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

Author: Biliaminu B. Salawu
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 SYN, SYN-ACK and ACK between a host (192.168.1.4) and a destination server (66.39.153.12) 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

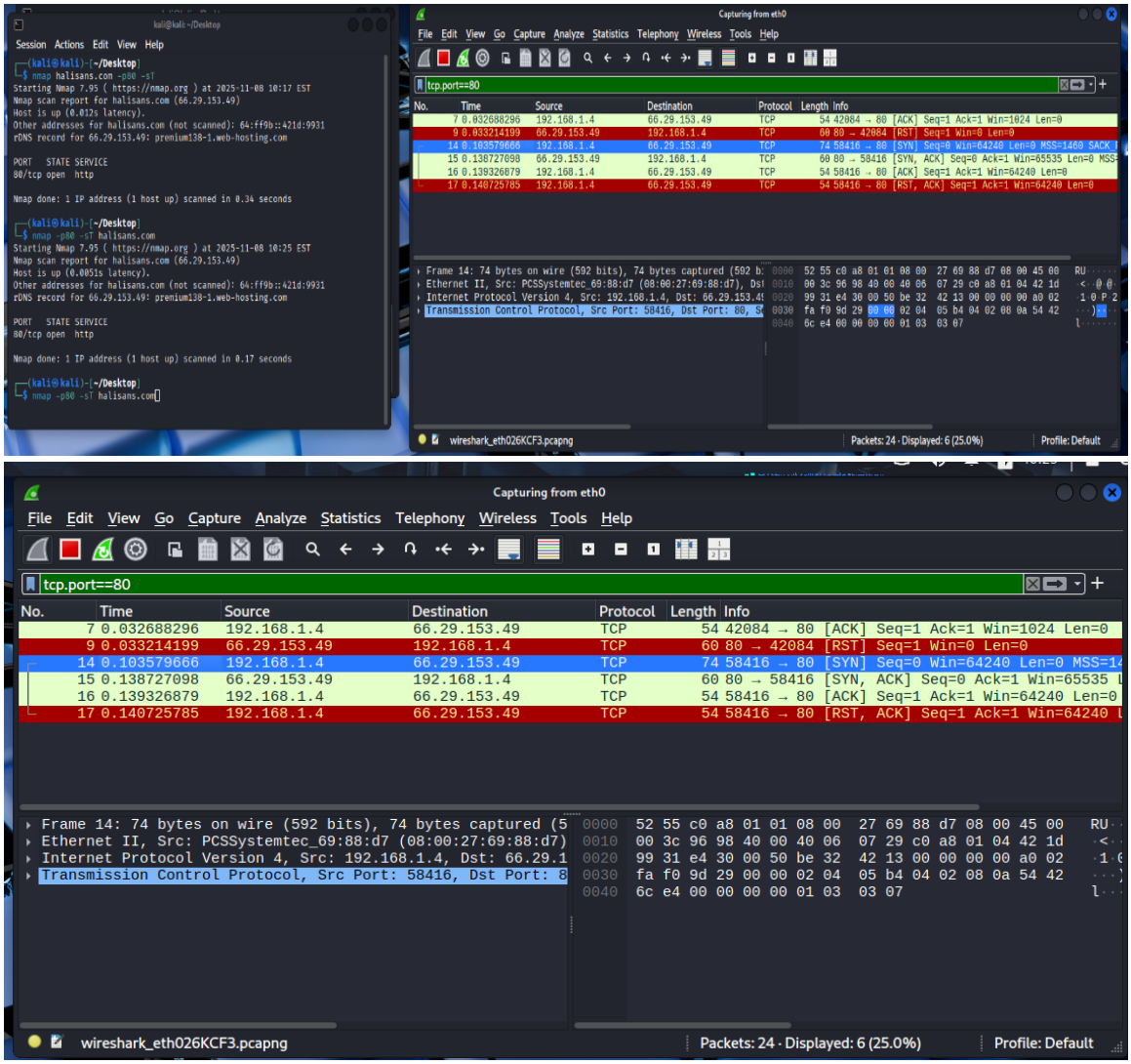
•Capture and identify the complete TCP 3-way handshake (SYN, SYN-ACK, ACK) between 192.168.1.4 and 66.39.153.12 on port 80.
•Demonstrate packet filtering in Wireshark by IP and TCP port.
•Explain stealth scan variants and show how they can evade detection.
• Provide practical detection and mitigation recommendations.

