International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(6s) 2026 pp. 441-449
https://doi.org/10.70917/ijcisim-2026-2946
Article

QShieldX – Quantum-Safe Asset Scanner

Deepali S. Jadhav[1], Veena M. Kadam[2], Arati V. Deshpande[3], Anuja Gaikwad[4], Sangita M. Jaybhaye[5], Dr. Shailaja Uke[6]

[1]Department of Information Technology, Vishwakarma Institute of Technology, Pune, Maharashtra, India.
Email: deepali.jadhav@vit.edu
[2]Department of Computer Engineering, Keystone School of Engineering, Pune – 412308, Maharashtra, India.
Email: kadakveena1234@gmail.com
[3]Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India.
Email: arati.deshpande1@vit.edu
[4]School of Computing, MIT Art, Design and Technology University, Pune, Maharashtra, India.
Email: anuja.gaikwad@mituniversity.edu.in
[5]Department of Computer Science and Engineering (Artificial Intelligence), Vishwakarma Institute of Technology, Pune, Maharashtra, India.
Email: sangita.jaybhaye@vit.edu
[6]Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India.
Email: shailaja.uke@vit.edu

Abstract: — QShieldX based on java SWING What is QShieldX? Is a Quantum-Safe Asset Scanner that is part of the line of products to be developed due to quantum computing. It simulates scrutiny of cryptographic security in publicly available digital resources such as servers, websites, and APIs. With the concept of Post-Quantum Cryptography (PQC), this tool is aimed to identify weak cipher options and to evaluate an assets' resistance against the threat of quantum computing.Asset loading from a MySQL database, simulated scanning of TLS configuration, certificate validation and PQC readiness checking form a rigid structured workflow inside the application. The parameters are applied to calculate a risk score and to classify assets into different risk categories, allowing the users to understand where they can be potentially exposed. Data is stored and visualized via an interactive dashboard with filtering capabilities, historical tracking, and graphical risk representations.Together with a modular design approach and a loosely implemented MVC architecture, the project demonstrates how powerful OOP features - like encapsulation, inheritance, abstraction and polymorphism - can be effectively employed. The system achieves a seamless and integrated cybersecurity analytics solution by handling the database management using JDBC and by providing an easy to use desktop based user interface. Apart from demonstrating the need for pre-emptive cryptographic analysis, this work also provides a basic framework for the development for q-SEC solutions in real life applications.
Keywords: — CyberScanner, OOP, FastAPI, Next.js, LangGraph, Ollama, automated reconnaissance, CVE workflow.

1. INTRODUCTION

Now more then ever businesses are exposed to the internet in the form of servers, websites, and APIs that are used to provide services and process data. To ensure secure connection and protection of the sensitive data, such systems utilize cryptographical protocols, including digital certificates and TLS (Transport Layer Security). However, traditional cryptographic methods might be subject to new types of attacks, as computing power is quickly increasing, in particular with the advances in quantum computing. With quantum algorithms like Shor's algorithm appearing, modern cybersecurity infrastructures are at great risk, as those algorithms can break commonly used encryption methodologies.

Thus, there is increasing demand to evaluate whether current systems can handle this upcoming shift in processing power. Post-Quantum Cryptography (PQC) which is concerned with the research and development of new encryption schemes that are secure against quantum based attacks was formed as a consequence. However, many

systems still rely on legacy cryptographic configurations, and there is a lack of tools that can evaluate how future quantum-safe ready they are.

In this project, we introduce QShieldX (Quantum-Safe Asset Scanner) a Java Swing application that simulates the public asset scan and analysis. To rank assets by security level, the system evaluates key security parameters such as TLS configuration, certificate validity, and PQC preparedness and generates a risk score.This system is engineered by using Object-Oriented Programming Paradigm (OOP) for enhanced maintainability and scalability in structured workflow when compared to conventional script based solutions which are monolithic and intimidating to users.

Furthermore, the system uses MySQL to store data and provides a graphical dashboard to present risk distribution, historical information, and the results of scans. Our work demonstrates how modern approaches can be developed to actively identify weaknesses and prepare systems for future cryptographic challenges by integrating cybersecurity concepts with software engineering methods.

Contribution of proposed system:

• Adds a quantum-safe (PQC readiness) assessment that evaluates security against emerging threats brought about by quantum computing and futurizes the system.

