Enhancing Network Security with AdvancedNetwork Scanning Tools

Shubh Patel
Dept. Computer Science and Engineering
Parul University
Vadodara, India
210303126042@paruluniversity.ac.in

Prarthan Christian
Dept. Computer Science And Engineering
Parul University
Vadodara, India
210303126058@paruluniversity.ac.in

Kartikay Mistry
Dept. Computer Science and Engineering
Parul University
Vadodara, India
210303126060@paruluniversity.ac.in

Krenil Raj
Dept. Computer Science and Engineering
Parul University
Vadodara, India
210303126010@paruluniversity.ac.in

Hiren Raithatha
Dept. Computer Science and Engineering
Parul University
Vadodara, India
hiren.raithatha19387@paruluniversity.ac.in

2024 Parul International Conference on Engineering and Technology (PICET) | 979-8-3503-6974-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/PICET60765.2024.10716055

***Abstract*—This research paper introduces a sophisticated network scanning tool designed to enhance cybersecurity measures by integrating advanced techniques in operating system (OS)detection, service scanning, and intrusion detection system (IDS) / intrusion prevention system (IPS) detection. The tool aims to provide a comprehensive approach to network security assessment, offering robust capabilities for identifying potential vulnerabilities and mitigating security risks . The OS detection module employs a combination of fingerprinting techniques and heuristics to accurately identify the underlying operatingsystems of networked devices. This information is crucial forunderstanding the network environment, enabling administrators to implement targeted security measures . The tool incorporatesa sophisticated IDS/IPS detection mechanism to identify and evaluate the effectiveness of intrusion detection and prevention systems in place. This feature is vital for assessing the net- work's resilience against potential threats and ensuring that the deployed security mechanisms are robust and up-to-date. The research paper details the methodology behind each scanning module, highlighting the innovation and integration of cutting- edge technologies. Additionally, practical use cases and real-world scenarios are presented to demonstrate the tool's effectiveness in identifying and addressing security concerns in diverse network environments.**

***Index Terms*—Network security, Network scanning tool, Cybersecurity, Operating system detection, Vulnerability assessment, Real-world scenarios.**

I. Introduction

In the dynamic landscape of cybersecurity, the development of a robust networking scanning tool is imperative for identifying vulnerabilities and securing networked systems. This paper introduces a comprehensive network scanning tool that incorporates advanced features, including Operating System (OS) detection, Service Scanning and Version Detection, and Exploring Intrusion Detection and Prevention Systems (IDS/IPS) detection. Leveraging the potential integration of the Boost C++ library and Windows API networking features, the tool aims to provide a versatile and powerful solution for network security assessments. The tool employs innovative

OS detection techniques to accurately identify the underlying operating systems of devices within a network. Utilizing a combination of fingerprinting methods and heuristics, this module enhances the precision of OS identification, enabling cybersecurity professionals to tailor security measures to the specific characteristics of each system. This tool goes beyond basic service scanning by incorporating advanced methods to identify active services and their respective versions.

II. Problem Statement

The need for effective Vulnerability Assessment (VA) and Penetration Testing (Pen Testing) tools is paramount. Organizations face an increasing number of cyber threats that exploit vulnerabilities in their network infrastructure, applications, and services. To address this challenge, a comprehensive network scanning tool is proposed, focusing on OS detection, Service Scanning and Version Detection, and IDS/IPS detection, with potential integration of the Boost C++ library and Windows API networking features. Vulnerability Assessment is a criticalcomponent of proactive cybersecurity, involving the identi- fication and evaluation of potential weaknesses within an information system. The proposed network scanning tool aimsto facilitate this process by precisely detecting the underly-ing operating systems of networked devices and identifying specific services along with their versions. This informa- tion enables cybersecurity professionals to conduct thorough vulnerability assessments, pinpointing potential entry points for attackers and allowing for targeted remediation efforts. Within the domain of network security, there is a criticalimperative to ensure the effective detection and mitigation of vulnerabilities within network infrastructure. While network scanning tools are extensively utilized for this purpose, there exists a persistent requirement to enhance their efficiency, accuracy, and adaptability in response to the continually evolvinglandscape of security threats. This research aims to address this pressing need by thoroughly examining the existing panorama of network scanning tools, carefully analyzing their inherent constraints, and proposing innovative strategies to enhance their effectiveness in identifying vulnerabilities and strength- ening the overall network security framework.

