Packet-Level Analysis of Network Reconnaissance Techniques Using Nmap and Wireshark

Frankie Rodriguez Rivera
Master in Computer Science
Advisor: Dr. Jeffrey Duffany, PhD.
Polytechnic University of Puerto Rico
Graduate Project EXPO, February 2026

Abstract *- This project reports an applied cybersecurity study that examines network reconnaissance methods, packet-level traffic inspection, and weaknesses in authentication security. This study aims to examine how reconnaissance and scanning activities appear in network traffic and how defenders can identify and interpret these patterns for cybersecurity monitoring and response. The project uses industry-standard tools, including Nmap and Wireshark, to examine network behavior in controlled, authorized settings. The early stages involve network discovery and service enumeration, where an attacker maps active hosts, identifies open ports, and infers operating system traits. This project tests the proposed scanning behavior using real network traffic and shows that scans produce clear traffic patterns that can be detected by network monitoring tools and intrusion detection systems. The project assesses weaknesses in authentication by running password-cracking exercises and reviewing cryptographic hashing, with a focus on the risks created by weak credentials. This project combines reconnaissance, packet analysis, secure authentication, and cryptographic checks to connect cybersecurity theory with hands-on defensive practice. It stresses the need for strong network visibility and layered security controls.*

Key Terms *– Authentication Security, Cryptographic Hashing, Network Reconnaissance, Nmap, Packet Analysis, TCP Flags, Wireshark.*

INTRODUCTION

As digital systems become more interconnected, cybersecurity is a central concern for both organizations and individuals. As networks grow and change, they can create new chances for attackers to gain unauthorized access to sensitive data or interrupt services. Many cyberattacks begin with reconnaissance, where attackers collect key details about possible targets; recognizing this early step helps guide the design of appropriate defense strategies. In network security work, reconnaissance refers to methods used to find which hosts are active and to identify their open ports, running services, and operating systems. Tools such as Nmap support this work by allowing users to carry out network discovery and service enumeration in a systematic way [1]. Even though these tools can give a broad overview, they often hide how the system actually communicates through network protocols, which can expose details relevant to identifying security weaknesses. Packet analysis is necessary to close this gap. Wireshark is a network protocol analyzer that lets cybersecurity researchers capture and review traffic in detail, which supports the study of protocol behavior and the identification of common patterns in network activity [2]. This analysis matters for understanding how reconnaissance appears in live networks and for spotting early indicators of possible exploitation. This project uses a practical, investigative approach to study how network reconnaissance, packet-level traffic inspection, and authentication security relate to one another. This project uses Nmap and Wireshark to examine how different scanning methods produce distinct network traffic patterns and how security monitoring systems can detect those patterns. The study examines authentication security and explains that weak credentials can greatly increase the risks linked to reconnaissance activities. This project seeks to explain how defenders can spot, study, and

reduce reconnaissance and attack preparation activities, so cybersecurity practitioners can better protect modern networks.

BACKGROUND

In many cyberattacks, modern networks are a common starting point because exposed services, devices, and traffic patterns can be observed and tested remotely by attackers. In many real-world cases, attackers do not start by exploiting a weakness. They start with reconnaissance, where they check which hosts are reachable, which ports are open, what services and versions are exposed, and what operating systems or device types are in use. This early-stage activity matters because it usually happens before defenders see any clear harm, but it still produces measurable traces that can be detected when an organization has strong visibility and monitoring.

A recurring problem for defenders is that reconnaissance is often treated as nothing more than a tool's output, such as a list of open ports. In practice, reconnaissance is best viewed as a network-protocol exchange: scan outputs arise when a tool sends specific packets and then reads and classifies the replies it receives. To study reconnaissance, it is not enough to rely on summary scan reports; you should examine packet-level traffic, including the TCP three-way handshake, TCP control flags, timing patterns, and fallback methods such as TCP probing when ICMP is blocked.

This project starts from a simple claim: good defensive cybersecurity depends on identifying the network traces that scanning and host discovery leave behind. This is practical when you use tools that are widely accepted in the field.

