

CyberSecurity Lab4

Soumen Kumar
B22ES006

Task 1:

In this Task, you will capture network traffic to explore how credentials are transmitted over different protocols. You will log in to a local FTP server (10.40.0.103) and an HTTP(http://10.40.0.103) server using the credentials: username test1 and password test1. Using Wireshark, you'll observe how these protocols transmit information in plaintext, making them vulnerable to interception. In contrast, you'll compare this with an HTTPS connection, where encryption protects the login details during transmission. This exercise highlights the importance of using secure protocols like HTTPS and FTPS over their insecure Counterparts.

Solution:

Using FTP Protocol

| ftp | | | | | |
|------|-----------|--------------|--------------|----------|---|
| No. | Time | Source | Destination | Protocol | Length Info |
| 876 | 16.778548 | 10.40.0.103 | 10.23.24.150 | FTP | 74 Response: 220 (vsFTPd 3.0.5) |
| 877 | 16.780954 | 10.23.24.150 | 10.40.0.103 | FTP | 68 Request: OPTS UTF8 ON |
| 879 | 16.781608 | 10.40.0.103 | 10.23.24.150 | FTP | 80 Response: 200 Always in UTF8 mode. |
| 967 | 24.178069 | 10.23.24.150 | 10.40.0.103 | FTP | 66 Request: USER test1 |
| 969 | 24.179105 | 10.40.0.103 | 10.23.24.150 | FTP | 88 Response: 331 Please specify the password. |
| 1202 | 30.129403 | 10.23.24.150 | 10.40.0.103 | FTP | 66 Request: PASS test1 |
| 1209 | 30.167296 | 10.40.0.103 | 10.23.24.150 | FTP | 77 Response: 230 Login successful. |
| 1454 | 45.907976 | 10.23.24.150 | 10.40.0.103 | FTP | 60 Request: QUIT |
| 1455 | 45.908863 | 10.40.0.103 | 10.23.24.150 | FTP | 68 Response: 221 Goodbye. |

In this,if we look at **USER** and **PASS** commands in the captured packets. We can see the username and password transmitted in **plaintext**, making them easily readable

Using HTTP Protocol

| No. | Time | Source | Destination | Protocol | Length Info |
|-----|-----------|---------------|---------------|----------|--|
| 277 | 5.756119 | 10.23.24.150 | 23.223.243.40 | HTTP | 165 GET /connecttest.txt HTTP/1.1 |
| 281 | 5.816804 | 23.223.243.40 | 10.23.24.150 | HTTP | 241 HTTP/1.1 200 OK (text/plain) |
| 403 | 10.455242 | 10.23.24.150 | 10.40.0.103 | HTTP | 577 GET /login_process.html?username=test1&password=test1 HTTP/1.1 |
| 408 | 10.457248 | 10.40.0.103 | 10.23.24.150 | HTTP | 766 HTTP/1.1 200 OK (text/html) |
| 410 | 10.470710 | 10.23.24.150 | 10.40.0.103 | HTTP | 579 GET /success.html HTTP/1.1 |
| 418 | 10.472446 | 10.40.0.103 | 10.23.24.150 | HTTP | 836 HTTP/1.1 200 OK (text/html) |

In this,under the **HTTP request body**, we can see the username and password in **plaintext**, making them vulnerable to interception