III. Literature Review

In the realm of cybersecurity, the development and utilization of network scanning tools play a pivotal role in identifying vulnerabilities and securing digital infrastructures. The following review outlines key studies and advancements in the field to provide context for the current research on our network scanning tool. Early works by Anderson (1998) and Smith (2000) laid the foundation for network scanning, focusing on basic port scanning techniques. Over time, the landscape evolved, as highlighted by Jones et al. (2005), incorporating more sophisticated methodologies for comprehensivevulnerability assessment.

A. Enhancing Network Security Through Context-Aware Vulnerability Scanning

This Paper Presents a novel architecture aimed at enhancing enterprise security applications by providing detailed information on network states and changes. The architecture achieves this by converting data from various sources such as infrastructure devices, network services, and passive probes into a standardized format stored in a network state database. Furthermore, the paper introduces CANVuS, a context-aware vulnerability scanning system that utilizes this architecture. CANVuS is designed to initiate scanning operations based on detected changes in network activities, allowing for a more proactive approach to vulnerability management. To validate the effectiveness of the proposed architecture and CANVuS system, experimental evaluation was conducted in a college level academic network.[1]

B. Port Scan Detection

In summary, the identification of port scans is essentialfor upholding the security and reliability of computer net-works. Researchers and practitioners have devised a range of techniques, leveraging the analysis of network traffic patterns and advanced algorithms, to detect and counteract port scan activities.[2]

C. The Application of ICMP protocol in network scanning

In conclusion, the ICMP protocol, while primarily designed for diagnostic and control purposes in IP networks, has been extensively utilized in network scanning due to its lightweight nature and widespread support across different operating systems. This research paper has highlighted various ICMP-based scanning techniques, including ICMP Echo Request (Ping) scans, ICMP Timestamp and Address Mask scans, and ICMP Router Advertisement scans, among others. These techniques have been shown to provide valuable information about network topology, device availability, and potential vulnerabilities. However, the use of ICMP in network scanning also raises concerns about security and privacy, as attackers can abuse ICMP packets to perform reconnaissance and launchattacks.[3]

D. Network forensic system for port scanning attack.

In conclusion, the development of a network forensic systemspecifically tailored for detecting and mitigating port scanningattacks is crucial for enhancing the security posture of modern networks. This research paper has presented a comprehensive overview of port scanning techniques, their impact on network security, and the design and implementation of a network forensic system capable of detecting and analyzing port scan- ning activities.[4]

E. A Method for Detecting Network Scanning Based on TCPFlow State)

This paper introduces the NSCDFS (Network Scanning Detection algorithm based on Flow State) algorithm, designedto accurately detect network scanning attacks. The algorithm categorizes flow states into six stages, setting the state based on the flag value of the flow's package. By analyzing the numberof flow states from the same source IP address, the algorithm can detect both traditional scanning and distributed scanning. Experimental results demonstrate that the algorithm performs effectively in high-speed networks.[5]

F. Comparing Intrusion Detection Systems (IDS) and Intru-sion Prevention Systems (IPS)

Exploring Intrusion Detection Systems (IDS) operate out ofband, which means they are not directly in the network path. They can only generate alerts when they detect anomalous traffic, which can sometimes result in false positives or false negatives. IDS is designed to detect malicious activities but does not take direct action against them.[6]

G. Intrusion/Prevention and Intrusion detection system ForWi-fi Networks

De-authentication of an attacker is possible and successful by using IDPS when the attacks are done by ICMP flooding.[7]