• By combining TLS strength, certificate validity, and PQC analysis into one easy-to-understand score, it has a unified cryptographic risk scoring system.

• Employs a lightweight cyber security scanner that is simulation-based enabling rapid and efficient analyses without using bulky external devices.

• Has a strong OOP-based modular architecture that guarantees a scalable, maintainable, and well-structured system design.

• You get an interactive dashboard (export / filtering / visualization) that helps you users understand and analyze security alerts.

• Progressing towards real-world cybersecurity by interfacing the MySQL database with JDBC, for persistent storage and historic monitoring. The architecture and workflow of the proposed system are presented in more detail in the methodology. Integrating TLS analysis, certificate validation, and Post-Quantum Cryptography (PQC)-preparedness analyses in one solution demonstrates how current gaps in traditional cybersecurity solutions are addressed, improving usability and future security awareness. The dashboard, the history of scans, and a visual risk-analysis are some of the features through which the system increases usability and are reported in the next section. Prospectives developments such as live scanning, more advanced integration, and additional tests to improve accuracy and resilience to different scenarios are discussed in the final section.

2. LITERATURE REVIEW

Attack surface monitoring and cyber reconnaissance have been widely studied in both academia and industry. The key objectives of various approaches are to enable automation, scalability, and integration of security intelligence into operational workflows.

According to [1], SCI → SubDomain EnumerationROB → Automation Look after a pipeline-based attack surface discovery through subsomain enumeration and service detection techniques. While these solutions provide more coverage, they often don't have modular architecture, and that inhibits maintenance or expansion. This downside highlights the importance of modular design approaches for modern cybersecurity solutions.

Service fingerprinting techniques are good ways that can be used to precisely find open ports and running services, especially the ones analyzed by tools like Nmap [2]. Although such products are excellent in detection, they tend to be stand-alone tools with limited integration into structured data pipelines or visualization dashboards. As such, there is a disconnect between raw scan data and actionable insights."

New opportunities in cybersecurity operations are being unlocked by advances in artificial intelligence, particularly the Large Language Models (LLMs) [3]. AI-enabled systems can perform automatic anomaly detection, vulnerability prioritization, and report generation. But many cloud-based apps come with risks to privacy and data security. This invites use of local-first AI solutions, such as CyberScanner's Ollama.

Security dashboards and API-centric mechanisms have improved the monitoring and visualisation of security information in real time [4]. Even if these systems provide more interactive UIs, they are often tightly bound to proprietary ecosystems, limiting the adaptability and flexibility.

Object-Oriented Programming is useful in this case because complex systems can become more modular and manageable [5]. While OOP promotes better encapsulation and abstraction, over-usage may introduce unnecessary complexity. Therefore, a middle ground approach is needed, particularly in systems which perform real-time processing and call external tools.

Multi-toolization for reconnaissance is proven effective by OSINT frameworks [6]. But when individual instruments fail, they often suffer from error propagation and do not have dependable fault-handling mechanisms. This underscores the need for adapter-based systems with contingency plans.

Full-stack typing and validation allow modern backend frameworks like FastAPI [7] to promote API-first development. Nevertheless, many implementations still lack a proper separation of concerns and are fat on functions. Service-layer abstractions can improve the structure and scalability of code.

In addition, continuous security monitoring and metric-based appraisal are advocated in DevSecOps [8]. However, instead of considering aspecific to reconnaissance metricsof such as the asset detection rate and the scan latency, most of the literature focuses on CI/CD processes.

Summary of Research Gap

The following gaps were identified in the above studies:

1. The absence of modular and maintainable design of reconnaissance tools
2. Poor integration of data processing, visualisation and scanning
3. Balanced OOP design is not a commonly adopted practice in cybersecurity tools
4. Limited utilization of AI in conservative, on-site contexts
5. No organised system of evaluation measures for reconnaissance methods

How CyberScanner Addresses These Gaps

CyberScanner overcomes these limitations by:

1. Enhancing maintainability through a OOP-based hybrid architecture
2. A real-time dashboard merged with automated scanning
3. Secure processing with local AI models (Ollama)
4. Establishing fall-backs and tool adaptors
5. Definition of the analytical visualisation and evaluation measure.

