Wireshark Project Report — TCP 3-Way Handshake & Stealth Scan Analysis

Author: Temitope C Folowosele

Role: Cybersecurity Analyst

Date: October 2025

Lab Host: kali (privileged)

Executive Summary

This report documents a Wireshark analysis performed to observe and validate a TCP 3-way handshake between a host (10.0.2.4) and a destination server (66.29.153.49) on destination port 80. The capture was filtered by TCP port and IP address to isolate the session. Additionally, the analysis covers stealth scanning techniques (stealth/SYN scans, decoy scans, time fragmentation scans), how they can bypass detection, and recommended detection & mitigation strategies.

Objectives

• Capture and identify the complete TCP 3-way handshake (SYN, SYN-ACK, ACK) between 10.0.2.4 and 66.29.153.49 on port 80.

• Demonstrate packet filtering in Wireshark by IP and TCP port (e.g., ip.addr==10.0.2.4 ip.addr==66.29.153.49 && tcp.port==80).

• Explain stealth scan variants (SYN/half-open, RST) and how they evade detection (use of nonstandard flag combos, fragmented/timed probes).

• Provide practical detection and mitigation recommendations.

Environment & Capture Details

Environment: Local lab (VM host: Kali).

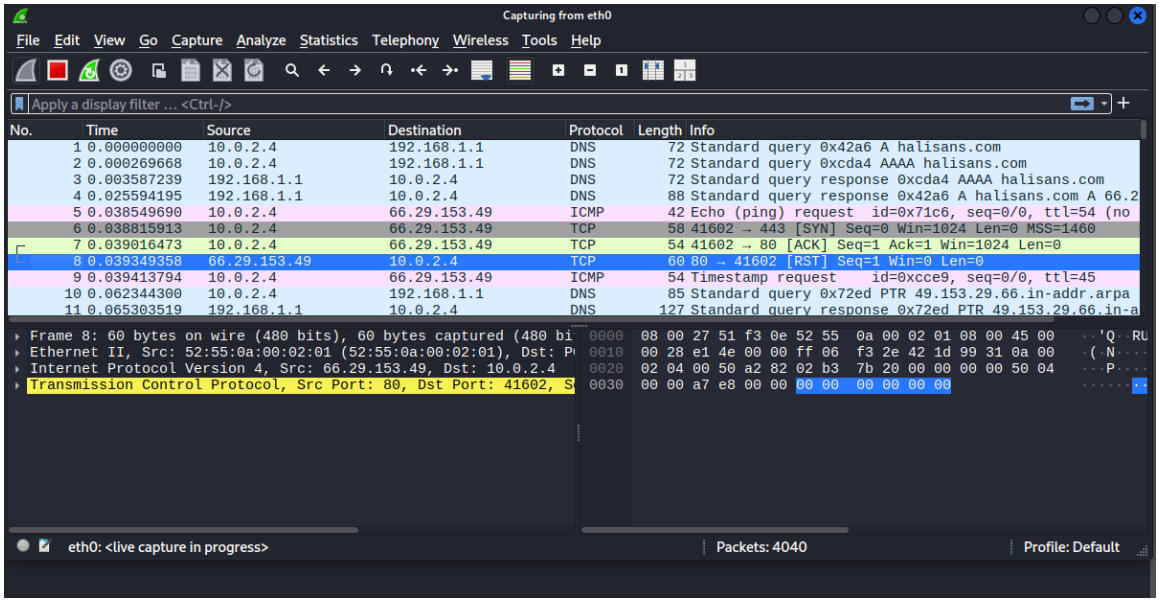
Wireshark version: (placeholder) capture saved as capture_wireshark.pcapng.

Host IP: 10.0.2.4

Destination IP: 66.29.153.49

Target service: HTTP (TCP port 80)

Capture method: Promiscuous mode on the host interface; capture start/end timestamps: (placeholders).