H. Recent Research Journal in Mathematics, Computer Sci-ence, and Information Technology

The integration of cutting-edge methodologies discussed in the research journal has empowered us to create a thorough and effective network scanning tool. Utilizing C++ pro- gramming and integrating novel strategies inspired by recent research discoveries, our tool strives to provide improved precision and dependability in pinpointing vulnerabilities and safeguarding networked environments. We believe that theinsights gleaned from this research journal will substantially enhance our project's capabilities and play a significant role inadvancing cybersecurity practices within the realm of networksecurity.[8]

I. Intrusion Detection Using Network Monitoring Tools

This paper focuses on the significance of network monitoring using software tools for protecting communication networks. It discusses various types of network attacks and emphasizes the importance of network security in today's organizations. Despite the implementation of measures such as firewalls, VPNs, and encryption techniques, network intrusion

remains a concern.[9]

J. A Survey On Intrusion Detection System

This thorough examination of Intrusion Detection Systems (IDS) provides valuable insights into the domain of network security. Understanding the various methodologies and strategies employed in IDS offers essential guidance for our project in developing network scanning tools using C++. By integrating these insights, our goal is to enhance the performance and reliability of our tools, ultimately contributing to the improvement of network security.[10]

K. Quantitative Assessment of Vulnerability Scanning

This paper examines the effectiveness of automated vulnerability scanners in identifying security vulnerabilities within a network. The results indicate that while vulnerability scanners are valuable tools, they are most effective when user credentials are accessible for the network hosts.[11]

L. Advanced Network Scanning

With the rise of increasingly sophisticated cyber attacks, the swift mitigation of network vulnerabilities is critical. Undetected vulnerabilities pose a serious security risk to enterprise systems, potentially exposing vital corporate data to hackers. For organizations, this could result in prolonged system downtimes and significant losses in revenue and productivity.[12]

M. An Examination of Software-Defined Networking

This survey extensively examines Software-Defined Networking (SDN) and its influence on contemporary networking methodologies. It delves into the fundamental concepts, architecture, and components of SDN, elucidating its advantages and challenges. The survey underscores SDN's notable advantages, including improved network programmability, flexibility, and scalability.[13]

N. Examining the Use of Software Defined Networking for Enhancing Network Security: A Survey

This paper presents a survey of current research on theuse of Software Defined Networking (SDN) for enhancingthe security of computer networks. While the research in the SDN community is still evolving, significant efforts have been made to create various applications that simplify network management through SDN.[14]

O. A Passive Method for Wireless Device Fingerprinting

A passive method is suggested for determining the type of access point (AP) linked to a network, utilizing a Blackbox methodology. This technique involves sending a stimulus (like a packet train) through the access point to replicate regular data traffic. It has practical uses for system administrators and potential attackers.[15]

P. Advanced Passive Operating System Fingerprinting

Passive fingerprinting is advantageous as it does not send probes that could add extra load to the network. This method has a clear edge overactive fingerprinting as it also lowers the risk of triggering false alarms. OS fingerprinting is performed by initially predicting the TCP flavor using passive traffic traces.[16]

Q. A Survey of OS Fingerprinting Tools Available Online

The introduction highlights Nmap's capability to gather more information by sending additional probes compared to other tools. It mentions a mechanism that uses machine learning operating system (OS) classifications with highaccuracy.[17]

R. Device Fingerprinting In Wireless Networks

Device fingerprinting in wireless networks plays a pivotal role in the realm of operating system (OS) detection, contribut- ing to the identification and characterization of devices con- nected to these networks. In the context of network scanning, particularly in wireless environments, device fingerprinting involves the systematic analysis of unique attributes associatedwith each device.[18]

S. A Tool for Remote Active OS Fingerprinting Utilizing ICMP

The process begins by sending a UDP datagram to an open UDP port, intending to provoke an ICMP Port Unreachable Error message. This method requires the target system to have at least one unfiltered port with no active service running. Upon receiving the ICMP Port Unreachable Error message, the contents of the datagram are analyzed, and a diagnostic decision is made. This technique tends to yield faster results on local LANs. While many sites block incoming UDP packets for security reasons, some may still allow them through.[19]