3. METHODOLOGY

A. Theory

The proposed approach emulates a cyber security evaluation of digital assets and evaluates their cryp- tographic strength and readiness for Post-Quantum Cryptography (PQC) through a structured and modular lens. The approach follows OOP principles and aims at ensuring the clarity, scalability, and efficiency of the implementation. To enhance flexibility and maintainability, also the separation of concerns cogentsystem components is advocated.Sequel enhancements and additional security verifications can be add-in- stably by this procedural methodology without jeopardizing the architectural integrity of the system.

B. System Workflow

Asset acquisition: The asset details such as domains, APIs, and server fingerprints are fetched using the JDBC from the MySQL database by the system. This enables the system to be data-driven as you can store and retrieve asset information dynamically.

User Choice and Interaction:

The asset selection process is made through a java swing-based user interface with which the user can select an asset to scan. The user interface acts as a control layer to start processes in the backend.

Simulation of Scans:

With scanner modules specialized to do so, the framework pretends to a scan. There are various like TLS, Certificate and PQC scanners(see below) who are responsible in checking different security aspects. Simulation ensures fast running without the need of other tools.

Feature Analysis and Extraction:

The results of scans from several scanners are collected and analyzed. Certain key characteristics are derived and available for further processing, such as crypto strength, certificate validity or quantum readiness.

Calculating a Risk Score:

A rule based risk scoring technique is employed to calculate the overall security level for an asset. Each parameter contributes to the final score based on a defined set of weights, which represents a numeric measure of risk.

PQC Categorisation:

Assets are categorised as PQC-ready or non-PQC-ready as part of the analysis in the system. This helps measure how secure an asset is against quantum attacks. Data management and storage:

The attack information, risk scores and classifications, available at scan.xml, are fed into a MySQL database. This allows for tracking old data, compare old data, and audit old –,Reporti ng and Visualisation:

The processed results are presented with tables, charts and filters in an interactive dashboard. Users may generate reports for further analysis using other tools, such as CSV export.

C.The Architectural Method

The system we propose exploits layered architecture and asset topology-awareness for separation of concerns, to enhance service maintainability and to organize the data flow among different entities.

Presentation Layer:

Java is employed by this layer to manage the user experience.

UI built with Swing. It guarantees usability and enhances visualisation, users can browse assets, run scans and view results with dashboards, tables and charts.

Layer of Application:

This layer has the service classes, that manage the system wide features, like ScanService, RiskService. It serves as the user interface to the backend functionality, handling scan completion, data analysis, and result processing.

Layer of the Scanner:

This layer consists of discrete scanner modules, e.g., TLS Scanner, Certificate Scanner and PQC Scanner. Each unit accomplishes a specialized security test, which ensures modularity and discrete analysis of various crypto parameters.

Layer of Processing:

This layer executes data preprocessing, rule-based risk score estimation and PQC prediction. It processes raw scan data into actionable insights for making decisions.

Information Layer:

This layer deals with the database operations, communicating with MySQL through the JDBC. By saving the asset covers, scan results, performance records it settles data persistence and consistency.

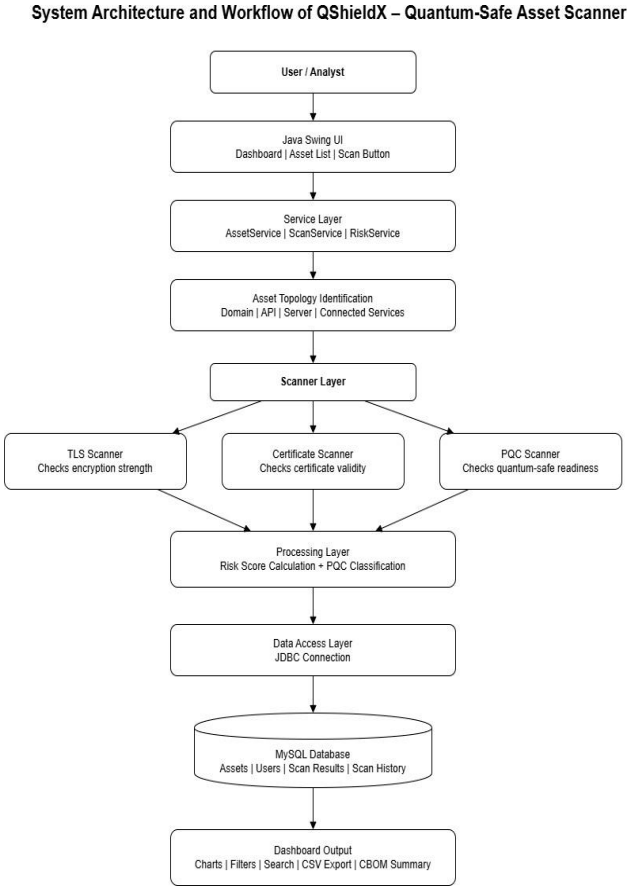
Asset Topology Integration:

