Proceedings of the 5th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/133/2025.20599

Enhancing the Security of Transmission Control Protocol (TCP): Challenges and Solutions for Modern Network Threats

Hongyi Chen[1,a,*]

[1]*Alcanta International College Board, Guangzhou Nansha, China*
a. stephenchen2019@163.com
**corresponding author*

Abstract: Transmission Control Protocol (TCP), the backbone of internet communication, ensures reliable, connection-oriented data transmission. Despite its widespread use in areas such as email, web browsing, and file transfer, TCP faces significant security vulnerabilities stemming from its design era, which prioritized functionality over security. Common threats include TCP sequence number prediction, session hijacking, SYN flood attacks, and TCP Reset attacks. Existing mitigation strategies, such as TCP-AO, SSL/TLS encryption, and network-based security measures like IDS/IPS, have reduced risks but face challenges like performance overhead and compatibility issues. This study reviews the root causes of TCP vulnerabilities, evaluates existing solutions, and highlights gaps in addressing threats within modern network architectures. While current measures are effective to an extent, future research must explore advanced technologies such as quantum cryptography, blockchain-based authentication, and AI-driven anomaly detection to enhance TCP security and adaptability. This work underscores the urgent need for interdisciplinary collaboration and innovation to secure TCP in evolving digital ecosystems.

Keywords: TCP security, session hijacking, denial-of-service attack, SSL/TLS, protocol optimization

1. Introduction

Transmission Control Protocol (TCP), as the core protocol for internet communication, is responsible for providing reliable, connection-oriented data transmission services [1]. It is widely used in critical areas such as email, web browsing, and file transfer. However, TCP was designed in the 1970s, when the network environment was simpler, and security concerns were not a primary focus by then. As the internet rapidly expanded and network attacks became more sophisticated, TCP exposed several security flaws that bring threats to the stability and reliability of modern networks. Vulnerabilities such as TCP sequence number prediction, session hijacking, SYN flood attacks, and TCP Reset attacks are some of the most prevalent threats. A CERT study indicates that the frequency of TCP sequence number prediction attacks has been increasing since 2000 [2]. Additionally, SYN flood attacks remain the most common form of Distributed Denial of Service (DDoS), causing significant economic losses and resource waste.

DOI: 10.54254/2755-2721/133/2025.20599

Both academia and industry have proposed a range of improvements to mitigate these threats, including TCP-AO (Authentication Option), SSL/TLS encryption, and real-time monitoring through IDS/IPS systems. However, these solutions face challenges such as performance overhead and compatibility issues in real-world deployments. Meanwhile, with the advent of 5G, the Internet of Things (IoT), and Software-Defined Networking (SDN), TCP security issue has become more complex, and the need to overcome it becomes more urgent. The security of TCP not only impacts the reliability of current network services but also the development of internet, the deeper research and innovation from both academia and industry is in urgent need. This study aims to investigate the security vulnerabilities inherent in the TCP protocol and to evaluate current advancements in mitigating these risks. This paper will use a comprehensive review of the literature on TCP vulnerabilities and associated attack methodologies will be conducted. Also, this research will focus on identifying the root causes of TCP vulnerabilities, analyzing the effectiveness of existing mitigation strategies, and proposing potential enhancements for future implementations.

2. Related Research on TCP Security

2.1. TCP Vulnerabilities

Research on TCP vulnerabilities has primarily focused on its design flaws and the security issues arising from its implementation. One of the most widely exploited vulnerabilities is TCP sequence number prediction [3]. According to RFC 1948, the lack of randomness in the initial sequence number generation algorithm allowed attackers to predict the sequence numbers of future sessions by observing patterns in historical communication. While modern operating systems have improved random number generation algorithms, the same issue can still occur in low-entropy systems. SYN flood attacks exploit the TCP three-way handshake process to launch denial-of-service attacks by sending a large volume of forged SYN requests in order to exhaust server resources. According to Akamai's 2022 DDoS report, SYN flood attacks account for over 50% of all DDoS incidents [4]. The root cause of these issues lies in the original TCP design, which did not consider potential attacks. The acknowledge of the root cause also provides a clear direction for future improvements. Recent research has focused on identifying variants of these vulnerabilities in emerging network environments such as IoT and SDN, and solutions proposed for them.