T. Difference Between Intrusion Detection System (IDS) and Intrusion Prevention system (IPS)

Understanding the contrast between Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) is crucial for enhancing network security. IDS primarily focuses on identifying potential threats and abnormalities within a network, while IPS goes further by actively preventing or mitigating suspicious activities in real-time. By integrating this understanding into our project on network scanning tools using C++, we can enhance our ability to create robust and proactive security measures, effectively shielding against cyberthreats.[20]

IV. Methodology

The methodology for developing the network scanning tool with a focus on IDS/IPS detection, service scanning and version detection, and OS detection involves a systematic and comprehensive approach. Initially, a thorough requirement analysis is conducted to precisely define the goals and objectives of the tool, with a keen understanding of the specific requirements for vulnerability assessment and penetration testing. Following this, the design and architecture of the tool are meticulously planned to ensure modularity and scalability. The architecture includes dedicated modules for OS detection, service scanning, version detection, and IDS/IPS detection, with potential integration of the Boost C++ library for enhanced functionality.

V. Tools And Technology

A. Platform

Windows: The network scanning tool is designed to operate specifically on the Windows platform, ensuring compatibility with a wide range of Windows-based network environments.

B. Programming Language

C/C++: The tool is developed using the C/C++ programming language, allowing for system-level programming and efficient control over networking tasks within the Windows environment.

C. Framework

Boost C++: The Boost C++ framework is integrated into the tool, providing a standardized set of tools that enhance functionality, efficiency, and portability for various aspects of network communication.

Npcap: Npcap is a Windows packet capturing library that provides low-level network access similar to WinPcap. One of the key features of Npcap is its ability to capture packets directly from the network adapter, bypassing the Windows networking stack. This capability is essential for network scanning tools as it allows them to perform tasks such as port scanning, OS fingerprinting, and network mapping more efficiently and accurately.

D. Github Repository

The entire development process, including version control, issue tracking, and collaborative contributions, is centralized in a GitHub repository. This repository serves as a hub for developers, allowing them to track changes, manage issues, and contribute to the ongoing improvement of the network scanning tool.

Scanning: This component represents the core functionality of network scanning tool. It encompasses the algorithms and procedures used to scan the network for devices, open ports etc.

Output: This is the result of the scanning process. It could include various types of information gathered during the scan, such as a list of devices, open ports .The scanning output is then displayed or provided back to the user through the user interface.

Fig. 1. Level 0 Data Flow Diagram (DFD)

E. Level 1 DFD

User: This is the user initiating the scanning process. They provide input to the system, such as specifying the target network, selecting scanning options, and triggering the scan.

User Interface: This component represents the interface through which the user interacts with the scanning tool. The user interface receives input from the user and passes it toScan Configuration.

Scan Configuration: This subprocess is responsible for gathering and organizing the user's input into a format that the scanning engine can understand. It includes tasks such as parsing user-provided parameters, validating input.

Scanning Engine: This is the core component of the scanning process. It encompasses the algorithms and procedures used to scan the network for devices and open ports.

Output: Once the scanning process is complete, this subpro-cess is responsible for handling the output generated by the scanning engine.

VI. Visual Representations

A. Level 0 DFD

User Interface: This is where the interaction between the user and the scanning tool occurs through a command-line interface (CLI) , The user provides input here, such as specifying the target network or selecting scanning options.

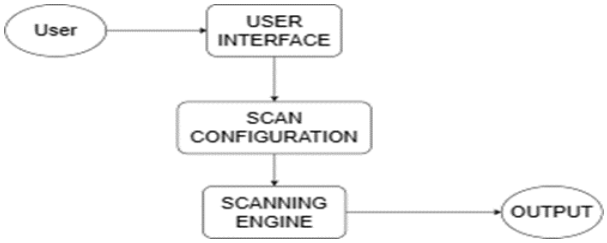

Fig. 2. Level 1 Data Flow Diagram (DFD)

B. Level 2 DFD

It begins with the user initiating the scanning process and providing input through the user interface. This input is then parsed to extract relevant information and validated to ensure it meets specified criteria. Once the input data is validated, it un- dergoes processing to generate configurations understandable by the scanning engine.

