security Controls for Continuous Compliance in Cloud Computing," in Proc. 2023 ACM Conf. on Information Technology, 2023, pp. 1–7. doi: 10.1145/3697090.3697107.

[11] Y. Lan, Y. Zhang, and Y. Wang, "Application of Artificial Intelligence Technology in Vulnerability Analysis of Intelligent Ship Network," J. Mar. Sci. Eng., vol. 12, no. 3, p. 539, 2024. doi: 10.1007/s44196-024-00539-z.

[12] M. A. Abu Bakar and B. Kijsirikul, "Enhancing Network Visibility and Security with Advanced Port Scanning Techniques," Sensors, vol. 23, no. 17, p. 7541, 2023. doi: 10.3390/s23177541.

[13] J. Clements et al., "Rallying Adversarial Techniques against Deep Learning for Network Security," arXiv preprint arXiv:1903.11688, 2019. [Online]. Available: https://arxiv.org/abs/1903.11688

[14] A. Alhajjar, M. A. Al-Razgan, and M. A. Al-Razgan, "Adversarial Machine Learning in Network Intrusion Detection Systems," Expert Syst. Appl., vol. 185, p. 115526, 2021. doi: 10.1016/j.eswa.2021.115526.

[15] A. Dehghantanha et al., "Artificial Intelligence Driven Cyberattack Detection System Using Ensemble Deep Learning Model," Alexandria Eng. J., vol. 64, pp. 1–10, 2024. doi: 10.1016/j.aej.2024.01.164.

[16] B. Biggio, G. Fumera, and F. Roli, "Security Evaluation of Pattern Classifiers under Attack," IEEE Trans. Knowl. Data Eng., vol. 26, no. 4, pp. 984–996, 2014. doi: 10.1109/TKDE.2013.52.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," arXiv preprint arXiv:1412.6572, 2014. [Online]. Available: https://arxiv.org/abs/1412.6572

[18] N. Papernot et al., "Distillation as a Defense to Adversarial Perturbations Against Deep Neur