2.2. TCP Security Protocol Enhancements

To address TCP's security weaknesses, several enhancement protocols have been proposed, including TCP-AO (Authentication Option) and SSL/TLS [5]. TCP-AO, defined in RFC 5925, is an upgrade of the TCP MD5 signature option, supporting stronger hashing algorithms and flexible key management mechanisms. Studies show that TCP-AO significantly reduces the success rate of spoofing attacks compared to traditional TCP MD5 and is suitable for dynamic routing protocols like BGP. However, its high key management costs and performance impacts pose challenges in practical deployment. SSL/TLS, an encryption layer on top of TCP, provides end-to-end encryption and authentication, effectively preventing eavesdropping and tampering [5]. According to Google's Transparency Report, as of 2023, over 90% of web traffic globally is encrypted via HTTPS (based on SSL/TLS). However, SSL/TLS can cause latency and additional resource overhead, particularly in high-traffic environments.

2.3. Network-based Security Measures

In addition to enhancing TCP itself, efforts on the study of network-based security measures such as Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) have been extensively

DOI: 10.54254/2755-2721/133/2025.20599

conducted. By monitoring network traffic in real time and analyzing packet characteristics, IDS/IPS can identify abnormal behavior and block attacks. Modern machine learning-based IDS systems can detect TCP SYN flood and sequence number prediction attacks. Though the accuracy has been over 98%, false positives remain a challenge. DDoS protection technologies, such as traffic scrubbing, differentiate malicious from legitimate traffic, are able to filter out attack requests. According to Cloudflare's 2023 report, TCP-based traffic scrubbing systems can mitigate over 80% of SYN flood attacks in under 10 seconds. Dynamic rate limiting and blacklisting mechanisms have also been proven to be effective in real-world applications.

2.4. System Patches and Secure Implementation

The operating system's implementation of TCP has direct impacts on its security, and different operating systems show varying TCP on their security mechanisms [6]. For instance, the Linux kernel introduced SYN Cookie technology, which significantly mitigates SYN flood attacks, while Windows relies more on rate limiting and queue management mechanisms. Research shows that SYN Cookies perform well in high-traffic attack scenarios, while may introduce processing delays for legitimate traffic. In recent years, automated vulnerability detection and patching systems have rapidly evolved. Static analysis tools allow researchers to quickly identify TCP security vulnerabilities in operating system implementations. The "TCP SACK PANIC" vulnerability exposed in 2021, caused by improper handling of SACK options in the Linux kernel, was quickly patched, and has significantly reduced related attack incidents. Regular patch updates are crucial to preventing TCP attacks, particularly in the face of zero-day vulnerabilities. By adhering to unified standards and best practices, implementation discrepancies that pose security risks can be effectively minimized.

3. Overview of TCP

3.1. Basic Mechanics

TCP is a connection-oriented, reliable transport layer protocol widely used for end-to-end data transmission across the internet. Its operation consists of three key stages: connection establishment, data transmission, and connection termination. In the connection establishment phase, TCP uses a three-way handshake to ensure both parties are capable of reliable data exchange [6]. This process involves the exchange of SYN and ACK flags to synchronize initial sequence numbers and verify the connection's validity, confirming that both sides are ready for data transmission. During the data transmission phase, TCP ensures reliable and ordered delivery of data using sequence and acknowledgment numbers. Each packet is assigned a unique sequence number, and the receiver sends acknowledgment numbers to indicate the successful receipt of data. If a packet is lost or corrupted, the receiver requests re-transmission, ensuring data integrity. TCP also employs flow control via the sliding window mechanism, adjusting the sender’s transmission rate to avoid overwhelming the receiver and prevent network congestion. Finally, in the connection termination phase, TCP uses a four-way handshake to safely close the communication, ensuring that both sides have finished transmitting data before the connection is fully terminated.

3.2. TCP Header Structure

The TCP header contains essential information for connection management, data transmission, and error detection. The fixed portion of the TCP header is 20 bytes long and includes fields such as source and destination ports, sequence and acknowledgment numbers, data offset, flags, window size, checksum, and urgent pointer. The sequence number indicates the position of the first byte of the data segment within the stream, helping the receiver reorder packets correctly and ensuring data is

DOI: 10.54254/2755-2721/133/2025.20599

delivered in the correct order. The acknowledgment number indicates the next expected byte, providing feedback on the successful reception of data.