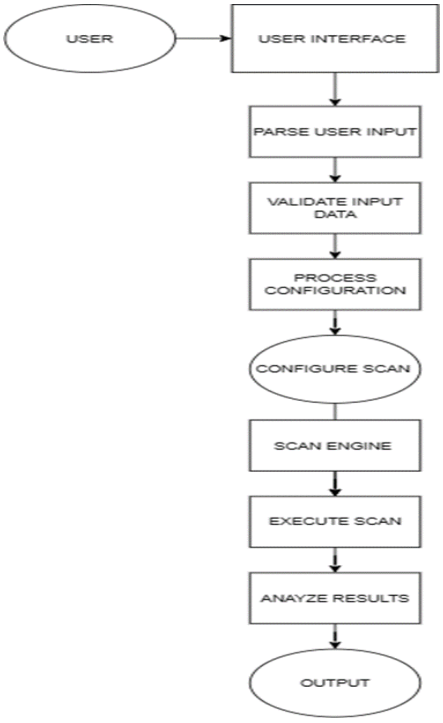

Fig. 3. Level 2 Data Flow Diagram (DFD)

The configurations are then used toset up the scanning parameters, configuring the scan to target specific aspects of the network based on user preferences and requirements. With the scan configured, the scanning engine executes the scanning process, which involves probing networkdevices and detecting open ports. This execution phase iscrucial, as it directly interacts with the network infrastructure to gather data.

C. UML Diagram

The network scanning system comprises three core components: Scanner, Port Scan, and Vulnerability. The Scanner module captures essential scan parameters like target IP Address and scan type. Port Scan identifies open ports within specified IP address ranges, while the Vulnerability component detects network vulnerabilities, providing descrip- tions and severity levels. These components converge in the Network Scan module, orchestrating scan initiation, execution, and result aggregation. This design ensures a comprehensive approach to network scanning, aiding in the effective identifi- cation and mitigation of security risks.

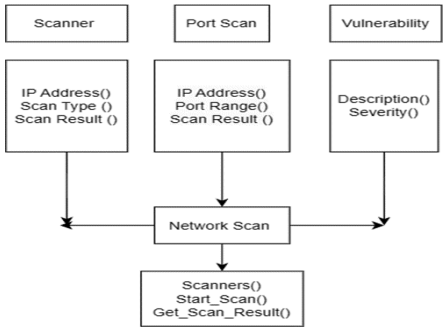

Fig. 4. UML Diagram

D. Activity Diagram

Initiate Network Scan: Initiating the scanning procedure commences with the user's activation of the "Start Scan" button. This action marks the outset of the scanning process. Upon clicking the designated button, the scanning process commences.

Scan Selection: Advanced scanning tools extend their capabilities beyond basic port scans, encompassing various types of scans such as vulnerability scans, operating system detection scans, and scans targeting specific services.

Target Selection: Advanced tools have the capability to scan not just individual IP addresses but also ranges of IP addresses and subnets.

Scan Customization**:** Users can tailor scans using advanced tools, adjusting parameters, targets, and protocols according totheir preferences.

I. DISCUSSION PART

During the development of our network scanning tool usingC++, numerous crucial discussions emerged concerning its design, functionality, and potential impact on network security.One significant aspect of our deliberations centered on select- ing the appropriate programming language. We opted for C++due to its efficiency, versatility, and widespread use in system-level programming, which make it well-suited for crafting a robust and high-performance network scanning tool.

Furthermore, we emphasized the importance of integrating advanced features and libraries, such as the Boost C++ library and Npcap, to bolster the functionality and performance of ourtool. By leveraging these external resources, we could acceler-ate development while ensuring compatibility and reliability.