Nmap is a common tool in network research and security work for host discovery, port scanning, service enumeration, and operating system fingerprinting [3]. It supports several scan types, and each one behaves differently at the transport layer (e.g., how packets are sent and how responses are handled). This includes SYN probes to identify half-open scanning, ACK probes to infer firewall rules, and OS fingerprinting based on several protocol probes.

Wireshark allows researchers to confirm and interpret these behaviors by capturing the packets produced during scans and during routine network traffic. Using Wireshark, a scan can be evaluated using concrete data, such as TCP flag patterns, response codes, retransmission rates, and the mix of observed protocols.

The motivation also reflects a second security point: scanning alone does not break into a system, but it can expose services that may be targeted next, especially when authentication is weak. In practice, once attackers find exposed services (for example, remote login ports, database endpoints, or web admin panels), they often test common defaults and known weaknesses first. If that works, they may move on to actions such as gaining access, collecting data, or using the system to reach other internal resources that refer to the acceleration due to gravity. After probing access points such as SSH, web-based admin panels, and UPnP, attackers often shift to credential-based attempts, including password guessing, dictionary-based guessing, and trying passwords exposed in prior data leaks. This is why a focus on authentication, password strength, basic password-cracking exercises, and careful handling of credentials fits within the same project. It follows naturally from the initial reconnaissance and shows how quickly exposure can lead to compromise when passwords are weak. The project ends by relying on cryptographic hashing, including the avalanche effect, as a core part of its defensive design. Hashing is a basic method for checking data integrity and for designing secure authentication systems, as long as it is applied correctly. Showing that even small changes to an input produce very different digests help explain why integrity checks can detect tampering, and why older algorithms such as MD5 and SHA-1 are unsuitable for security-critical use today.

This project is based on three related cybersecurity needs:

- Reconnaissance is usually the first stage of an attack cycle. Security teams should identify these early probing activities as soon as possible so they can respond before later stages begin.
- Packet-level inspection is needed to understand and identify scanning behavior, rather than relying only on the scanner's reported results.
- Weak authentication increases the risk of reconnaissance, and principles of cryptographic integrity can guide the design of stronger defensive controls.

This foundation presents the project as a single workflow aligned with routine security operations: assess exposure through reconnaissance, confirm behavior with traffic inspection, assess downstream risk during authentication, and strengthen trust controls through hashing and integrity checks. leave same words, just separate paragraphs

PROBLEM

Network reconnaissance is often the first step in a cyberattack. It is sometimes missed or misunderstood by defenders because it does not cause immediate harm or visible disruption. In many settings, reconnaissance is treated as the output of scanning tools, such as lists of open ports or identified services, instead of being examined as the network traffic patterns that produce those results. This can cause security teams to miss early signs of malicious activity or to misclassify reconnaissance traffic as normal background traffic.

A related issue is that scanning methods can produce different packet-level signatures, especially in their TCP control flags, the way targets respond, and the protocols they use. Without inspecting traffic at the packet level, it is hard to determine whether the observed activity is host discovery, low-profile port scanning, firewall probing, or operating system fingerprinting. This constraint reduces how well intrusion detection systems, firewall rules, and network monitoring approaches can work.

Reconnaissance can increase security risk when authentication controls are weak. After identifying publicly reachable services, attackers often move to credential attacks, and weak or reused passwords can quickly give them access to the system. Many organizations underestimate how quickly information gathered during reconnaissance can lead to an attack when passwords are weak and cryptographic practices are inadequate.

This project focuses on a basic gap: we do not have a clear, packet-level view of what reconnaissance and scanning look like on real networks. It also examines how these early actions, together with weak authentication, can make it more likely that an attacker will succeed.

NETWORK RECONNAISSANCE USING NMAP

In this phase of the project, Nmap was used to perform authorized network reconnaissance within a controlled local environment and against an approved external test host. The reconnaissance process began with network identification, where the local subnet was defined as 192.168.0.0/24 based on the scanning host's IP configuration. Establishing clear network boundaries ensured that all scanning activity remained ethical, authorized, and focused. The wireless router at 192.168.0.1 was identified as the primary target due to its role as the network gateway.

