Understanding the TCP Three-Way Handshake in Computer Networking

Author: Sk. Shihab

Internship ID: ITID1897

Internship Domain: Offensive Cyber Security

1. Abstract

This research paper explores the TCP Three-Way Handshake, a fundamental mechanism that establishes reliable connections over the Internet. As part of the *Offensive Cyber Security Internship* at InLighnX Global Pvt. Ltd., this paper aims to provide a clear understanding of how TCP sessions are initiated, ensuring data integrity and communication reliability. The handshake process involves three critical steps—SYN, SYN-ACK, and ACK—which enable both the client and server to synchronize and acknowledge connection parameters before data transmission begins. This mechanism not only underpins the majority of network communication protocols but also plays a crucial role in cybersecurity, where attackers often exploit it through techniques like SYN flood attacks. Understanding how the handshake works is essential for aspiring cybersecurity professionals, especially in offensive roles, to both test for vulnerabilities and design effective countermeasures. This paper offers both theoretical and practical insights into one of the building blocks of secure networking.

2. Introduction

The Transmission Control Protocol (TCP) is one of the foundational protocols of the Internet, providing reliable, connection-oriented communication between devices [1]. At the core of TCP's reliability lies the **three-way handshake**, a mechanism used to establish and synchronize connections before data transmission begins. This handshake process ensures both devices agree on initial parameters, enabling an orderly, loss-checked exchange of data packets [2].

The three-way handshake plays a critical role not only in network stability but also in cybersecurity, especially in **offensive cybersecurity**, where understanding protocol behavior is key to identifying vulnerabilities. Attackers often exploit misconfigurations or behaviors during this handshake phase to launch reconnaissance scans, spoofed connections, or Denial-of-Service (DoS) attacks [3]. For example, tools such as Nmap and hping3 use crafted TCP SYN packets to fingerprint systems and detect open ports, based on how they respond during the handshake process [4].

In this research paper, we explore the technical working of the TCP three-way handshake in detail—its steps, structure, and use cases. We also examine potential **security weaknesses** associated with this process and how it is leveraged in offensive security assessments. The goal is to provide a clear, technical understanding of the handshake for cybersecurity interns and students, enabling them to recognize both its necessity and its exploitable points.

3. What is the TCP Three-Way Handshake?

Definition

The **TCP Three-Way Handshake** is the process used by two devices to establish a reliable connection over the Transmission Control Protocol (TCP). It involves an exchange of three specific messages—**SYN**, **SYN-ACK**, and **ACK**—between the client and server to initiate and synchronize communication. This handshake ensures that both devices agree on initial parameters such as sequence numbers before data is transmitted [1].

The handshake is described in **RFC 793**, the original specification of TCP, as a "procedure used to establish a connection" that prevents unauthorized or unsynchronized communication [1]. Modern educational resources also explain that this handshake is the foundation for TCP's reliable, connection-oriented nature [2][3].

Establishing a Reliable Connection

TCP is a **connection-oriented** protocol, meaning it requires a setup phase before data transfer begins. The handshake plays a crucial role in this phase by:

- Confirming both sides are ready to communicate,
- Exchanging and acknowledging **Initial Sequence Numbers (ISNs)**,
- Ensuring that no old or duplicate packets interfere with the connection [3][4].

This process prevents issues such as miscommunication or data loss. Without the handshake, packets could arrive out of order, be duplicated, or go completely unnoticed. As RFC 793 notes, the handshake "reduces the possibility of false connections" and ensures that both endpoints are properly synchronized [1].

4. Three-Way Handshake Steps

The steps involved in the TCP Three-Way Handshake are as follows:

i. **Client → Server: SYN**

The client initiates the connection by sending a TCP segment with the **SYN (synchronize)** flag set. This message includes the client's Initial Sequence Number (ISN). It is a request to establish a connection and synchronize sequence numbers.

ii. **Server → Client: SYN-ACK**

The server responds with a segment containing both the **SYN** and **ACK (acknowledgment)** flags. It acknowledges the client's SYN by setting the acknowledgment number to the client's ISN + 1 and also includes its own ISN for synchronization in the reverse direction.

iii. **Client → Server: ACK**

Finally, the client sends a segment with only the **ACK** flag set, acknowledging the server's SYN by setting the acknowledgment number to the server's ISN + 1. No actual data is sent yet, but the connection is now fully established [2][3][4].

Once this exchange is complete, both devices have synchronized sequence numbers and enter the **ESTABLISHED** state, ready to transfer application data reliably and in order [2].