Environment & Capture Details

Environment: Local lab (VM host: kali). Wireshark version: (placeholder)—capture saved as capture_wireshark.pcapng.
Host IP: 192.168.1.4
Destination IP: 66.39.153.12
Target service: HTTP (TCP port 80)
Capture method: Promiscuous mode on the host interface; capture start/end timestamps are left as placeholders.

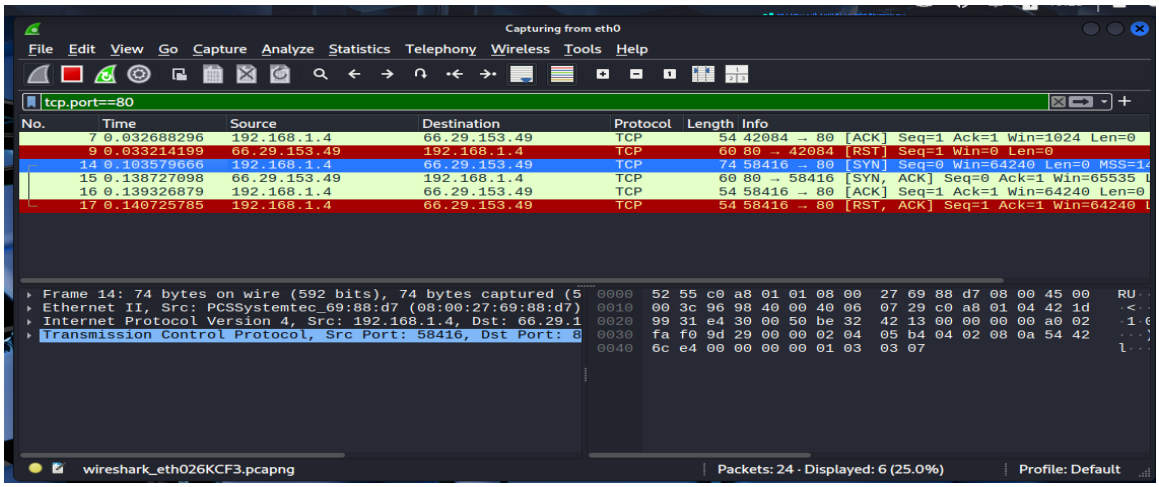
Identifying the 3-Way Handshake

The TCP 3-way handshake can be identified by locating three packets in sequence:
1. SYN — Client (192.168.1.4) sends TCP segment with SYN bit set (tcp.flags.syn==1, tcp.flags.ack==0).
2. SYN-ACK — Server (66.39.153.12) replies with SYN+ACK (tcp.flags.syn==1, tcp.flags.ack==1).
3. ACK — Client (192.168.1.4) 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 66.39.153.12. To isolate scan traffic, the following Wireshark filters were used: Filter by source host and port ip.addr==192.168.1.4 and port tcp.port == 80



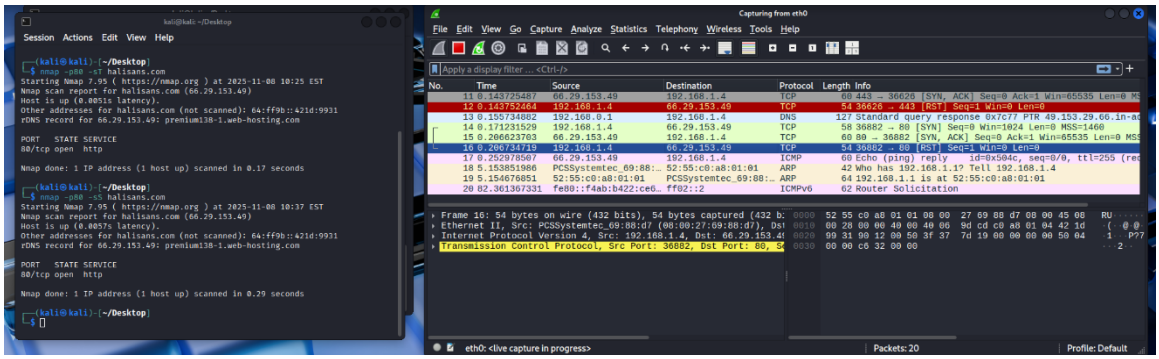
Evasion Techniques Observed / Discussed

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).