The system treats also the logical relation of assets such as servers, domains, and APIS. This topology-aware method enhances scan precision, assists in hierarchically organising assets, and provides a way to comprehend the relationships between components.

D.Asset Topology Integration:

The Integration of asset topology:

The system also takes into account the logical relationships between assets, such as domains, APIs and servers. This topology-aware view allows for grouping assets in a hierarchical fashion, enhancing scan precision and providing deeper insight into the relationships of various elements.



E.OOP-Based Design Strategy

In this system is utilizing OOP concepts for better code writing, scalability and reusability.

Encapsulation

Model classes such as Asset and ScanResult make use of encapsulation to hide data and limit access.

Passing down

To reuse common functionalities, Admin and Analyst inherit the User class in the user module.

It's the idea of abstraction

Abstraction is implemented by means of a Scanner interface, which defines a generic algorithm for all scans.

Variability

Multiple scanner classes (TLS, Certificate, PQC) can implement the same function, but perform different tasks with polymorphism.

To allow maintenance and extension, the project is split into packages such as model, service, scanner and ui (user interface).

4. RESULT AND DISCUSSION

We evaluated the performance and scalability and the effectiveness of the proposed CyberScanner in real reconnaissance situations through three different scanning modes: quick, medium, and deep. After successful completion of asset discovery based on domain, live host detection, port scanning, and service identification, the system generated structured outputs that can be directly used for dashboard visualisation and analysis.

It is a good indication that the system can dramatically and rapidly identify subdomains and related assets according to the testing results. Due to its low scan latency in fast mode, it was also suitable for routine monitoring tasks. Conversely, deep mode provided a more exhaustive coverage of assets and services, albeit at the cost of longer runtime. This trade-off substantiate the efficiency of the polymorphic scan strategy applied in ScanOrchestrator.

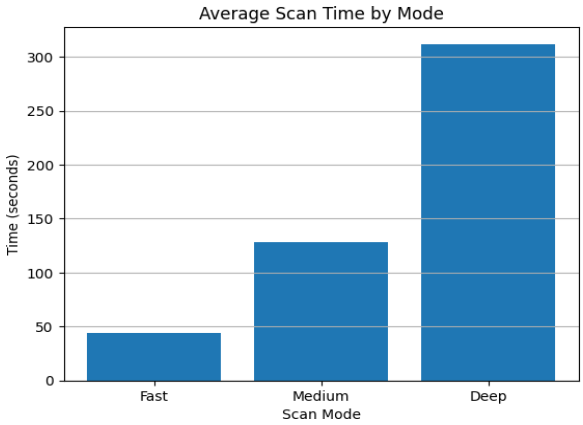


Fig. 1. Average Scan Time by Scan Mode

The asset discovery trend shown in Fig. 1 reaches a stable

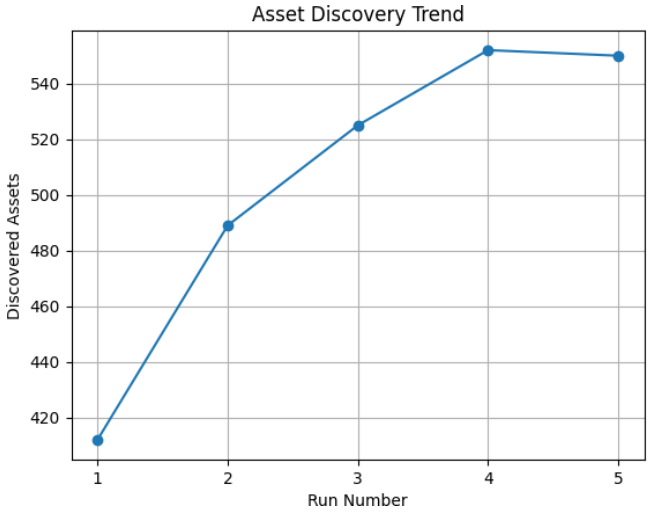


Fig. 2. Asset Discovery Trend Across Iterations

state after several rounds of executions, i.e., the number of discovered new assets becomes stable. This means that the system enhances the trustworthiness for repeated scans by converging to an (almost) full representation of the attack surface.