Host discovery scans were then conducted to identify active devices on the local network. Using Nmap's ping scan functionality, multiple online hosts were detected, including the router, the scanning host, and additional consumer and IoT devices. This step provided a baseline understanding of the network topology and confirmed which systems were viable targets for deeper analysis.

Following host discovery, port and service enumeration was performed using TCP SYN scanning techniques. The wireless router was found to expose common management and network services, including HTTP (80/tcp), HTTPS

(443/tcp), and UPnP (5000/tcp). These findings illustrated how reconnaissance can quickly reveal a device's attack surface and identify potential entry points for further probing or exploitation.

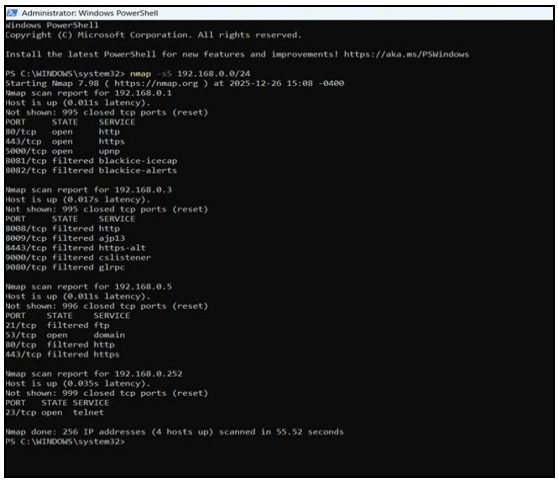

Figure 1
Nmap TCP SYN Scan Results Showing Open and Filtered Ports on Local Network Hosts

Operating system detection was also evaluated using Nmap's OS fingerprinting features. When conditions permitted, Nmap inferred that certain devices were running embedded or Linux-based operating systems. The accuracy of these results depended on firewall behavior and the availability of both open and closed ports, highlighting how defensive controls influence reconnaissance outcomes.

Multiple Nmap scan types were compared to demonstrate how scan selection affects visibility and detectability. SYN scans efficiently identified open ports while minimizing connection completion, ACK scans provided insight into firewall filtering behavior, fast scans reduced scan duration at the cost of detail, and aggressive scans combined several techniques to produce comprehensive system information. Dedicated OS fingerprinting scans further demonstrated Nmap's ability to infer platform characteristics based on TCP/IP stack behavior.

To validate reconnaissance against an external system, the authorized host scanme.nmap.org was scanned. The results confirmed the presence of open services such as SSH and HTTP, demonstrating Nmap's effectiveness for external reconnaissance while adhering to acceptable use policies.

From a defensive perspective, this section emphasized that host discovery and port scanning generate identifiable traffic patterns that can be detected by IDS, firewalls, and network monitoring tools. Early detection of such activity allows defenders to assess intent and respond before exploitation occurs. Overall, this reconnaissance phase established a clear foundation for subsequent packet-level analysis by showing how scanning activity exposes network structure and services while leaving observable traces in network traffic.

Network Traffic Analysis Using Wireshark

This section presented a structured analysis of network traffic using Wireshark with the objective of understanding how network communication manifests at the packet level during normal operations and reconnaissance-related activity. Through a controlled one-minute capture, real traffic generated by the system was observed, allowing examination of data encapsulation from the data link layer through the application layer.

The capture methodology was designed to reflect realistic network conditions, avoiding artificial traffic generation or simulated scenarios. During the capture period, several thousand packets were recorded, illustrating the typical behavior of a modern local network characterized by extensive use of reliable transport protocols, frequent domain name resolution, and a predominance of encrypted communications.

Protocol analysis revealed a distribution consistent with a standard operational environment. TCP accounted for most of the reliable, connection-oriented traffic, while UDP was associated primarily with discovery services and background processes. DNS traffic appeared consistently because of automated system and application queries, and ARP activity reflected normal address resolution within the local network. The absence of

unencrypted HTTP traffic further confirmed the widespread adoption of security mechanisms in contemporary applications.