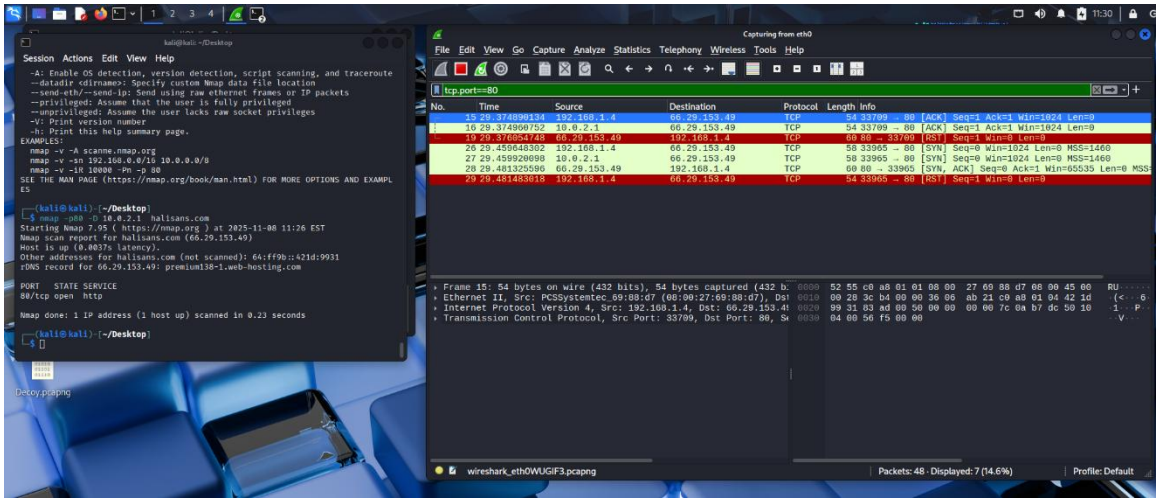


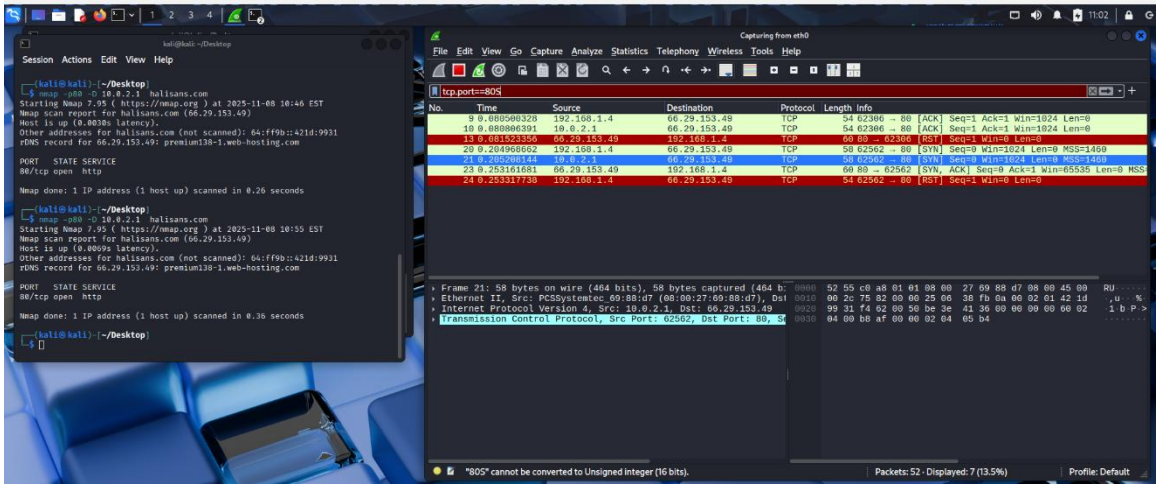
2) Decoy Scans

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.