Using HTTPS Protocol

| No | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|----------------|----------------|----------|--------|---|
| 337 | 2.595971 | 10.23.24.150 | 8.8.8.8 | TLSv1.2 | 243 | Application Data |
| 338 | 2.595980 | 10.23.24.150 | 8.8.8.8 | TLSv1.2 | 243 | Application Data |
| 340 | 2.596436 | 10.23.24.150 | 8.8.8.8 | TLSv1.2 | 185 | Application Data |
| 341 | 2.596436 | 8.8.8.8 | 10.23.24.150 | TLSv1.2 | 553 | Application Data |
| 342 | 2.596436 | 8.8.8.8 | 10.23.24.150 | TLSv1.2 | 85 | Application Data |
| 344 | 2.596514 | 8.8.8.8 | 10.23.24.150 | TLSv1.2 | 93 | Application Data |
| 345 | 2.596514 | 8.8.8.8 | 10.23.24.150 | TLSv1.2 | 553 | Application Data |
| 347 | 2.596622 | 8.8.8.8 | 10.23.24.150 | TLSv1.2 | 85 | Application Data |
| 349 | 2.597390 | 10.23.24.150 | 8.8.8.8 | TLSv1.2 | 93 | Application Data |
| 351 | 2.598823 | 13.107.42.14 | 10.23.24.150 | TLSv1.2 | 226 | Application Data |
| 352 | 2.600032 | 10.23.24.150 | 52.231.230.148 | TLSv1.2 | 432 | Client Hello (SNI=clarity.ms) |
| 353 | 2.748354 | 52.231.230.148 | 10.23.24.150 | TLSv1.2 | 591 | Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done |
| 372 | 2.749254 | 10.23.24.150 | 52.231.230.148 | TLSv1.2 | 178 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 374 | 2.749737 | 10.23.24.150 | 52.231.230.148 | TLSv1.2 | 533 | Application Data |
| 375 | 2.749889 | 10.23.24.150 | 52.231.230.148 | TLSv1.2 | 524 | Application Data |
| 379 | 2.824250 | 10.23.24.150 | 51.6.207.171 | TLSv1.2 | 400 | Client Hello (SNI=clarity.ms) |
| 380 | 2.864461 | 104.26.5.157 | 10.23.24.150 | TLSv1.3 | 514 | Application Data |
| 384 | 2.864461 | 104.26.5.157 | 10.23.24.150 | TLSv1.3 | 60 | Application Data |
| 385 | 2.866384 | 10.23.24.150 | 104.26.5.157 | TLSv1.2 | 57 | Application Data |
| 388 | 2.893964 | 52.231.230.148 | 10.23.24.150 | TLSv1.2 | 105 | Change Cipher Spec, Encrypted Handshake Message |
| 389 | 2.893964 | 52.231.230.148 | 10.23.24.150 | TLSv1.2 | 123 | Application Data |
| 391 | 2.894200 | 10.23.24.150 | 52.231.230.148 | TLSv1.2 | 82 | Application Data |
| 392 | 2.894637 | 52.231.230.148 | 10.23.24.150 | TLSv1.2 | 443 | Application Data, Application Data |
| 393 | 2.895988 | 10.23.24.150 | 8.8.8.8 | TLSv1.2 | 244 | Application Data |
| 394 | 2.895982 | 10.23.24.150 | 8.8.8.8 | TLSv1.2 | 244 | Application Data |
| 397 | 2.920915 | 8.8.8.8 | 10.23.24.150 | TLSv1.2 | 140 | Application Data |
| 398 | 2.920915 | 8.8.8.8 | 10.23.24.150 | TLSv1.2 | 553 | Application Data |
| 399 | 2.920915 | 8.8.8.8 | 10.23.24.150 | TLSv1.2 | 85 | Application Data |

```

# Frame 373: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits) on interface veth\WF_6C1A6F-CB-36-4E13-86A5-2293754EAB
# Ethernet II, Src: HP-578c312 (04:69:9b:57:9c:12), Dst: Cisco-Ethernet II (cc:bb:dc:ff:03:14)
# Internet Control Message Version 4, Src: 10.23.24.150, Dst: 52.231.230.148
# Transmission Control Protocol, Src Port: 50373, Dst Port: 443, Seq: 1819, Ack: 6378, Len: 93
# Transport Layer Security
0000 cc b6 c8 fe 03 1f 84 69 93 92 57 12 00 00 45 00 0000 i i W 6
0010 00 85 e1 dd 40 00 00 00 00 00 00 17 15 96 34 47 0000 @ . . . . .
0020 00 04 05 c5 43 00 30 69 00 01 0f 11 ff 00 58 13 0000 . . . . . P
0030 00 ff 3c 00 00 16 03 83 00 25 10 00 21 20 00 00 0000 . . . . . X 1
0040 00 c5 3f 3e ae 4c 45 08 2a 2d 68 ff fe 76 27 01 0000 . . . . . h v
0050 14 03 83 00 01 16 03 83 00 28 00 00 00 00 00 0000 . . . . .
0060 00 00 00 01 c5 47 22 41 42 56 4b 2b 29 59 52 3c 7c0000 . . . . . v22
0070 00 2f 9c c5 2a 43 83 5e 06 56 10 07 02 41 00 0000 . . . . . LS + d 2
0080 6a aa 7d

```

Unlike HTTP and FTP, the login credentials here are **encrypted**, we can see encrypted TLS packets

Task 2:

What is TCP handshake delay in Wireshark, and how does it impact network performance by increasing latency and slowing down communication? How can TCP handshake delay be measured using SYN, SYN-ACK, and ACK packets to identify network congestion or server-side issues? What methods can be used to analyze application-level delays, such as HTTP request-response time or DNS query-response time, to diagnose performance bottlenecks and optimize data transmission?