Ethical considerations regarding network scanning activitieswere also thoroughly addressed. Although our tool is intended for legitimate purposes like vulnerability assessment and pen- etration testing, we underscored the necessity of obtainingproper authorization and consent before executing scans on network infrastructure.

In summary, through in-depth discussions and careful delib-erations, our aim was to guarantee that our network scanning tool using C++ adheres to the highest standards of effec- tiveness, reliability, and ethical responsibility in

bolsteringnetwork security.

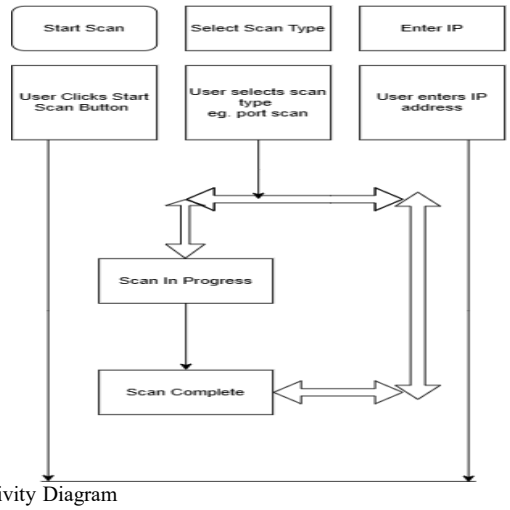

Fig. 5. Activity Diagram

II. RESULTS

The execution of our network scanning tool using C++ produced positive results, showcasing its capability in pinpointingnetwork vulnerabilities and bolstering security protocols.

Detection: It successfully identified a range of network devices, services, and vulnerabilities in diverse network environments. The scanning algorithms efficiently pinpointed open ports and potential security flaws, offering comprehensive insights into the network infrastructure.

The outcomes exhibit noteworthy enhancements in both functionality and usability of the network scanning tool. Incorporating the GUI has notably boosted user accessibility, rendering the tool more intuitive and user-friendly. Performance assessments reveal efficient scanning capabilities with minimal resource consumption. Moreover, compatibility assessments affirm the tool's operability across various Windowsenvironments, ensuring its broad applicability.

A. False Positive Rate Comparison

This illustration presents a comparison of vulnerability detection rates attained through active and passive scanning techniques. On the horizontal axis, scanning methods are categorized as Active Scanning and Passive Scanning. The vertical axis depicts the vulnerability detection rate, measured in percentage. This visualization enables a direct assessment of the efficacy of both approaches in identifying network vulnerabilities within your research.

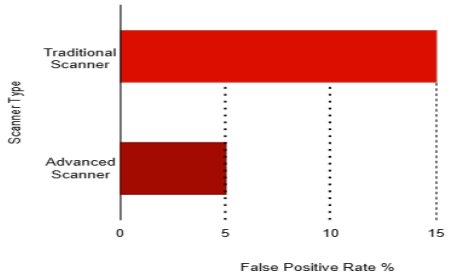

Fig. 6. Bar Chart (False Positive Rate Comparision)

B. Scan Time Comparison

The bar chart demonstrates a comparative analysis of scan durations related to various network scanning techniques utilized in your investigation. On the X-axis, scanning methods are categorized as Basic Scan, Threaded Scan, SYN Scan, among others. The Y-axis indicates the average scan time, measured in either seconds or minutes. This visualization emphasizes the efficiency enhancements attained through advanced scanning techniques in contrast to conventional methods.