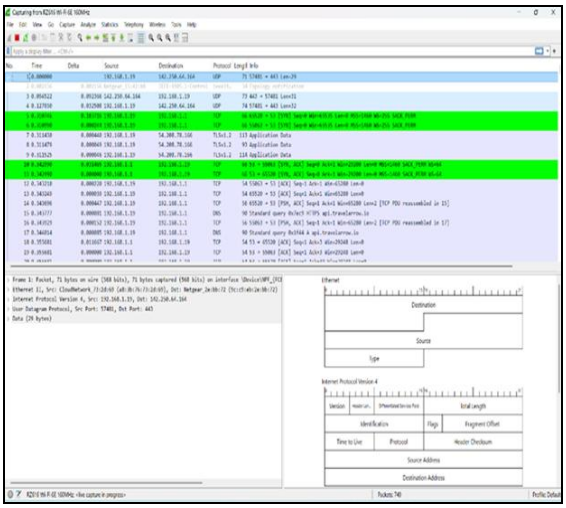

Figure 2
Live Packet Capture Displaying TCP, UDP, DNS, and TLS Traffic in Wireshark

Filtering techniques demonstrated their value as an analytical tool by reducing background noise and enabling focused inspection of specific traffic flows. Through the use of both inclusion and exclusion filters, relevant traffic was isolated, the absence of certain protocols was verified, and the dominance of encrypted protocols such as TLS and QUIC was clearly observed. This highlighted the importance of filtering as a critical capability in forensic investigations and security monitoring.

The use of advanced analysis and statistical features provided an aggregated perspective of network behavior that complemented packet-level inspection. Conversation and endpoint statistics identified primary communication nodes, while I/O graphs and protocol hierarchy views facilitated interpretation of traffic trends and protocol distribution. These tools illustrated how large volumes of packet data can be transformed into meaningful and actionable network intelligence.

Finally, preserving packet captures in standard formats ensured reproducibility of the analysis and enabled later review or collaboration with other analysts. Collectively, this section demonstrated how Wireshark can convert raw network traffic into valuable security insights, establishing a foundation for anomaly detection, incident analysis, and validation of reconnaissance activity examined in subsequent sections of the project.

Advanced Packet-Level Correlation

This section examined how different Nmap scanning techniques manifest at the packet level by correlating scan activity with observed TCP flag behavior in Wireshark. By capturing live traffic during multiple scan executions, it was possible to directly observe how reconnaissance techniques interact with the transport layer and how their behavior can be distinguished through packet inspection.

Packet-level correlation was performed using a local host within a private IPv4 network while targeting a publicly accessible system configured for scanning research. Across all scans, consistent and repeatable TCP flag patterns were observed that reflected the intended purpose of each scanning technique. Focusing on the initial packets generated by each scan proved especially effective, as these packets most clearly conveyed probing intent while minimizing interference from background traffic.

The analysis demonstrated that different scan types generate distinct transport-layer signatures. SYN-based scans produced half-open connection attempts characterized by SYN packets without full session establishment, while ACK-based scans generated unsolicited ACK probes indicative of firewall rule testing rather than service discovery. Fast and aggressive scans exhibited similar SYN-centric behavior but varied in volume and diversity, reflecting their respective optimization goals and expanded reconnaissance scope. Host discovery scans relied on TCP probing when ICMP was unavailable, and operating system fingerprinting generated the most diverse traffic, incorporating multiple protocols beyond TCP alone.

Results confirmed that reconnaissance activity leaves observable and interpretable traces at the packet level. TCP flag behavior provided a reliable

mechanism for differentiating between host discovery, firewall testing, service probing, and operating system identification. Variations in packet sequencing and responses were attributable to target system behavior and network conditions rather than inconsistencies in scanning methodology.

Overall, this analysis demonstrated that packet-level inspection is an effective approach for validating scan behavior and identifying reconnaissance activity in live networks. By correlating scan intent with transport-layer behavior, defenders can leverage protocol analysis to detect, classify, and respond to scanning activity, reinforcing the importance of packet-level visibility in both offensive security testing and defensive monitoring environments.