Solution:

The **TCP handshake delay** refers to the time it takes for a client and server to establish a connection. This process follows three steps:

1. **SYN (Client → Server):** The client sends a request to start a connection.
2. **SYN-ACK (Server → Client):** The server acknowledges the request.
3. **ACK (Client → Server):** The client confirms, and the connection is established.

If this handshake takes too long, it increases **network latency**, slowing down communication. High delays can be caused by **network congestion**, **slow server response**, or **inefficient routing**.

| No. | Time | Source | Destination | Protocol | Length | Info |
|--|----------|-----------------|-----------------|----------|--------|---|
| 1494 | 5.553017 | 172.31.24.35 | 142.250.193.227 | QUIC | 1292 | Initial, DCID=abc7329c3d728711, PN: 1, CRYPTO |
| 1495 | 5.553085 | 172.31.24.35 | 142.250.193.227 | QUIC | 1292 | Initial, DCID=abc7329c3d728711, PN: 2, PADDING, PING, PING, CRYPTO, PING, PING, PADDING, PING, PADDING, PING, PADDING, CRYPTO, PING, CRYPTO, CRYPTO, PING, PING |
| 1498 | 5.603801 | 172.31.24.35 | 142.250.193.227 | QUIC | 1292 | Initial, DCID=abc7329c3d728711, PN: 4, CRYPTO |
| 1995 | 6.470312 | 172.31.24.35 | 142.250.193.227 | QUIC | 1292 | Initial, DCID=abc7329c3d728711, PN: 6, PADDING, CRYPTO, CRYPTO, CRYPTO, PADDING, CRYPTO, PADDING, CRYPTO, PADDING, CRYPTO, PADDING, PING, PING |
| 2092 | 7.672326 | 172.31.24.35 | 142.250.193.227 | QUIC | 1292 | Initial, DCID=abc7329c3d728711, PN: 8, CRYPTO |
| 2248 | 8.509754 | 172.31.24.35 | 142.250.193.227 | TCP | 66 | 61655 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM |
| 2249 | 8.509843 | 172.31.24.35 | 142.250.193.227 | TCP | 66 | 61655 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM |
| 2250 | 8.509823 | 172.31.24.35 | 142.250.193.227 | TCP | 66 | 61654 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM |
| 2251 | 8.509467 | 172.31.24.35 | 142.250.193.227 | TCP | 66 | 61655 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM |
| 2252 | 8.509467 | 172.31.24.35 | 142.250.193.227 | TCP | 66 | 61656 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM |
| 2253 | 8.509572 | 172.31.24.35 | 142.250.193.227 | TCP | 66 | 61655 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM |
| 2263 | 8.574616 | 142.250.193.227 | 172.31.24.35 | TCP | 66 | 443 → 61656 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256 |
| 2264 | 8.574616 | 142.250.193.227 | 172.31.24.35 | TCP | 66 | 443 → 61653 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256 |
| 2265 | 8.574616 | 142.250.193.227 | 172.31.24.35 | TCP | 66 | 443 → 61657 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256 |
| 2266 | 8.574616 | 142.250.193.227 | 172.31.24.35 | TCP | 66 | 443 → 61654 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256 |
| 2267 | 8.574739 | 172.31.24.35 | 142.250.193.227 | TCP | 54 | 61656 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=0 |
| 2268 | 8.574755 | 172.31.24.35 | 142.250.193.227 | TCP | 54 | 61653 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=0 |
| 2269 | 8.574758 | 172.31.24.35 | 142.250.193.227 | TCP | 54 | 61657 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=0 |
| 2270 | 8.574761 | 172.31.24.35 | 142.250.193.227 | TCP | 54 | 61654 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=0 |
| 2271 | 8.574774 | 142.250.193.227 | 172.31.24.35 | TCP | 66 | 443 → 61652 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256 |
| 2273 | 8.574792 | 172.31.24.35 | 142.250.193.227 | TCP | 54 | 61652 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=0 |
| 2274 | 8.575140 | 172.31.24.35 | 142.250.193.227 | TCP | 1466 | 61656 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=1412 [TCP PDU reassembled in 2273] |
| 2275 | 8.575140 | 172.31.24.35 | 142.250.193.227 | TLSv1.3 | 433 | Client Hello (SHA-256, TLSv1.3) |
| 2276 | 8.575323 | 172.31.24.35 | 142.250.193.227 | TCP | 1466 | 61657 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=1412 [TCP PDU reassembled in 2277] |
| 2277 | 8.575323 | 172.31.24.35 | 142.250.193.227 | TLSv1.3 | 401 | Client Hello (SHA-256, TLSv1.3) |
| 2278 | 8.575518 | 172.31.24.35 | 142.250.193.227 | TCP | 1466 | 61653 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=1412 [TCP PDU reassembled in 2279] |
| 2279 | 8.575518 | 172.31.24.35 | 142.250.193.227 | TLSv1.3 | 433 | Client Hello (SHA-256, TLSv1.3) |
| 2280 | 8.575675 | 172.31.24.35 | 142.250.193.227 | TCP | 1466 | 61656 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=1412 [TCP PDU reassembled in 2281] |
| 2281 | 8.575675 | 172.31.24.35 | 142.250.193.227 | TCP | 1466 | 61656 → 443 [ACK] Seq=1 Ack=1 Win=65288 Len=1412 [TCP PDU reassembled in 2281] |
| <div> <div>Frame 2253: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on Interface \Device\NPF_{A7C4B2C3-5140-442A-B5E8-58531EAC12F3}</div> <div> <div>Ethernet II, Src: Intel i43-fd:fe (08:a5:e2:43:fd:fe), Dst: IFT-VMPP-ViD-BA (08:00:00:00:00:00)</div> <div>Internet Protocol Version 4, Src: 172.31.24.35, Dst: 142.250.193.227</div> <div>Transmission Control Protocol, Src Port: 61657, Dst Port: 443, Seq. #: 0, Len: 0</div> </div> </div> <div> <div>0000 00 00 5e 00 01 8a 00 a5 e2 43 fd fe 00 00 45 00</div> <div>0010 00 34 2e 55 40 00 00 06 00 00 ac 1f 18 23 be fa</div> <div>0020 c1 a2 fd 40 01 2b ce fd 04 ff 00 00 00 00 00 02</div> <div>0030 ff ff 15 47 00 00 02 04 05 04 01 03 03 00 01 01</div> <div>0040 04 02</div> </div> | | | | | | |