The flags in the TCP header include control flags such as SYN, ACK, FIN, and RST, which are used for connection establishment, confirmation, termination, and resetting. The window size field helps with flow control by informing the sender of the receiver's available buffer space, allowing the sender to adjust the transmission rate dynamically [7]. The checksum ensures the integrity of the entire packet, detecting errors in both the header and the data. Optional fields, such as timestamps for round-trip time (RTT) measurement and Selective Acknowledgments (SACK) for partial re-transmissions, provide additional functionality. However, the openness of the TCP header structure introduces security vulnerabilities, as these fields can be exploited in attacks like sequence number prediction, packet forgery, and TCP hijacking.

3.3. Core Features of TCP

TCP's core features—reliability, flow control, and congestion control—ensure efficient data transmission in diverse network conditions. Reliability is achieved through acknowledgment for each sent packet, if an acknowledgment is not received, the packet is re-transmitted. TCP also reorders out-of-order packets to ensure data is delivered correctly. Flow control prevents the sender from overwhelming the receiver by dynamically adjusting the sending rate with the sliding window protocol. The receiver updates the window size in real time based on its buffer status, optimizing resource use and transmission efficiency. Congestion control helps TCP adapt to network conditions by adjusting the sending rate to avoid congestion. Using algorithms like slow start, congestion avoidance, fast re-transmit, and fast recovery, TCP manages network load and improves overall transmission performance.

4. TCP Vulnerabilities and Challenges

4.1. Sequence Number Prediction

Sequence number prediction is an attack that exploits the predictable nature of the initial sequence number generation in early TCP implementations, which lacked sufficient randomness. Attackers could predict the sequence numbers of future sessions by observing historical communication patterns, allowing them to inject malicious data or hijack sessions [8]. The Morris Worm incident is one well-known case where attackers successfully exploited TCP sequence number prediction to breach systems. According to CERT research, attackers were able to predict and forge valid sequence numbers within seconds by collecting enough network data samples.

Although modern operating systems have improved the algorithms for generating sequence numbers, this vulnerability persists in low-entropy systems, such as IoT and embedded devices, which often rely on weak or inadequate random number generators. Sequence number prediction can lead to severe consequences, including session hijacking, unauthorized data injection, and potential security breaches, such as man-in-the-middle attacks or Denial of Service (DoS) attacks [8].

4.2. Session Hijacking

Session hijacking is an attack that takes control of a communication session by exploiting vulnerabilities in TCP connections. It often combines with TCP sequence number prediction, where the attacker predicts the sequence number, forges packets, and impersonates one of the parties to inject or modify data. This can have serious consequences, especially in scenarios like remote login (e.g., SSH) or file transfer (e.g., FTP), where attackers can steal sensitive information or execute malicious commands [9]. The main cause of session hijacking is the lack of authentication in the TCP

DOI: 10.54254/2755-2721/133/2025.20599

design. TCP only verifies IP addresses and port numbers, which are transmitted in plaintext and can be easily intercepted. According to a 2022 Cybersecurity Report, around 30% of network attacks exploit session hijacking on unencrypted TCP connections. Even with encryption, attackers can hijack sessions if encryption keys are leaked via man-in-the-middle (MITM) attacks.

To defend against session hijacking, using SSL/TLS for encryption ensures data integrity and authentication. Improving TCP's authentication, such as through digital signatures, can also help. However, due to the complexity of modern network infrastructure, fully addressing this issue remains challenging.

4.3. Denial-of-Service (DoS) Attacks

Denial-of-Service (DoS) attacks are one of major threats to the TCP protocol. These attacks take advantage of TCP's three-way handshake by sending a large number of forged SYN requests, which forces the target server to exhaust its resources and become unable to handle legitimate user connections. This type of attack, known as a SYN Flood, is one of the most common forms of DoS attacks [10]. The destructive power of SYN Flood attacks stems from their simplicity and efficiency. The attacker only needs to forge the source IP address and send SYN requests without completing the handshake, causing the server to allocate resources while blocking legitimate connection attempts. Although modern operating systems have reduced the impact of such attacks by introducing SYN Cookies—technology that encodes connection state information in the sequence number to prevent unnecessary resource allocation—this solution still carries a performance overhead, especially in high-traffic environments.

To mitigate DoS attacks, network defenses like traffic filtering, blacklisting, and dynamic rate limiting are commonly used [10]. Additionally, defense systems based on distributed architectures can quickly adapt and distribute traffic during an attack. However, these solutions come with additional computational and bandwidth costs, and they cannot fully eliminate the risk of attacks. Therefore, further optimization of the TCP protocol is necessary to enhance its resilience against resource depletion attacks.