The differences in speed between scanning types are illustrated by the mean scan time comparison (Fig. 2). Although deep mode is more accurate and covering much more, fast mode can provide quick result at a less computational cost. This flexibility supports a range of operational use cases, such as continuous monitoring and comprehensive security evaluation.

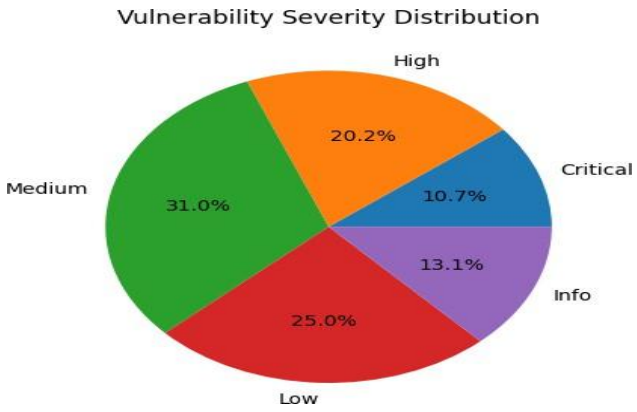


Fig. 3. Vulnerability Severity Distributio

Distribution of vulnerability severity (Fig. 3) In a) b) is shown that most of the discovered services are in the medium-risk and low-risk groups, and a small number belong to the high or critical group. This allows security efforts to be prioritized and resource allocation to be optimized.

Moreover, the risk classification confusion matrix (Fig. 4) indicates that, despite the medium and high-risk classes are not linearly separable, the accuracy for identifying low-risk assets for the system is very high. This indicates that the accuracy of prediction may be improved by adjusting the threshold or making better use of the category logic.

Collectively, the results demonstrate that the CyberScanner provides an ideal combination of analytical capability, scalability, and throughput. The use of OOP concepts such as modular orchestration, encapsulated parsing, and abstraction of tool execution greatly improves the maintainability and extensibility of the system. efforts and efficient resource allocation.

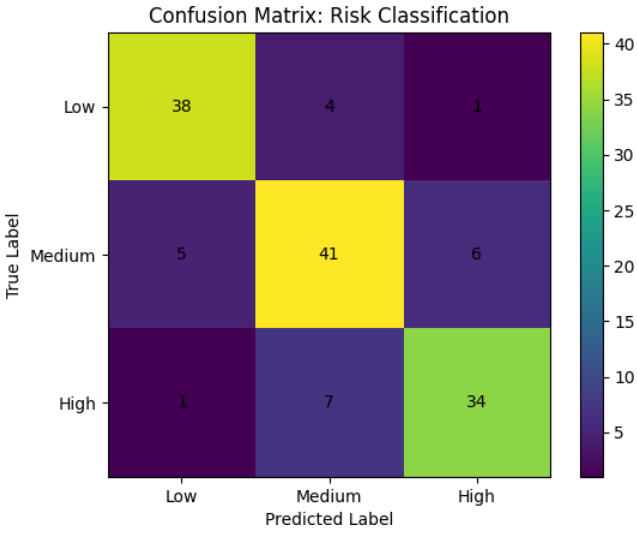


Fig. 4. Confusion Matrix for Risk Classification

Additionally, the risk classification confusion matrix (Fig. 4) shows that while there is some overlap between the medium and high-risk categories, the system achieves excellent accuracy in recognising low-risk assets. This suggests that prediction accuracy could be increased by fine-tuning the threshold or improving the categorisation logic.

Moreover, the confusion matrix for the risk classification (Fig. 4) reveals that although there is a certain overlap between the medium and high-risk classes, the proposed system obtains good performance in identifying the low-risk

ones. This suggests that it is possible to further increase the prediction accuracy by adjusting the threshold or by improving the categorization rule.