5. Steps of the TCP Three-Way Handshake

In TCP, establishing a connection requires a three-way handshake [8]. The handshake uses three packets: **SYN**, **SYN-ACK**, and **ACK**, named for "synchronize", "synchronize-acknowledge", and "acknowledge" respectively [8]. This three-message exchange ensures that both client and server agree on each other's initial sequence numbers before any data is exchanged.

Table 1: *Steps of the TCP Three-Way Handshake*

Step	**Description**
SYN	The client sends a SYN (synchronize) packet to initiate the connection.
SYN-ACK	The server replies with a SYN-ACK (synchronize-acknowledge) to acknowledge the SYN and provide its own sequence number.

Step	Description
ACK	The client sends an ACK (acknowledge) to confirm the server's sequence number, completing the handshake.

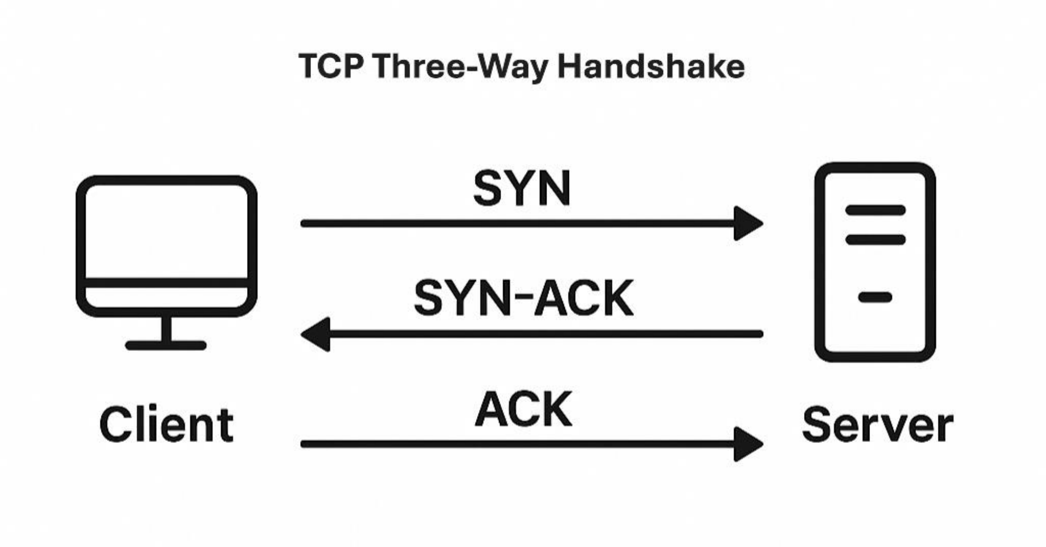

Figure 1: *Diagram illustrating the TCP three-way handshake sequence (client-server exchange of SYN, SYN-ACK, ACK).*

In **Step 1 (SYN)**, the client initiates the connection by sending a TCP packet with the **SYN** flag set [6]. This packet includes the client's **Initial Sequence Number (ISN)**, usually chosen randomly. It signals the intent to begin communication and awaits a response from the server.

In **Step 2 (SYN-ACK)**, the server replies with a packet that has both the **SYN and ACK** flags set [6]. The ACK confirms the client's SYN by setting the acknowledgment number to the client's ISN + 1. At the same time, the server provides its own ISN in the SYN part of the message, completing its part of the negotiation.

In **Step 3 (ACK)**, the client sends a final TCP packet with the **ACK** flag set [6]. This packet confirms receipt of the server's SYN by setting the acknowledgment number to the server's ISN + 1. After this, both parties have acknowledged each other's sequence numbers, and the TCP connection is now fully established.

6. Why is it Important?

The TCP three-way handshake is a **critical component of reliable communication** over the Internet. It forms the foundation of how computers establish trustworthy connections before any data is transferred [7]. Without this handshake, two devices would have no way to confirm each other's readiness or ensure that data is delivered in order.

One key function of the handshake is to let both parties **agree on essential connection parameters**, such as **Initial Sequence Numbers (ISNs)**. These sequence numbers help track the order of data packets and ensure that each segment of information arrives without duplication or loss [8]. If packets arrive out of order or are duplicated, TCP can use these numbers to reorder or discard them as needed—ensuring consistency and reliability.

Another reason the handshake is important is because it **prevents false or half-open connections**. For example, if a device crashes and reboots without properly closing a TCP session, the three-way handshake helps prevent miscommunication by requiring fresh confirmation from both parties before re-establishing a session [7].

Moreover, the TCP handshake is widely used in many everyday applications and services. **Web browsing (HTTP/HTTPS)**, **file transfers (FTP)**, **remote access (SSH)**, **email (SMTP/IMAP)**, and **secure transactions** all rely on TCP's connection-oriented design, which starts with this three-step handshake [9]. Each of these services depends on the reliability and order-preserving nature of TCP, which would not be possible without a proper connection setup.