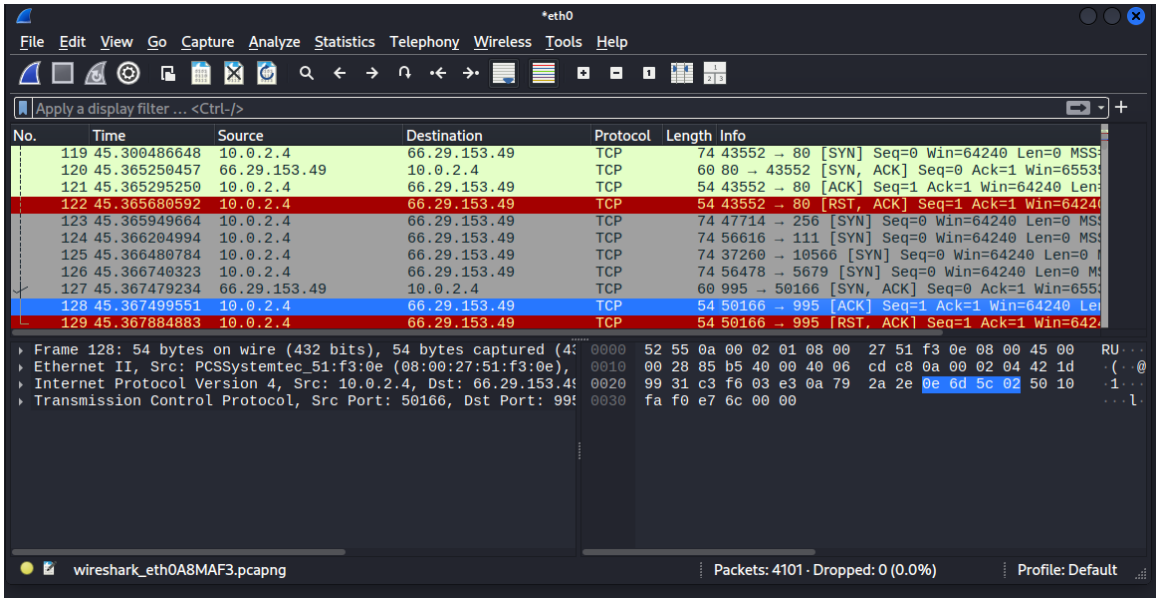
Identifying the 3-Way Handshake

The TCP 3-way handshake can be identified by locating three packets in sequence:

1. SYN — Client (10.0.2.4) sends TCP segment with SYN bit set (tcp.flags.syn==1, tcp.flags.ack==0).

2. SYN-ACK — Server (66.29.153.49) replies with SYN+ACK (tcp.flags.syn==1, tcp.flags.ack==1).

3. ACK — Client (10.0.2.3) sends ACK (tcp.flags.ack==1, tcp.flags.syn==0) to complete the handshake.

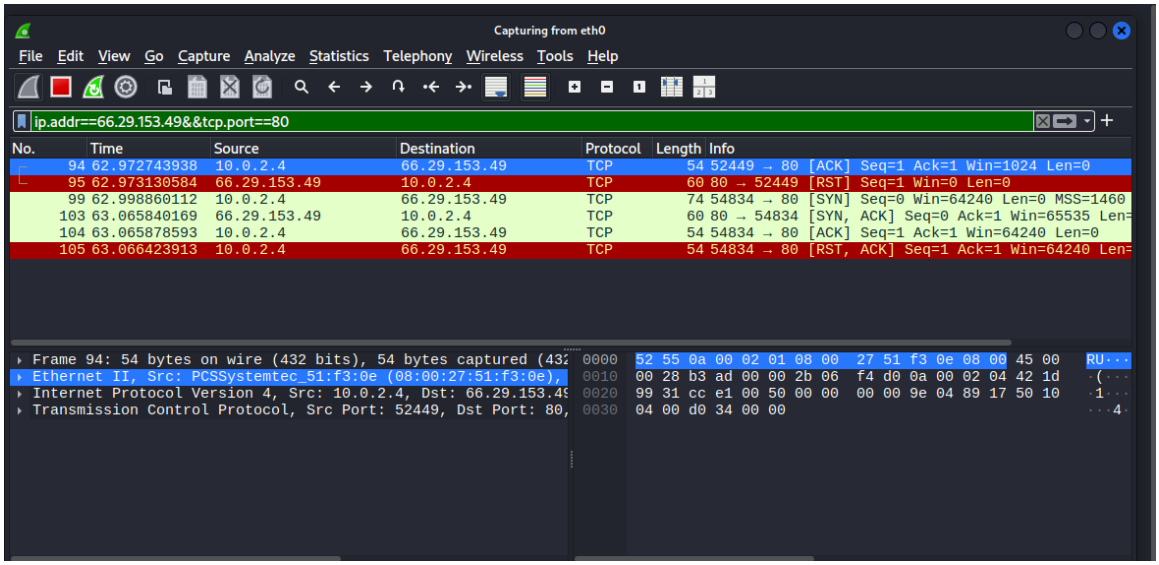


Port Scanning & Filters Used

Port scanning was performed targeting port 80 on 66.29.153.49. To isolate scan traffic, the following Wireshark filters were used:

Filter by source host and port

ip.addr == 66.29.153.49&&tcp.port==80



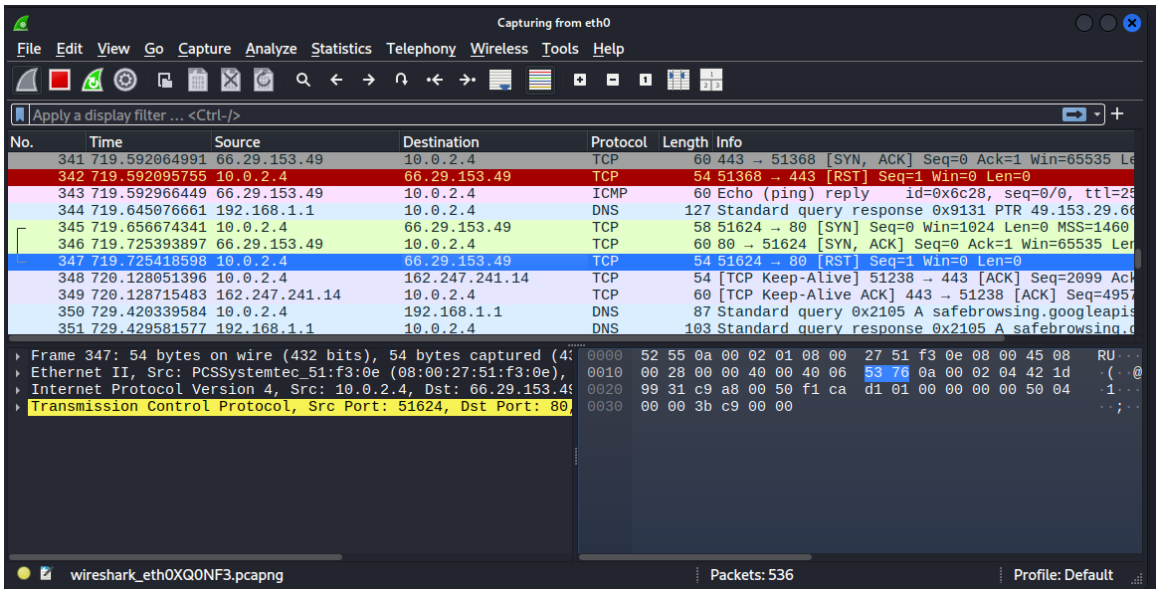
Evasion Techniques Observed / Discussed

The following stealth scan techniques were discussed. These techniques can make detection harder for simple signature-based IDS/IPS or manual review.

1) Stealth (SYN) Scans — Half-Open Scans

Behavior: The attacker sends a SYN and analyzes the response. If SYN-ACK is received the port is open; attacker sends RST instead of completing the handshake (no final ACK), avoiding full connection establishment.

Why it can bypass detection: Some naive detection rules look for completed handshakes or payloads; dropping the handshake completion can avoid certain logging. However, modern IDS/IPS and connection tracking usually detect large volumes of SYNs and incomplete handshakes (SYN floods or unusual SYN/SYN-ACK ratios).

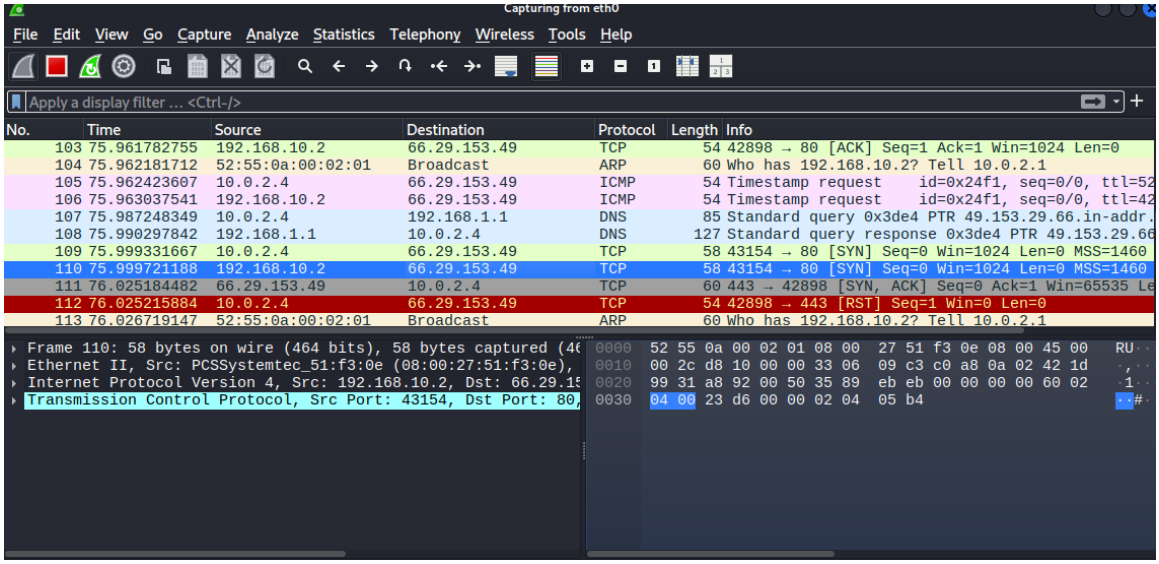


2) Decoy Scans

Behavior: The attacker uses multiple spoofed source IPs (decoys) along with the true source to mix legitimate-looking traffic with malicious probes. This blends probe traffic with benign-looking flows, making attribution and detection harder.