Authentication and Password Security

This section examined the role of authentication weaknesses as a critical escalation point following successful network reconnaissance. Once exposed services and authentication endpoints are identified through scanning activity, weak credential practices significantly increase the likelihood of compromise. The analysis demonstrated that predictable passwords, credential reuse, and insufficient complexity transform reconnaissance from passive information gathering into an effective attack pathway [4].

The prevalence of weak passwords within real-world datasets highlights a persistent security gap [5]. Analysis of commonly used credential lists revealed that simple, easily guessable passwords remain widespread, drastically reducing the effort required for credential-based attacks. Practical testing further demonstrated that password-protected resources secured with low-entropy credentials offer minimal resistance, even when encryption mechanisms are technically present. These findings reinforce the principle that encryption strength is fundamentally undermined by weak authentication controls.

To validate these concepts in practice, a password-protected Microsoft Word document was analyzed using John the Ripper within a Windows PowerShell environment. The document's encryption hash was extracted using a Python-based utility and subjected to an offline dictionary-based attack using a common password list. The password was successfully recovered as "abc", illustrating how weak credentials can be compromised quickly once access points are identified. This exercise demonstrated how authentication weaknesses can directly follow reconnaissance activity and lead to system exposure.

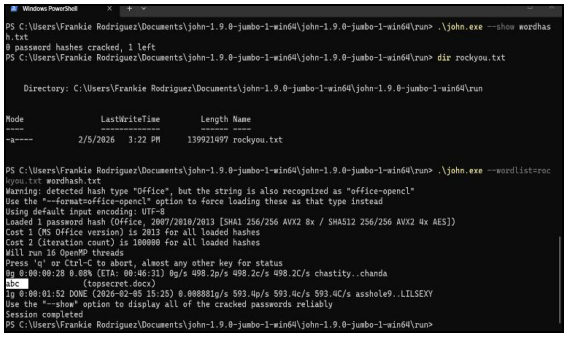

Figure 3

Password Cracking Results Obtained using John the Ripper and SHA-1 Hash Generation Illustrating How Weak, Low-Entropy Credentials can be Recovered Quickly Once Authentication Data is Exposed

Cryptographic Hashing and Data Integrity

This section explored cryptographic hash functions as a foundational mechanism for ensuring data integrity and authenticity. Hashing algorithms were examined for their deterministic behavior, fixed-length output, and sensitivity to input variation. Identical inputs consistently produced identical hashes, while even minor changes resulted in entirely different outputs, confirming the reliability of hash functions as digital fingerprints for data verification.

A key focus of the analysis was the avalanche effect, which ensures that minimal input changes yield drastically different hash values. This property was demonstrated by hashing nearly

identical inputs that differed by only a single character, resulting in hash outputs with virtually no similarity. The observed behavior illustrates why cryptographic hashes are effective at detecting tampering, preventing partial inference, and supporting integrity validation.

While legacy algorithms such as MD5 and SHA-1 are no longer suitable for cryptographic security due to collision vulnerabilities, their behavior remains useful for illustrating fundamental hashing principles [6] [7]. Modern systems rely on stronger algorithms, such as SHA-256 and SHA-3, which preserve these properties while providing enhanced resistance to cryptographic attacks. Hashing remains central to secure system design, supporting password storage, file integrity monitoring, digital signatures, and trust validation across contemporary computing environments.

Together, these sections demonstrate that authentication strength and data integrity are inseparable from effective cybersecurity defense. Weak credentials negate the protections offered by encryption, while cryptographic hashing ensures that data authenticity and integrity can be verified even in hostile environments. When combined with strong password policies and modern cryptographic standards, these mechanisms form a critical defensive layer that limits the impact of reconnaissance and reduces the risk of successful exploitation.

