TCP FLAGS

TCP FLAGS

TCP uses a set of control bits known as **TCP flags** to manage communication between devices.

PURPOSE OF TCP FLAGS

- TCP flags provide control and management capabilities for TCP connections.

- They are used to **initiate**, **maintain**, and **terminate** connections, as well as manage data transmission.

COMMON TCP FLAGS

SYN (Synchronize): Initiates a connection.

ACK (Acknowledgment): Acknowledges received data.

FIN (Finish): Indicates the end of data transmission.

RST (Reset): Resets a connection.

PSH (Push): Pushes data to the application.

URG (Urgent): Marks data as urgent.

COMBINING TCP FLAGS

Multiple TCP flags can be set in a single packet.

Combinations like

SYN-ACK, **RST-ACK**, or **FIN-ACK**

serve specific purposes during connection setup, teardown, and management.

SIGNIFICANCE OF TCP FLAGS

- Understanding TCP flags is essential for ***network administrators***, ***security analysts***, and ***troubleshooting*** network issues.

- Proper flag interpretation aids in ***diagnosing*** network problems and ***optimizing performance***.

TCP THREE-WAY HANDSHAKE

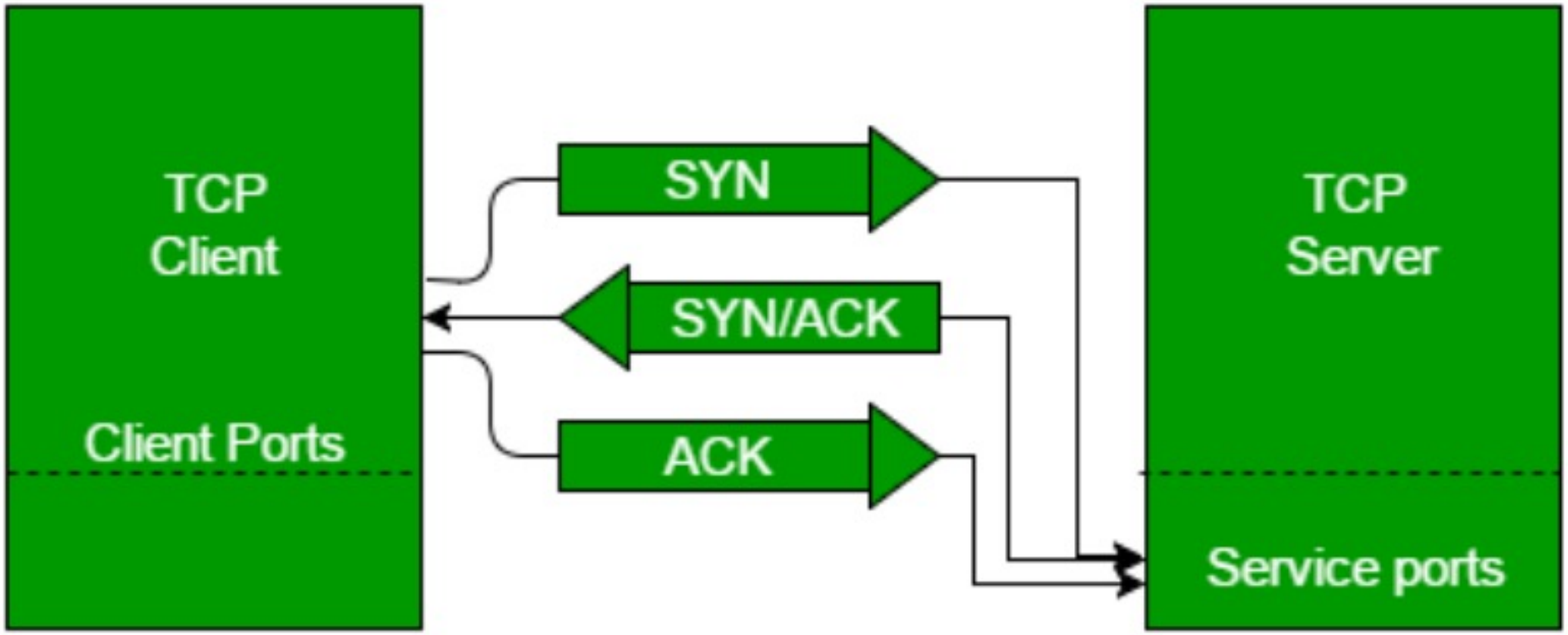
The **TCP Three-Way Handshake** is a fundamental process in TCP communication.

It is used to establish a **reliable connection** between two devices over a network.

PURPOSE

The primary purpose of the Three-Way Handshake is to **ensure both parties agree** on initial sequence numbers and establish **communication** parameters.

3-WAY HANDSHAKE



NOTE

- communication is always initiated by the client from a random port
- Servers communicate from a well-known port

NMAP SCANNING

WHAT IS NMAP

Nmap, short for Network Mapper, is a widely used **open-source** tool and network scanning utility for network discovery and security auditing.

Nmap is available for various operating systems, including Windows, Linux, and macOS.

NMAP USE CASE

- Host discovery
- Port scanning
- OS detection
- Service and version detection
- Network mapping (host mapped to service)

SCAN TYPES

→ TCP Connect Scans (-sT) (nmap –sT 10.10.10.10)

→ SYN “Half-open” or “Stealth” Scans (-sS) (nmap –sS 10.10.10.10)

→ UDP Scans (-sU) (nmap –sU 10.10.10.10)

→ TCP Null Scans (-sN) (nmap –sN 10.10.10.10)

→ TCP FIN Scans (-sF) (nmap –sF 10.10.10.10)

→ TCP Xmas Scans (-sX) (nmap –sX 10.10.10.10)

USAGE

Command: **nmap [target]**

Example: **nmap 192.168.1.1**

PORT SPECIFICATION

- Specify ports to scan using **-p** option.

- Examples:
  - **nmap -p 80,443 192.168.1.1**
  - **nmap -p- 192.168.1.1** (Scans all 65,535 ports)

SERVICE DETECTION

Use **-sV** to detect service versions.

- Example: **nmap -sV 192.168.1.1**

OS FINGERPRINTING

Use **-O** to perform OS detection.

- Example: **nmap -O 192.168.1.1**

SCRIPTING

Nmap supports custom scripts with

-sC for default scripts and

-sV --script [script] for specific ones.

- Example:
- **nmap -sV --script smb-os-discovery 192.168.1.1**
- nmap -sV -p21-1024 --script vulners.nse 192.168.223.132

OUTPUT VERBOSITY

- Adjust the output verbosity with **-v** (increases verbosity) and **-vv** (maximum verbosity).
- Example: **nmap -v -sV 192.168.1.1**

OUTPUT FORMAT

- Nmap offers various output formats like text, XML, and grepable.
- Use **-oX** for XML output.
- Example: **nmap -oX output.xml 192.168.1.1**

TIMING AND PERFORMANCE

- Control scan timing with **-T** option (from 0 to 5).
- Example: **nmap -T4 192.168.1.1** (Aggressive timing)

STEALTH SCANNING

- Use options like **-sS**, **-sA**, or **-sN** for stealthy scans.
- Example: **nmap -sS 192.168.1.1**

INTERACTIVE MODE

- Access interactive mode with **-iL** to read target list from a file.
- Example: **nmap -iL targets.txt**

NMAP DEMO

➤→ NMAP