4.4. TCP Reset Attacks

TCP Reset (RST) attacks involve sending forged RST packets to forcibly terminate legitimate connections. These attacks are simple and effective, as the attacker only needs to send a packet with the RST flag to both parties, disrupting the connection without advanced skills or privileges. They are particularly damaging to long-running services like BGP (Border Gateway Protocol) and video streaming. By predicting or sniffing TCP sequence numbers, attackers can forge Reset packets, causing victims to mistakenly close the connection. The vulnerability that enables these attacks lies in the lack of packet authenticity checks in TCP. The attacker only needs to forge a valid source IP address, port number, and sequence number to succeed. According to CERT reports, such attacks have increased, particularly in public networks. A notable case was the 2008 Reset attack on YouTube, which caused major service disruptions. The primary defense is using packet signing or encryption, such as the TCP-AO protocol, to ensure integrity and authenticity. Additionally, intrusion detection systems (IDS) at network boundaries can monitor for abnormal RST traffic and help mitigate the risk of these attacks.

4.5. TCP Spoofing Attacks

TCP spoofing attacks involve forging source addresses and packet contents to bypass network security measures and gain unauthorized access to a target system. Attackers exploit the trust inherent in the TCP protocol, often masquerading as a trusted host to send malicious packets that deceive the

DOI: 10.54254/2755-2721/133/2025.20599

target system into performing harmful actions. These attacks can lead to data leakage, service disruption, and loss of system control. The key to TCP spoofing lies in predicting TCP sequence numbers and forging IP addresses. In trust-based networks, the target system typically does not verify the sender's identity, allowing forged packets to bypass security checks. A well-known example is the "Mitnick attack," where the attacker used TCP spoofing to bypass an access control list (ACL) and take control of the victim's system.

The primary defense against TCP spoofing is strengthening identity verification, such as using certificate-based authentication. Additionally, configuring firewalls and intrusion detection systems (IDS) at network boundaries to block packets with forged IP addresses can help mitigate the risk. While these measures reduce the likelihood of successful TCP spoofing, the inherent design of the TCP protocol still leaves it vulnerable to continuous security threats.

5. Countermeasures and Future Directions

5.1. End-to-End Encryption with SSL/TLS

Using SSL/TLS for end-to-end encryption is a key defense against TCP security threats. SSL/TLS protects against eavesdropping, data tampering, and man-in-the-middle (MITM) attacks by providing encryption, authentication, and message integrity checks. Operating above TCP, SSL/TLS uses asymmetric encryption to establish a secure channel, then switches to symmetric encryption for efficient data protection [5]. According to Google Transparency Report, over 90% of global web traffic was encrypted with HTTPS (SSL/TLS) by 2023, highlighting the importance of this technology. SSL/TLS mitigates many TCP vulnerabilities, making attacks like sequence number prediction and session hijacking difficult due to encrypted, untraceable data. However, SSL/TLS can introduce performance overhead, especially during the handshake phase in high-traffic environments. Poor SSL/TLS configurations, such as outdated encryption algorithms, can still expose vulnerabilities. Continued deployment and optimization of SSL/TLS, including hardware acceleration and lightweight versions for resource-constrained devices, will further enhance TCP security in diverse network environments.

5.2. Improving the TCP Protocol

Enhancing the TCP protocol's built-in security is essential to addressing its vulnerabilities. The original TCP design lacked mechanisms for identity verification and data integrity, making it vulnerable to attacks like sequence number prediction, Reset attacks, and SYN Floods. One proposed solution is the TCP-AO (Authentication Option) protocol, which replaces the traditional TCP MD5 signature with stronger encryption and authentication. TCP-AO supports advanced hashing algorithms and flexible key management, improving resistance to spoofing. Additionally, improving TCP's congestion control and state management, such as using SYN Cookies to defend against SYN Flood attacks, has proven effective, though it can impact performance for legitimate traffic. Research also explores encrypting or signing TCP header fields to protect critical information. These improvements enhance TCP security, but challenges remain in compatibility and performance overhead. Future work will focus on lightweight security mechanisms and maintaining backward compatibility for broader adoption.