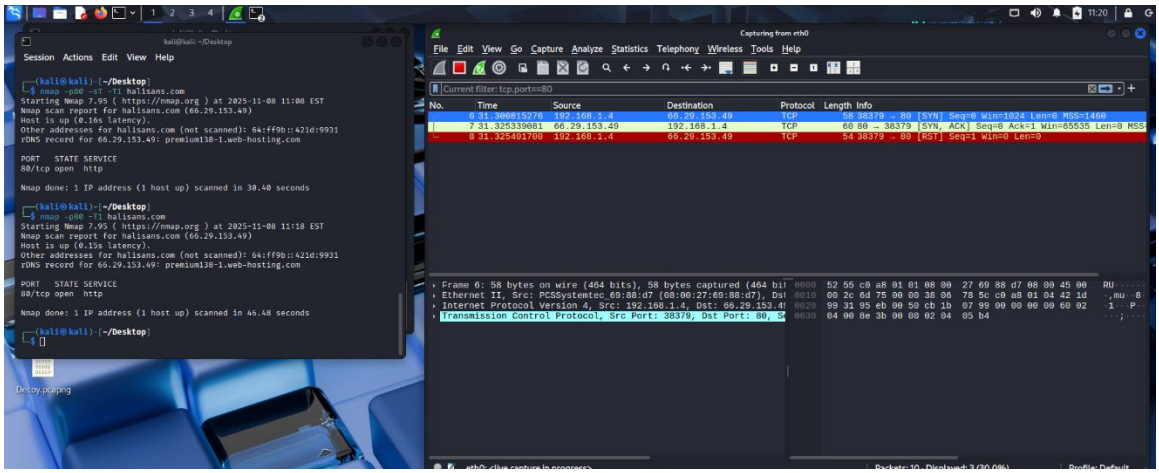


3) Time Fragmentation / Fragmented Scans

Behavior: Attackers fragment packets or spread probe payloads across multiple small fragments and/or time the fragments to arrive slowly. Fragments may evade signature-based detection that inspects single packets.

Why it can bypass detection: If IDS lacks full IP fragment reassembly or has limits on reassembly buffers/timeouts, the signature won't match. Time-based fragmentation spaces probe to avoid threshold-triggered alarms.

Detection strategies: Enable full IP reassembly in IDS/Wireshark, tune reassembly timeouts, monitor unusual fragmentation patterns, and correlate with flow/session metrics.