From above :Handshake Delay can be calculated by subtracting the timestamp of ACK request and SYN request:

Delay=TimeStamp(ACK)-TimeStamp(SYN)
 In my case of a google search,this delay was around 13 ms

Ways to Analyze Application-Level Delays

Analyzing application-level delays involves measuring the time taken for various network interactions, such as HTTP requests, DNS queries, and database responses. For web applications, HTTP request-response time is a key metric; this can be analyzed by filtering for **HTTP traffic**, identifying **GET or POST requests**, and measuring the delay until the **HTTP 200 OK response**. Similarly, **DNS query-response time** can be evaluated by tracking the time difference between a **DNS request and its corresponding response**, which helps identify slow DNS resolution issues. Other methods include using **network performance monitoring tools** like Ping, Traceroute, or browser developer tools to inspect **resource loading times** and detect bottlenecks. By identifying these delays, developers can optimize performance through **caching, compression, load balancing, or server-side optimizations**, ensuring faster and more efficient data transmission

| | | | | | | |
|---|-----------|-----------------|-----------------|------|------|--|
| 6607 | 25.432725 | 172.31.24.35 | 23.57.228.121 | HTTP | 454 | GET /r/fwFwZ8HfWu5tA73gUgDgCgUAB8Rr2bWARTWtEYyaspRAZgQfHgQQUgr+rPZFOn8VwB13rKz7z2tUk1V8BCEDAv13udH0qNzF1N28RqqsMS3D HTTP/1.1 |
| 6608 | 25.432946 | 23.57.228.121 | 172.31.24.35 | HTTP | 413 | HTTP/1.1 304 Not Modified |
| 6610 | 25.466484 | 172.31.24.35 | 23.57.228.121 | HTTP | 455 | GET /r/fwFwZ8HfWu5tA73gUgDgCgUAB8Rr2bWARTWtEYyaspRAZgQfHgQQUgr+rPZFOn8VwB13rKz7z2tUk1V8BCEDAv13udH0qNzF1N28RqqsMS3D HTTP/1.1 |
| 6612 | 25.481791 | 23.57.228.121 | 172.31.24.35 | HTTP | 413 | HTTP/1.1 304 Not Modified |
| 6619 | 25.514636 | 172.31.24.35 | 142.250.193.227 | HTTP | 254 | GET /r/r1.cr1 HTTP/1.1 |
| 6622 | 25.520487 | 142.250.193.227 | 172.31.24.35 | HTTP | 277 | HTTP/1.1 304 Not Modified |
| 6628 | 25.558693 | 172.31.24.35 | 23.57.240.163 | HTTP | 281 | GET / HTTP/1.1 |
| 6630 | 25.572818 | 23.57.240.163 | 172.31.24.35 | HTTP | 317 | HTTP/1.1 304 Not Modified |