7. Real-World Applications

The TCP three-way handshake is not just a theoretical protocol concept—it plays a vital role in our daily digital interactions. Every time a user visits a website using **HTTP or HTTPS**, the browser first performs a TCP handshake with the server before any webpage data is transferred [10]. This handshake ensures that both client and server are ready for reliable communication before delivering content.

Similarly, when connecting to a server via **SSH (Secure Shell)** for remote command-line access, the handshake is performed before authentication or data transfer begins. **Email clients** (like Outlook or Thunderbird) also rely on TCP to communicate with servers using protocols such as **SMTP**, **IMAP**, or **POP3** [11]. Other tools such as **FTP (File Transfer Protocol)** and **remote desktop applications** depend on this process to establish secure and stable sessions.

These use cases highlight how the TCP handshake underpins most forms of reliable communication on the Internet, making it a fundamental building block of modern networking.

8. Security Considerations

While the TCP three-way handshake ensures reliable communication, it also opens a door for certain **security vulnerabilities**, especially in the context of offensive cybersecurity. One of the most well-known issues is the **SYN flood attack**, a type of **Denial-of-Service (DoS)** attack. In this attack, an attacker sends a high volume of SYN packets without completing the handshake (i.e., not sending the final ACK), causing the server to hold open half-complete connections. This can **consume system resources** and **prevent legitimate users** from connecting [12].

To combat such attacks, **firewalls** and **Intrusion Prevention Systems (IPS)** are used to detect and block suspicious patterns of TCP handshake behavior. These systems often track the rate and completeness of TCP connection attempts and filter out malicious traffic [13].

Another defensive technique is the use of **TCP SYN cookies**, a method by which the server encodes the state of the connection in the initial SYN-ACK response. This allows the server to avoid storing any state until it receives the final ACK from the client, **mitigating the memory exhaustion** risk during SYN flood attacks [14].

Understanding these risks and mitigations is essential for cybersecurity professionals and researchers exploring how attackers exploit protocol behavior—and how defenders can respond.

9. Conclusion

The **TCP three-way handshake** is a foundational process that ensures reliable, ordered, and synchronized communication between two devices in a network. By exchanging **SYN**, **SYN-ACK**, and **ACK** packets, TCP ensures that both the client and server are ready and agree on connection parameters before any real data is transferred.

In modern networking, this handshake is critical. Whether you're browsing a website, sending an email, or remotely accessing a server, the handshake silently powers your interaction. It ensures **data integrity**, **session establishment**, and **robustness** across networks that are often unpredictable and noisy.

Beyond functionality, the handshake plays a central role in **network security**. It forms the basis for trust between two communicating devices, but it's also a point of vulnerability exploited in certain types of cyberattacks. As such, understanding the handshake is not only crucial for network engineers but also for cybersecurity professionals working to defend modern digital infrastructure.

In essence, the TCP three-way handshake is more than a technical process; it's the first agreement that two machines make before starting any meaningful digital conversation.

10. References

Internship Materials & Learning Resources

- **[15]** InLighnX Tech. *Introduction to Basic Networking Concepts.* InLighn Tech Internship Material (PDF).
- **[16]** InLighnX Tech. *Introduction to Sockets.* Internship PDF.
- **[17]** InLighnX Tech. *TCP/IP Networking Video Lecture Series.*

External Sources

- **[1]** Postel, J. (1981). *Transmission Control Protocol (RFC 793).* Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc793
- **[2]** Codecademy Team. (2023). *The TCP Three-Way Handshake Explained.* Codecademy. https://www.codecademy.com/resources/blog/tcp-3-way-handshake
- **[3]** TutorialsPoint. (2023). *TCP 3-Way Handshake Process.* https://www.tutorialspoint.com/tcp-3-way-handshake-process
- **[4]** WireX Systems Blog. (2021). *TCP: Network Protocol Explained.* https://www.wirexsystems.com/blog/tcp/
- **[5]** Mozilla Developer Network. *TCP Overview.* https://developer.mozilla.org/en-US/docs/Glossary/TCP
- **[6]** Wikipedia Contributors. *Transmission Control Protocol.* Wikipedia. https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- **[11]** Stallings, W. (2013). *Data and Computer Communications* (10th ed.). Pearson.
- **[12]** CERT Coordination Center. (1996). *TCP SYN Flooding and IP Spoofing Attacks.* https://www.kb.cert.org
- **[13]** Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS).* NIST Special Publication 800-94.
- **[14]** Lemon, J. (2002). *Resisting SYN Flood DoS Attacks with TCP SYN Cookies.* FreeBSD Journal.