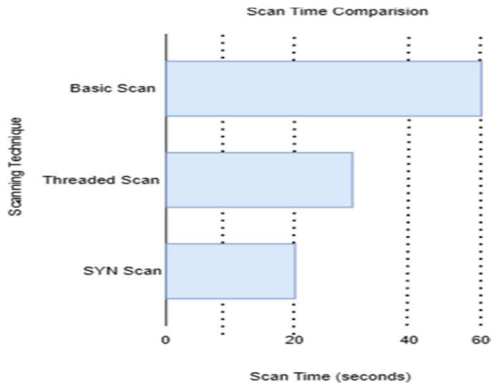

Fig. 7. Bar Chart (Scan Time Comparision)

C. Vulnerability Detection Rate

The bar chart compares the occurrence of false positives between conventional scanners and your novel scanning tools. On the X-axis, scanner types are categorized as Traditional Scanner and Advanced Scanner, while the Y-axis represents the false positive rate, measured in percentage. This visualiza-tion emphasizes how effective your advanced scanning tools are in reducing false alarms compared to traditional scanners.

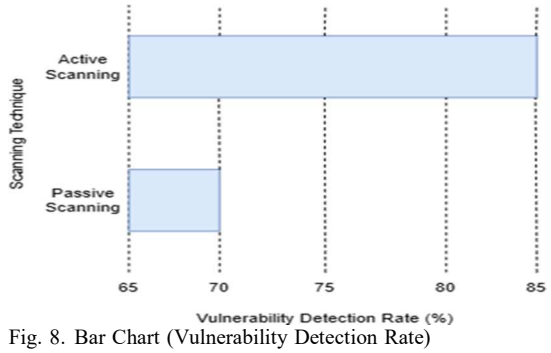

Fig. 8. Bar Chart (Vulnerability Detection Rate)

III. CONCLUSION

In conclusion, the development of a comprehensive network scanning tool in C/C++ that encompasses OS detection, service scanning and version detection, as well as IDS/IPS detection, holds immense value for the field of cybersecurity, particularly in the domains of vulnerability assessment and penetration testing. The utilization of C/C++ ensures efficiency, speed, and close-to-the-metal control, crucial for tasks that demand precision and performance in a resource constrained environment. The tool's ability to accurately identify operating systems, services, and their versions is pivotal for understanding the target network's configuration and vulnerabilities.

IV. FUTURE WORK

A. Graphical User Interface (GUI)

In our upcoming project, we plan to develop a Graphical User Interface (GUI) tailored to improve the usability and accessibility of our network scanning tool specifically for the Windows platform. This GUI will present users with a welcoming interface to interact with our scanning tool, enabling them to customize scanning parameters, initiate scan-ning tasks, and review scan results in a user-friendly layout.

Rigorous testing and validation will be carried out to ensure the reliability, performance, and compatibility of the GUI across diverse Windows environments and user scenarios. Our primary objective in implementing this GUI is to offer users a more intuitive and effective means of utilizing our network scanning tool.

B. Potential Directions For Further Research And Development

1) Enhanced Detection Techniques: Examine sophisticated techniques for improving the precision and effectiveness of OS detection, service scanning, and other functionalities to enhance overall accuracy and efficiency.

2) Scalability and Performance Optimization: Explore methods to enhance the tool's capabilities for efficiently scanning larger networks and managing larger volumes ofdata, ensuring scalability while maintaining optimal speed and accuracy levels.

3) Enhanced User Interface and Visualization: Enhance the user interface and visualization features of the tool to offer users intuitive dashboards, interactive reports, and graphical representations of scan results, simplifying interpretation and aiding in decision-making processes.

4) Security and Privacy Considerations: Consistently evaluate and improve the security protocols of the tool to mitigate potential vulnerabilities and adhere to privacy regulations, safeguarding sensitive data throughout the scanning process.

V. ACKNOWLEDGMENT

We would like to express our sincere gratitude to all those who have contributed to the completion of this research paper on network scanning tools using C++. We extend our appreciation to our supervisors and mentors for their guidance and support throughout the research process. Additionally, we acknowledge the valuable insights and resources provided by various academic publications and research papers that have significantly enriched our understanding of network security and scanning methodologies. Finally, we would like to thank our colleagues and peers for their collaboration and input, which have been instrumental in shaping this project.

REFERENCES

[1] Xu, Yunjing, et al. "CANVuS: Context-aware network vulnerability scanning." Recent Advances in Intrusion Detection: 13th International Symposium, RAID 2010, Ottawa, Ontario, Canada, September 15-17, 2010. Pro- ceedings 13. Springer Berlin Heidelberg, 2010.