| 6642 | 25.623293 | 172.31.24.35 | 23.18.239.251 | HTTP | 330 | GET /dglc/rTrusteeKofcSigningS44005SW3A202CAl.cr1 HTTP/1.1 |
| 6754 | 25.969859 | 23.18.239.251 | 172.31.24.35 | HTTP | 1385 | Certificate Revocation List(Dissocier bug, protocol X509AF: C:\gitlab-builds\builds\VSQ3pox2\0\wiresark\wiresark\packet.c:912: failed assertion "saved_layers_len < prefs.gul_max_tree_de- |
| 6762 | 25.951121 | 172.31.24.35 | 23.18.24.67 | HTTP | 335 | GET /msdownload/update/v3/static/trusted/r/authorootstl cab807f3e2683879954 HTTP/1.1 |
| 6764 | 25.969342 | 23.18.24.67 | 172.31.24.35 | HTTP | 321 | HTTP/1.1 304 Not Modified |
| 6765 | 25.979277 | 172.31.24.35 | 142.250.193.227 | HTTP | 256 | GET /r/gsr1.cr1 HTTP/1.1 |
| 6766 | 25.965955 | 142.250.193.227 | 172.31.24.35 | HTTP | 277 | HTTP/1.1 304 Not Modified |
| 6767 | 26.000153 | 172.31.24.35 | 23.18.239.251 | HTTP | 388 | GET /dglc/rstlshwBovES.cr1 HTTP/1.1 |
| 6770 | 26.033283 | 23.18.239.251 | 172.31.24.35 | HTTP | 516 | HTTP/1.1 304 Not Modified |
| 6772 | 26.047666 | 172.31.24.35 | 142.250.193.227 | HTTP | 254 | GET /r/r4.cr1 HTTP/1.1 |
| 6773 | 26.062729 | 142.250.193.227 | 172.31.24.35 | HTTP | 277 | HTTP/1.1 304 Not Modified |
| 6780 | 26.093994 | 172.31.24.35 | 18.66.63.59 | HTTP | 422 | GET /r/fwFwZ8HfWu5tA73gUgDgCgUAB8Rr2bWARTWtEYyaspRAZgQfHgQQUgr+rPZFOn8VwB13rKz7z2tUk1V8BCEDAv13udH0qNzF1N28RqqsMS3D HTTP/1.1 |
| 6782 | 26.107823 | 18.66.63.59 | 172.31.24.35 | HTTP | 584 | HTTP/1.1 304 Not Modified |
| <div> <div>Frame 6601: 413 bytes on wire (3304 bits), 413 bytes captured (3304 bits) on Interface \Device\NPF_{A7C4B2C3-5140-442A-B5E8-58531EAC12F3}</div> <div> <div>Ethernet II, Src: Cisco i803-b2 (08:e5:9b:09:b8:02), Dst: Intel i43-fd:fe (08:a5:e2:43:fd:fe)</div> <div>Internet Protocol Version 4, Src: 23.57.228.121, Dst: 172.31.24.35</div> <div>Transmission Control Protocol, Src Port: 80, Dst Port: 61657, Seq. #: 1, Ack: 481, Len: 359</div> <div>Hypertext Transfer Protocol</div> </div> </div> <div> <div>0000 08 a5 e2 43 fd fe 00 a5 9e b9 03 62 88 00 45 00</div> <div>0010 01 8f cf a2 40 00 f0 06 ee d8 17 39 e4 79 ac 1f</div> <div>0020 18 23 00 56 f0 e3 a1 7c 31 27 54 5e dd 0a 50 10</div> <div>0030 03 f5 6d 15 00 00 4d 54 5a 38 2f 31 2e 31 20 33</div> <div>0040 30 34 20 4e 6f 74 20 4d 6f 64 69 69 65 64 6d</div> </div> | | | | | | |