Why it can bypass detection: Alerts generated per-source may be diluted; threshold-based systems may not detect low-rate probes spread across many decoy IPs.

Detection strategies: Correlate destination-side logs, look for identical probe patterns (same TTL, window size, TCP options), and use anomaly detection across multiple sources.

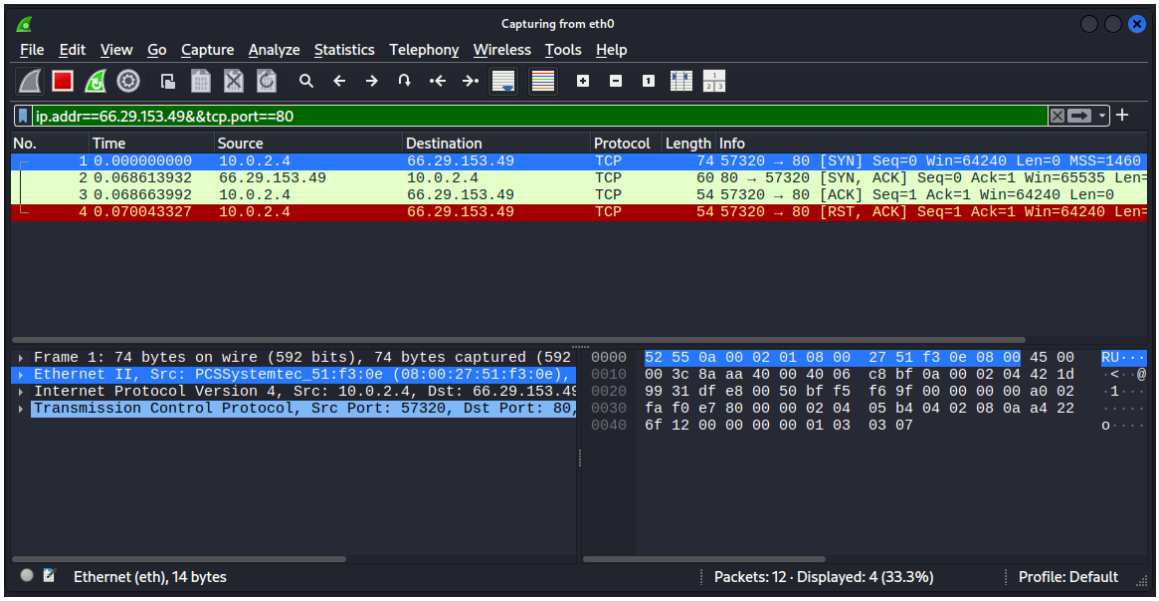


3) Time Fragmentation / Fragmented Scans

Behavior: Attackers split packets into small fragments or send them slowly over time to hide probe payloads. This can evade signature-based detection that inspects single packets.

Why it can bypass detection: If the IDS lacks full IP fragment reassembly or has limited buffer/timeouts, signatures won't match. Time-spaced fragments avoid triggering rate or threshold-based alerts.

Detection strategies: Enable full IP reassembly in IDS/Wireshark, adjust timeouts and buffers, and monitor unusual fragmentation or timing patterns with flow/session correlation.



Detection & Mitigation Recommendations

1. Use stateful network devices and enable connection tracking—this helps detect incomplete handshakes.
2. Enable IP fragment reassembly in IDS/IPS (and ensure adequate buffers/timeouts).
3. Correlate network flow telemetry (NetFlow/sFlow) with packet captures to detect distributed scans at low-rate
4. Implement rate limiting and SYN cookies to mitigate SYN-based evasions and floods.
5. Use behavioral detection (anomaly-based IDS) to spot patterns across decoys or time-sliced probes.
6. Log and centralize alerts; enrich with context (TCP options, TTL, packet sizes) for better triage.

7. Deploy honeypots to attract scans and analyze attacker tools and techniques safely.

Conclusion

The Wireshark analysis verified the TCP 3-way handshake between 10.0.2.4 and 66.29.153.49 on port 80 and demonstrated how attackers can use stealthy scanning techniques to avoid naive detection. Combining packet-level inspection with flow telemetry and behavioral analytics increases detection resilience against decoy and fragmentation-based evasions.