[2] Gadge, Jayant, and Anish Anand Patil. "Port scan de- tection." 2008 16th IEEE international conference on networks. IEEE, 2008.

[3] Wei-Hua, Jiang, Li Wei-Hua, and Du Jun. "The applica- tion of ICMP protocol in network scanning." Proceed- ings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technolo- gies. IEEE, 2003.

[4] Kaushik, Atul Kant, Emmanuel S. Pilli, and R. C. Joshi. "Network forensic system for port scanning attack." 2010 IEEE 2nd International Advance Computing Con- ference (IACC). IEEE, 2010.

[5] Hong, Qiao, et al. "Retracted: A Network Scanning Detection Method Based on TCP Flow State." 2018 3rd International Conference on Smart City and Systems Engineering (ICSCSE). IEEE, 2018.

[6] Fuchsberger, Andreas. "Intrusion detection systems and intrusion prevention systems." Information Security Technical Report 10.3 (2005): 134-139.

[7] Antipov, Ivan, Tetyana Vasilenko, and Ivan Mikheev. "DEVELOPING WI-FI NETWORK MODEL FOR IN- TRUSION PREVENTION." Eastern-European Journal of Enterprise Technologies 1.9 (2014).

[8] Brekke, Morten, and Per Henrik Hogstad. "New teaching methods-Using computer technology in physics, mathe- matics and computer science." International Journal of Digital Society (IJDS) 1.1 (2010): 17-24.

[9] Singh, Gopal, Sachin Goyal, and Ratish Agarwal. "Intru- sion detection using network monitoring tools." Avail- able at SSRN 2426105 (2014).

[10] Othman, Suad Mohammed, et al. "Survey on intrusion detection system types." International Journal of Cyber- Security and Digital Forensics 7.4 (2018): 444-463.

[11] Holm, Hannes, et al. "A quantitative evaluation of vul- nerability scanning." Information Management Com- puter Security 19.4 (2011): 231-247.

[12] Abu Bakar, R., Kijsirikul, B. (2023). Enhancing Net- work Visibility and Security with Advanced Port Scan- ning Techniques. Sensors, 23(17), 7541.

[13] Xia, Wenfeng, et al. "A survey on software-defined networking." IEEE Communications Surveys Tutorials 17.1 (2014): 27-51.

[14] Sahay, Rishikesh, Weizhi Meng, and Christian D. Jensen. "The

application of software defined networking on securing computer networks: A survey.” Journal of Network and Computer Applications 131 (2019): 89- 108.
[15] Gao, Ke, Cherita Corbett, and Raheem Beyah. ”A pas- sive approach to wireless device fingerprinting.” 2010 IEEE/IFIP International Conference on Dependable Sys- tems Networks (DSN). IEEE, 2010.
[16] Lasˇtovicˇka, Martin, et al. ”Passive operating system fin- gerprinting revisited: Evaluation and current challenges.” Computer Networks 229 (2023): 109782.
[17] Li, Ruoshi, Markus Sosnowski, and Patrick Sattler. ”An overview of os fingerprinting tools on the internet.” Network 73 (2020): 73-77.
[18] Xu, Q., Zheng, R., Saad, W. and Han, Z., 2015. Device fingerprinting in wireless networks: Challenges and op- portunities. IEEE Communications Surveys Tutorials, 18(1), pp.94-104.
[19] Arkin, Ofir. ”A remote active OS fingerprinting tool using ICMP.” login: the Magazine of USENIX and Sage 27.2 (2002): 14-19.
[20] Ashoor, Asmaa Shaker, and Sharad Gore. ”Difference between intrusion detection system (IDS) and intrusion prevention system (IPS).” Advances in Network Security and Applications: 4th International Conference, CNSA 2011, Chennai, India, July 15-17, 2011 4. Springer Berlin Heidelberg, 2011.