HTTP Delay= TS(HTTP OK)-TS(GET/POST Req)

For above it comes around 15.6ms

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|-----------|----------------|----------------|----------|--------|--|
| 580 | 3.757643 | 172.31.24.35 | 172.16.100.205 | DNS | 70 | Standard query 0xicd8 A dns.google |
| 581 | 3.757660 | 172.31.24.35 | 172.16.100.205 | DNS | 70 | Standard query 0xbab5 HTTPS dns.google |
| 582 | 3.758590 | 172.31.24.35 | 172.16.100.205 | DNS | 80 | Standard query 0xbed0 A tunnel.googleip.net |
| 583 | 3.761852 | 172.16.100.205 | 172.31.24.35 | DNS | 354 | Standard query response 0xbed0 A tunnel.googleip.net A 216.239.34.157 NS ns3.google.com NS ns2.google.com NS ns1.google.com NS ns4.google.com A 216.239.34.10 A 216.239.32.10 A 216.239.3.. |
| 517 | 3.772792 | 172.16.100.205 | 172.31.24.35 | DNS | 146 | Standard query response 0xbab5 HTTPS dns.google SOA ns1.zdns.google |
| 518 | 3.772792 | 172.16.100.205 | 172.31.24.35 | DNS | 243 | Standard query response 0xicd8 A dns.google A 8.8.4.4 A 8.8.8.8 NS ns3.zdns.google NS ns2.zdns.google NS ns1.zdns.google NS ns4.zdns.google A 216.239.34.114 A 216.239.32.114 A 216.239.36.. |
| 1544 | 5.657318 | 172.31.24.35 | 172.16.100.205 | DNS | 74 | Standard query 0xb97e A www.jfg-nc.com |
| 1545 | 5.657769 | 172.31.24.35 | 172.16.100.205 | DNS | 74 | Standard query 0xca24 HTTPS www.jfg-nc.com |
| 1612 | 5.778075 | 172.31.24.35 | 172.16.100.205 | DNS | 70 | Standard query 0xb9f2 A dns.google |
| 1613 | 5.778384 | 172.31.24.35 | 172.16.100.205 | DNS | 70 | Standard query 0xb931d HTTPS dns.google |
| 1614 | 5.779186 | 172.16.100.205 | 172.31.24.35 | DNS | 243 | Standard query response 0xb9f2 A dns.google A 8.8.4.4 A 8.8.8.8 NS ns1.zdns.google NS ns4.zdns.google NS ns3.zdns.google NS ns2.zdns.google A 216.239.34.114 A 216.239.32.114 A 216.239.36.. |
| 1615 | 5.779186 | 172.16.100.205 | 172.31.24.35 | DNS | 146 | Standard query response 0xb931d HTTPS dns.google SOA ns1.zdns.google |
| 1978 | 6.361975 | 172.16.100.205 | 172.31.24.35 | DNS | 155 | Standard query response 0xca24 HTTPS www.jfg-nc.com CNAME jfg-nc.com SOA ns1.hostatbhd.com |
| 1986 | 6.584187 | 172.16.100.205 | 172.31.24.35 | DNS | 182 | Standard query response 0xb97e A www.jfg-nc.com CNAME jfg-nc.com A 170.249.213.194 NS ns1.hostatbhd.com NS ns2.hostatbhd.com A 170.249.213.194 A 170.249.213.195 |
| 6404 | 22.631783 | 172.31.24.35 | 172.16.100.205 | DNS | 79 | Standard query 0x1d63 A time.cloudflare.com |
| 6405 | 22.639268 | 172.16.100.205 | 172.31.24.35 | DNS | 529 | Standard query response 0x1d63 A time.cloudflare.com A 162.159.200.1 A 162.159.200.123 NS ns7.cloudflare.com NS ns4.cloudflare.com NS ns3.cloudflare.com NS ns6.cloudflare.com NS ns5.clou.. |
| 6681 | 25.388143 | 172.31.24.35 | 172.16.100.205 | DNS | 76 | Standard query 0x3c73 A ocpp.entrust.net |
| 6682 | 25.410277 | 172.31.24.35 | 172.16.100.205 | DNS | 87 | Standard query 0x3c73 A ocpp.entrust.net |