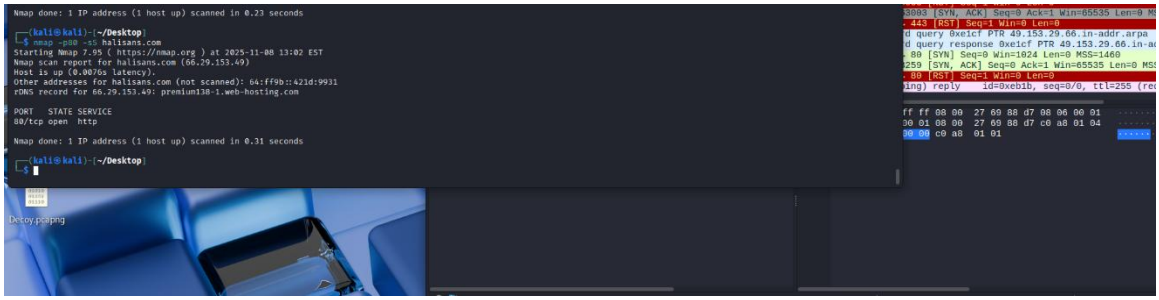


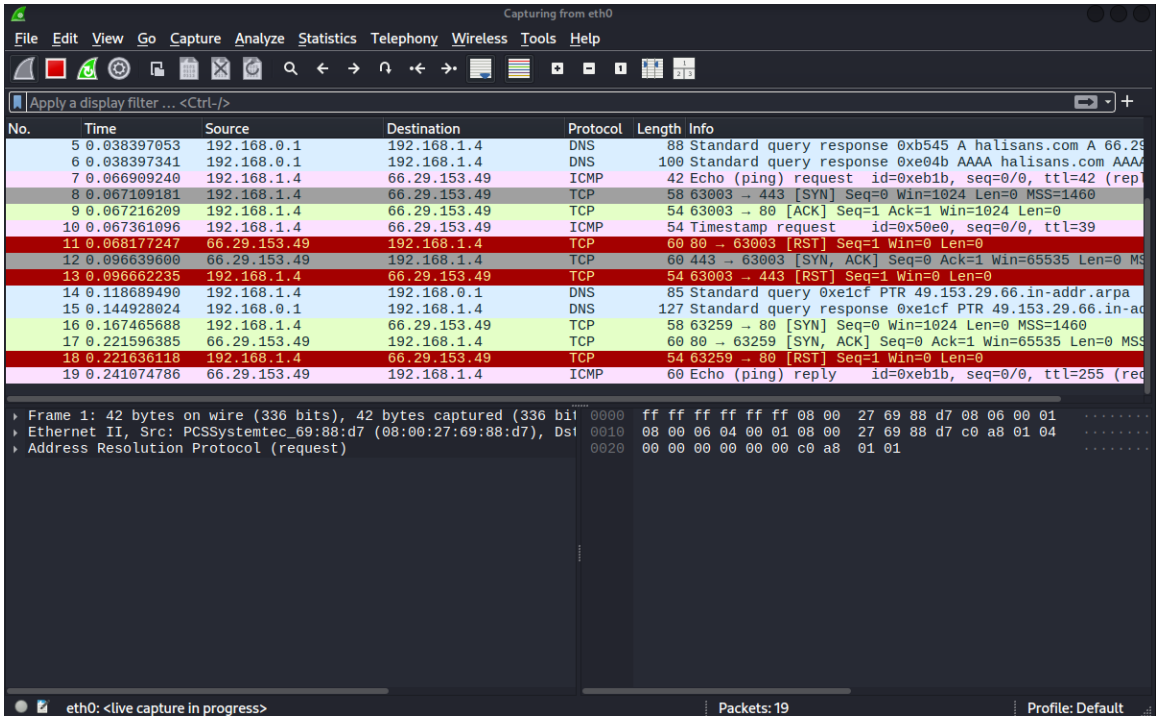
Detection & Mitigation Recommendations

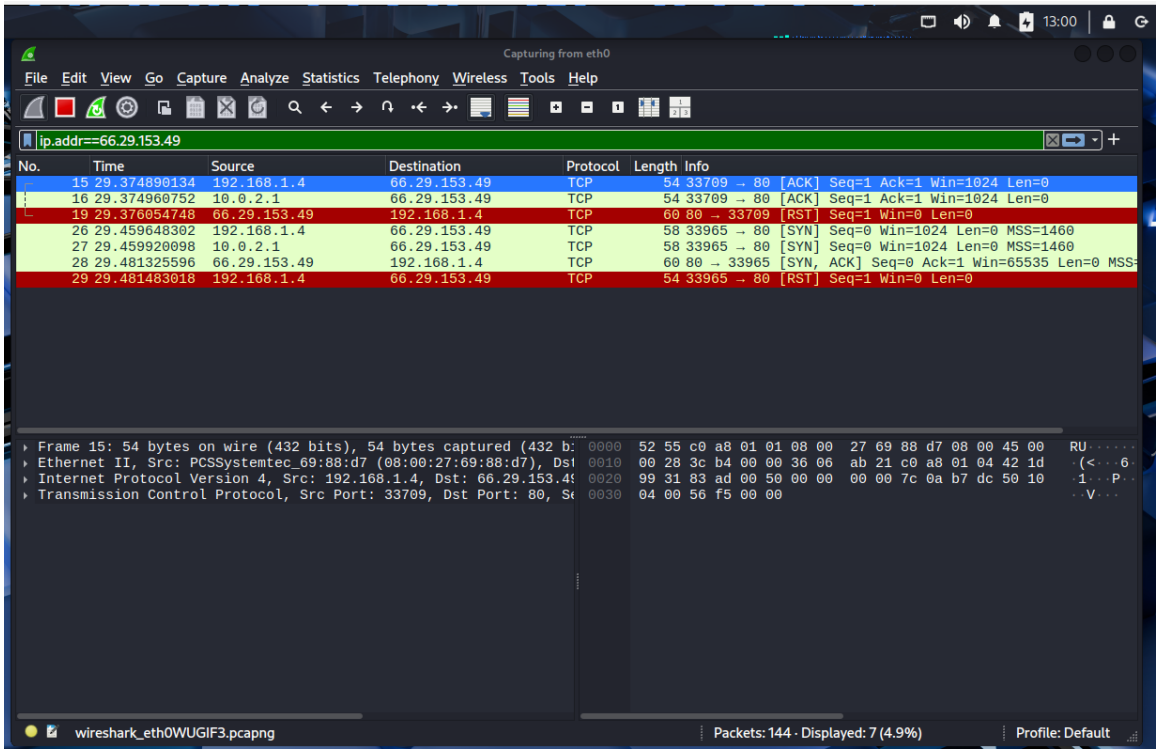
1. Use stateful network devices and enable connection tracking—this helps detect incomplete handshakes.
2. Enable IP fragment reassembly in IDS/IPS (and ensure adequate buffers/timeouts).
3. Correlate network flow telemetry (NetFlow/sFlow) with packet captures to detect distributed low-rate scans.
4. 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 192.168.1.4 and 66.39.153.12 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.




Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
5 0.038397053 192.168.0.1 192.168.1.4 DNS 88 Standard query response 0xb545 A halisans.com A 66.29
6 0.038397341 192.168.0.1 192.168.1.4 DNS 100 Standard query response 0xe04b AAAA halisans.com AAAA
7 0.066909240 192.168.1.4 66.29.153.49 ICMP 42 Echo (ping) request id=0xeb1b, seq=0/0, ttl=42 (rep
8 0.067109181 192.168.1.4 66.29.153.49 TCP 58 63003 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9 0.067216209 192.168.1.4 66.29.153.49 TCP 54 63003 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
10 0.067361096 192.168.1.4 66.29.153.49 ICMP 54 Timestamp request id=0x50e0, seq=0/0, ttl=39
11 0.068177247 66.29.153.49 192.168.1.4 TCP 60 80 → 63003 [RST] Seq=1 Win=0 Len=0
12 0.096639600 66.29.153.49 192.168.1.4 TCP 60 443 → 63003 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MS
13 0.096662235 192.168.1.4 66.29.153.49 TCP 54 63003 → 443 [RST] Seq=1 Win=0 Len=0
14 0.118689490 192.168.1.4 192.168.0.1 DNS 85 Standard query 0xe1cf PTR 49.153.29.66.in-addr.arpa
15 0.144928024 192.168.0.1 192.168.1.4 DNS 127 Standard query response 0xe1cf PTR 49.153.29.66.in-ad
16 0.167465688 192.168.1.4 66.29.153.49 TCP 58 63259 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
17 0.221596385 66.29.153.49 192.168.1.4 TCP 60 80 → 63259 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS
18 0.221636118 192.168.1.4 66.29.153.49 TCP 54 63259 → 80 [RST] Seq=1 Win=0 Len=0
19 0.241074786 66.29.153.49 192.168.1.4 ICMP 60 Echo (ping) reply id=0xeb1b, seq=0/0, ttl=255 (req
Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bi
Ethernet II, Src: PCSSystemtec_69:88:d7 (08:00:27:69:88:d7), Dst
Address Resolution Protocol (request)
0000 ff ff ff ff ff ff 08 00 27 69 88 d7 08 06 00 01
0010 08 00 06 04 00 01 08 00 27 69 88 d7 c0 a8 01 04
0020 00 00 00 00 00 00 c0 a8 01 01
eth0: <live capture in progress>
Packets: 19
Profile: Default


13:00
Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr==66.29.153.49
No. Time Source Destination Protocol Length Info
15 29.374890134 192.168.1.4 66.29.153.49 TCP 54 33709 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
16 29.374960752 10.0.2.1 66.29.153.49 TCP 54 33709 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
19 29.376054748 66.29.153.49 192.168.1.4 TCP 60 80 → 33709 [RST] Seq=1 Win=0 Len=0
26 29.459648302 192.168.1.4 66.29.153.49 TCP 58 33965 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
27 29.459920098 10.0.2.1 66.29.153.49 TCP 58 33965 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
28 29.481325596 66.29.153.49 192.168.1.4 TCP 60 80 → 33965 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS
29 29.481483018 192.168.1.4 66.29.153.49 TCP 54 33965 → 80 [RST] Seq=1 Win=0 Len=0
Frame 15: 54 bytes on wire (432 bits), 54 bytes captured (432 b
Ethernet II, Src: PCSSystemtec_69:88:d7 (08:00:27:69:88:d7), Dst
Internet Protocol Version 4, Src: 192.168.1.4, Dst: 66.29.153.4
Transmission Control Protocol, Src Port: 33709, Dst Port: 80, S
0000 52 55 c0 a8 01 01 08 00 27 69 88 d7 08 00 45 00
0010 00 28 3c b4 00 00 36 06 ab 21 c0 a8 01 04 42 1d
0020 99 31 83 ad 00 50 00 00 00 00 7c 0a b7 dc 50 10
0030 04 00 56 f5 00 00
wireshark_eth0WUGIF3.pcapng
Packets: 144 · Displayed: 7 (4.9%)
Profile: Default