5.3. Deploying IDS/IPS Systems

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are essential for network-level defense against TCP threats. These systems monitor traffic in real time, analyzing packet patterns to detect and respond to anomalies. Modern IDS/IPS systems often integrate machine

DOI: 10.54254/2755-2721/133/2025.20599

learning to improve detection accuracy for attacks like SYN Floods and sequence number prediction. Research shows that deep learning-based IDS can achieve over 98% accuracy in detecting abnormal TCP traffic. IDS/IPS systems identify suspicious patterns, forge packets, and block potential attacks, such as dynamically adjusting traffic thresholds during SYN Floods or blocking suspicious IP addresses. However, traditional IDS/IPS systems may face performance bottlenecks in high-traffic networks and can generate false positives, blocking legitimate traffic. Distributed IDS/IPS solutions can address these challenges by sharing analysis tasks across multiple nodes, reducing single points of failure. Next-generation IDS/IPS systems will incorporate AI for adaptive learning, improving detection accuracy and response times. As attack techniques evolve, IDS/IPS will need to become more intelligent to handle increasingly complex TCP threats.

5.4. Timely System Patches

Regular system patching is crucial to maintaining TCP security by fixing known vulnerabilities and reducing the risk of successful attacks. For example, operating system patches have addressed vulnerabilities like TCP Reset attacks and SACK PANIC. Research shows that servers failing to apply patches are more likely to be targeted. In 2022, over 40% of cyberattacks were linked to unpatched systems. Patching not only addresses security flaws but also optimizes TCP implementation, improving overall system performance. For instance, the continuous improvement of the SYN Cookie mechanism in Linux has mitigated SYN Flood threats. However, patching can lead to compatibility and performance issues, especially in large-scale deployments. Automated patch management systems enhance patch efficiency and coverage, enabling organizations to quickly address new threats. Future research will focus on developing smarter patch prioritization systems and improving the adaptability of automated patch tools in various network environments to ensure ongoing TCP security.

6. Conclusion

This study highlights key vulnerabilities, such as TCP sequence number prediction, session hijacking, SYN flood attacks, and TCP Reset attacks, which continue to be exploited in various forms, particularly in emerging environments like IoT and 5G networks. While advancements like TCP-AO, SSL/TLS encryption, and network-based security measures such as IDS/IPS have mitigated many of these risks, challenges such as performance overhead, compatibility issues, and false positives remain. System-level enhancements, including secure implementations and regular patching, are critical in addressing these vulnerabilities.

However, the study discusses general vulnerabilities in TCP, it may not fully address the specific challenges posed by emerging network architectures like IoT, SDN, and 5G. These environments introduce unique characteristics and risks that require more targeted research. Future studies should Investigate advanced and emerging security measures, such as quantum cryptography, blockchain-based network authentication, and AI-driven anomaly detection, to assess their feasibility and effectiveness in securing TCP connections.

References

[1] *De Almeida, L. F. F., Pereira, L. A. M., Sodré, A. C., Mendes, L. L., Rodrigues, J. J., Rabelo, R. A., & Alberti, A. M. (2020). Control networks and smart grid teleprotection: Key aspects, technologies, protocols, and case-studies. IEEE Access, 8, 174049-174079.*

[2] *POPOOLA, Olugbemiga Solomon. "An Overview of the Evolutionary and Revolutionary Trends of Computer Network Intrusion and Detection." Available at SSRN 4532805 (2023).*

[3] *Adedeji, K. B., Abu-Mahfouz, A. M., & Kurien, A. M. (2023). DDoS attack and detection methods in internet-enabled networks: Concept, research perspectives, and challenges. Journal of Sensor and Actuator Networks, 12(4), 51.*

DOI: 10.54254/2755-2721/133/2025.20599

[4] Dummer, S., & Rath, S. (n.d.). A retrospective on DDoS trends in 2023 and actionable strategies for 2024. Akamai. Retrieved from https://www.akamai.com/blog/security/a-retrospective-on-ddos-trends-in-2023

[5] Oppliger, R. (2023). SSL and TLS: Theory and Practice. Artech House.

[6] Fall, K. R., & Stevens, W. R. (2012). Tcp/ip illustrated (Vol. 1). Addison-Wesley Professional.

[7] Nyangaresi, V. O., Ogara, S. O., & Abeka, S. O. (2017). TCP IP header attack vectors and countermeasures.

[8] Acharya, S., & Tiwari, N. (2016). Survey of DDoS attacks based on TCP/IP protocol vulnerabilities. IOSR Journal of Computer Engineering, 18(3), 68-76.

[9] Baitha, A. K., & Vinod, S. (2018). Session hijacking and prevention technique. Int. J. Eng. Technol, 7(2.6), 193-198.

[10] Zlomislić, V., Fertalj, K., & Sruk, V. (2017). Denial of service attacks, defences and research challenges. Cluster Computing, 20, 661-671.