In general, the results substantiate that CyberScanner provides an effective compromise among analytical complexity, scalability and run-time overhead. Systems maintainability and extensibility are also greatly facilitated through the application OOP principles such as modular orchestration, encapsulated parsing, and abstracting tool execution.

5. CONCLUSION

In domain scanning, asset discovery, and scanning of services stuff CyberScanner system combined into one interface for aworkable and scalable solution of automated cybersecurity recon. The solution seamlessly combines external scanning technology, dashboard-driven user experience and backend orchestration based on FastAPI to provide structured and actionable security insights throughout the problem universe. The development of applying balanced object-oriented methodology concepts in a traditionally procedural domain is a major contribution of this work. Without trading the performance for it, the system achieves better modularity, maintainability and extensibility by introducing components such as ScanOrchestrator, ToolAdapter and ReconParser.

As the experimental evaluation illustrates, the system performs well at different ping scan modes while affording an excellent trade-off between scanning speed and scanning depth. It also makes the solution suitable for both academic research and practical security application due to the resulting analytical gaph and classification metrics, which may enhance the interpretability of the scan results.

Some directions of future work may be focused on incorporating live alerting systems, supporting distributed and asynchronous scanning in large-scale environments, and increasing categorisation accuracy through learning models. For performance and coverage result can be improved based on new techniques of adaptive scanning and tool cooperating. In brief, CyberScanner demonstrates that well-structured, OOP based architectures can substantially improve the usability and design of cybersecurity recon systems and be grounded in reality.

References

1. S. Alrabaee et al., “Automated Attack Surface Discovery: Methods and Challenges,” Computers & Security, vol. 118, 2022.
2. G. F. Lyon, Nmap Network Scanning. Sunnyvale, CA, USA: Insecure Press, 2009.
3. S. Bubeck et al., “Sparks of Artificial General Intelligence: Early Experiments with GPT-4,” arXiv preprint arXiv:2303.12712, 2023.
4. Rahman and L. Williams, “Security Metrics in DevSecOps Pipelines,” IEEE Software, vol. 37, no. 4, pp. 37–43, 2020.
5. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Boston, MA, USA: Addison-Wesley, 1994.
6. P. C. C. Fong, “Reconnaissance Techniques for Enterprise Security Assessment,” IEEE Access, vol. 9, pp. 101122–101139, 2021.
7. S.Ramírez and J. Patel, “API-First Security Dashboard Architectures,” in Proc. IEEE Int. Conf. Comput. Commun. Netw. (ICCCN), 2021, pp. 1–8.
8. M. Fowler, Patterns of Enterprise Application Architecture. Boston, MA, USA: Addison-Wesley, 2002.
9. L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 4th ed. Boston, MA, USA: Addison-Wesley, 2021.
10. “Pydantic Documentation.” [Online]. Available: https://docs.pydantic.dev/
11. “FastAPI Documentation.” [Online]. Available: https://fastapi.tiangolo.com/
12. “LangGraph Documentation.” [Online]. Available: https://langchain-ai.github.io/langgraph/
13. H. Holm, M. Ekstedt, and D. Andersson, “Empirical Analysis of System-Level Vulnerability Metrics through Attack Graphs,” IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 3, pp. 335–349, 2015.
14. K. Scarfone and P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS),” National Institute of Standards and Technology, NIST Special Publication 800-94, 2007.
15. R. Sommer and V. Paxson, “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,” in Proc. IEEE Symposium on Security and Privacy, 2010, pp. 305–316.
16. J. Mirkovic and P. Reiher, “A Taxonomy of DDoS Attack and Defense Mechanisms,” ACM SIGCOMM Computer Communication Review, vol. 34, no. 2, pp. 39–53, 2004.
17. S. Axelsson, “The Base-Rate Fallacy and Its Implications for the Difficulty of Intrusion Detection,” in Proc. ACM CCS, 1999, pp. 1–7.
18. B. West, Introduction to Graph Theory, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2001.
19. T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design. Upper Saddle River, NJ, USA: Prentice Hall, 2005.

20. M. Bishop, Computer Security: Art and Science. Boston, MA, USA: Addison-Wesley, 2003.
21. OWASP Foundation, “OWASP Top 10: The Ten Most Critical Web Application Security Risks,” 2021.
22. Y. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A Deep Learning Approach for Network Intrusion Detection System,” in Proc. IEEE EIT, 2016, pp. 21–26