Figure 4
SHA-1 Hash Comparison of Nearly Identical Input Strings Showing the Avalanche Effect, where a Single-Character Change Produces Significantly Different Hash Values, Supporting Data Integrity Verification

RESULTS AND DISCUSSION

This project found that network reconnaissance and attack preparation create consistent packet-level patterns that can be observed and interpreted. In both the local network and an authorized external test network, Nmap scans generated different TCP/IP traffic patterns, which I recorded and examined in Wireshark. These results show that reconnaissance leaves measurable traces in network traffic, which can support detection and later forensic review when monitoring is set up and maintained properly.

In the host discovery and port-scanning phase, we mainly relied on TCP SYN probes to determine which hosts were active and which services were reachable on open ports. Packet captures repeatedly showed half-open attempts: SYN packets arrived, but the three-way handshake was never completed, which is consistent with stealthy scanning. By contrast, ACK-based scans returned specific flag patterns that do not show whether a service is reachable; instead, they point to how a firewall is filtering the traffic. These differences show that the aim of a scan shapes the packet layout, and that close checks of TCP flags can separate goals such as finding live hosts, probing firewall rules, and listing exposed services.

Aggressive scanning and OS fingerprinting produced more complex network traffic, with several protocols in use and a wider range of flag settings. These scans used a mix of TCP, UDP, and ICMP probes, which produced a wider variety of packets than simpler scans. When restrictive display filters were applied, some scan elements did not appear at first. This suggests that traffic analysis should consider multiple protocols and views, rather than depending on a single-protocol view. From a defense standpoint, this supports the need for monitoring approaches that can spot reconnaissance that spans multiple protocols.

Wireshark captures showed that even during routine operation, modern networks can produce large amounts of traffic over very short periods of time. The heavy use of encrypted protocols like

TLS and QUIC, along with the lack of unencrypted HTTP traffic, matches the security standards most modern applications follow today. Display filters and statistical tools were needed to separate the reconnaissance traffic we cared about from the general background noise. Wireshark's statistical views summarized protocol distribution, endpoint activity, and traffic patterns, making it easier to identify overall trends than by examining packets one at a time.

The authentication and password security review showed that weak credentials greatly increase the risks identified during reconnaissance. We examined hashed password samples with John the Ripper. A dictionary attack recovered weak, commonly reused passwords in seconds, which suggests many users chose low-entropy credentials. These findings suggest that encryption and hashing offer only partial protection when authentication controls are weak. The password-strength evaluation suggests that strong security is driven mainly by entropy, randomness, and password length, not by how easy a password is to remember or how simple it looks. It also supports the view that reconnaissance is often only the opening phase of a longer attack sequence. In our cryptographic hashing experiments, we examined how integrity checks help keep secure systems trustworthy. Hash values were computed using online tools and Python hashing functions to confirm that the output was deterministic and to check the results independently. Two nearly identical input strings that differed by one character produced SHA-1 hashes that differed at 39 of the 40 hexadecimal positions, providing empirical support for the avalanche effect. This behavior shows why cryptographic hashes can detect tampering and support checks of data authenticity.MD5 and SHA-1 are not suitable for security-critical systems because their collision weaknesses are well established. Still, looking at how these older algorithms behave can help explain core hashing ideas that also appear in newer cryptographic designs.

Taken together, these results suggest that reconnaissance activity, weaknesses in authentication, and integrity checks are closely linked and should be considered as parts of the same security problem. In a security assessment, network scanning can show which services are exposed and what basic system details they reveal. Weak passwords or default logins can make it easy for an attacker to gain access quickly. The design and use of cryptographic controls then influence whether unauthorized changes to data can be noticed and verified. Knowing how these factors relate to each other matters for both offensive security testing and defensive network protection. When paired with strong authentication rules and current cryptographic standards, monitoring and hashing help add a key layer of defense by lowering the chance that an attacker can exploit the system successfully.

CONCLUSION

This project demonstrated that packet-level inspection can be used to study network reconnaissance, traffic analysis, authentication security, and cryptographic integrity mechanisms, and to draw clear cybersecurity insights from observed behavior. Nmap scans were paired with Wireshark traffic captures to compare theoretical expectations with network behavior observed in controlled local tests and on authorized external systems.

The observed relationship between scan types and TCP/IP packet behavior indicates that common reconnaissance tools leave consistent, identifiable patterns in network traffic. Analysis of TCP flags, protocol usage, and packet sequencing showed that different scanning methods serve distinct objectives and can be identified through standard network monitoring techniques. These results confirm that packet-level visibility can help detect early indicators of attack activity before exploitation occurs.

The project also demonstrated that reconnaissance alone does not imply system

compromise; however, weak authentication significantly increases the likelihood of successful attacks once services are identified. Password analysis and cracking exercises showed that low-entropy credentials can undermine the protections offered by encryption. Cryptographic hashing experiments illustrated how integrity mechanisms help prevent undetected data modification and support trust and verification in modern systems.

In conclusion, this project supports the importance of defense-in-depth as a foundational cybersecurity principle. Network visibility, packet analysis, strong authentication controls, and sound cryptographic practices must work together to reduce security risk. By linking theoretical concepts with practical observation, this study provides insight relevant to cybersecurity practitioners tasked with detecting, assessing, and responding to network-based threats in real-world environments.

FUTURE WORK

This project offers a detailed packet-level analysis of reconnaissance activity and authentication security, but some topics still need further study. Future research could apply this analysis to enterprise-scale settings that include intrusion detection systems, intrusion prevention systems, and advanced firewall technologies. Examining how reconnaissance traffic is identified, recorded, or stopped in these environments would offer clearer insight into how well defenses perform in real settings.

Future work could add automated detection tools such as Suricata or Snort to link packet-level measurements with alert output and rule-based detection results. This would help link raw packet analysis with day-to-day security monitoring tasks in a security operations center.

Future research should broaden the authentication analysis to cover multi-factor authentication, defenses against credential stuffing, and current password-hashing methods such as bcrypt, scrypt, and Argon2. Assessing how these mechanisms hold up against post-reconnaissance attacks would improve the project's defense-focused analysis.

A next step would be to expand the cryptographic analysis to cover digital signatures, certificate checks, and secure key management, so the study can better explain how networked systems build trust and protect data integrity. These next steps would extend the work in this project and help clarify how reconnaissance activity, authentication security, and cryptographic controls influence one another in complex, real-world cybersecurity settings.

REFERENCES

[1] A. Nath, *Packet analysis with Wireshark: Leverages the power of Wireshark to troubleshoot your networking issues by using effective packet analysis techniques and performing an improved protocol analysis* [Online Book], Birmingham, UK: Pack. Publishing, 2015. Available: https://books.google.com.pr/books?hl=es&lr=&id=UzzlCwAAQBAJ&oi=fnd&pg=PP1&dq=wireshark+case+studies+pdf&ots=fk1QnSlGl4&sig=mc1OFvqrgyk1zuR-_9EcUky6htc&redir_esc=y#v=onepage&q&f=false.

[2] TryHackMe. (n. d.). *Wireshark: The Basics* [Online]. Available: https://tryhackme.com/room/wiresharkthebasics.

[3] L. Bock. *Learn Wireshark: a definitive guide to expertly analyzing protocols and troubleshooting networks using Wireshark*, 2nd ed., 2022. Available: https://books.google.com.pr/books?hl=es&lr=&id=4HF5EAAAQBAJ&oi=fnd&pg=PP1&dq=wireshark+case+studies+pdf&ots=Clob1_OKuX&sig=4xc5W-_QGvzl25HTkSdKzAbBDeI&redir_esc=y#v=onepage&q&f=false.

[4] Wikipedia. (2025, Nov. 24). *Brute-force attack* [Online]. Available: https://en.wikipedia.org/wiki/Brute-force_attack.

[5] Wikipedia. (2026, Feb. 7). *Password cracking* [Online]. Available: https://en.wikipedia.org/wiki/Password_cracking.

[6] Wikipedia. (2025, Nov. 3). *MD5* [Online]. Available: https://en.wikipedia.org/wiki/MD5.

[7] Wikipedia. (2026, Jan. 22). SHA-1 [Online]. Available: https://en.wikipedia.org/wiki/SHA-1