| 6683 | 25.419213 | 172.16.100.3 | 172.31.24.35 | DNS | 491 | Standard query response 0x3c73 A ocpp.entrust.net CNAME ocpp.entrust.net.edgekey.net CNAME 6913.dscx.akamaiedge.net A 23.57.228.121 NS n7dscx.akamaiedge.net NS n2dscx.akamaiedge.net NS .. |
| 6613 | 25.498359 | 172.31.24.35 | 172.16.100.205 | DNS | 70 | Standard query 0x7abc A c.pki.gog |
| 6614 | 25.499222 | 172.16.100.205 | 172.31.24.35 | DNS | 369 | Standard query response 0x7abc A c.pki.gog CNAME pki-gog.l.google.com A 142.250.193.227 NS ns2.google.com NS ns1.google.com NS ns3.google.com NS ns4.google.com A 216.239.34.10 A 216.23.. |
| 6616 | 25.511869 | 172.16.100.205 | 172.31.24.35 | DNS | 491 | Standard query response 0x3c73 A ocpp.entrust.net CNAME ocpp.entrust.net.edgekey.net CNAME 6913.dscx.akamaiedge.net A 23.57.228.121 NS n2dscx.akamaiedge.net NS n3dscx.akamaiedge.net NS .. |
| 6623 | 25.541572 | 172.31.24.35 | 172.16.100.205 | DNS | 74 | Standard query 0xb4b1 A x1.c.lencr.org |
| 6624 | 25.544486 | 172.16.100.205 | 172.31.24.35 | DNS | 583 | Standard query response 0xb4b1 A x1.c.lencr.org CNAME crl.root-x1.letsencrypt.org.edgekey.net CNAME e8652.dscx.akamaiedge.net A 23.57.240.163 NS n4dscx.akamaiedge.net NS n7dscx.akamaiedg.. |
| 6632 | 25.587123 | 172.31.24.35 | 172.16.100.205 | DNS | 77 | Standard query 0x1c14 A cr13.digicert.com |
| 6633 | 25.590178 | 172.16.100.205 | 172.31.24.35 | DNS | 533 | Standard query response 0x1c14 A cr13.digicert.com CNAME cac-ocsp.digicert.com.edgekey.net CNAME e3913.cd.akamaiedge.net A 23.10.239.251 NS n1cd.akamaiedge.net.. |
| 6757 | 25.917886 | 172.31.24.35 | 172.16.100.205 | DNS | 83 | Standard query 0x6c3f A ctld1.windowsupdate.com |
| 6758 | 25.920897 | 172.16.100.205 | 172.31.24.35 | DNS | 545 | Standard query response 0x6c3f A ctld1.windowsupdate.com CNAME ctld1.windowsupdate.com.delivery.microsoft.com CNAME wu-b-net.trafficmanager.net CNAME download.windowsupdate.com.edgesuite.. |
| 6759 | 25.924656 | 172.16.100.205 | 172.31.24.35 | DNS | 545 | Standard query response 0x6c3f A ctld1.windowsupdate.com CNAME ctld1.windowsupdate.com.delivery.microsoft.com CNAME wu-b-net.trafficmanager.net CNAME download.windowsupdate.com.edgesuite.. |
| <pre> > Frame 6682: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0xvc0mp6 [A7C82C3-5140-442A-85E8-58531EAC12F3] > Ethernet II, Src: Intel43-1610 (68:05:22:45:1610), Dst: IFT-vb00-432D_0e (08:00:5e:00:01:0e) > Internet Protocol Version 4, Src: 172.31.24.35, Dst: 172.16.100.3 > User Datagram Protocol, Src Port: 59017, Dst Port: 53 > Domain Name System (Query) 0000 00 00 5e 00 01 0a 00 a5 e2 43 fd f0 80 00 45 00 ..E 0010 00 3c d0 50 08 00 00 11 00 00 c1 12 23 ac 19 00 ..# 0020 64 43 e6 59 00 35 00 2a d4 91 3c 73 01 00 00 01 ..S *cs 0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..o csp entr 0040 75 73 74 03 6e 65 74 80 00 01 00 01 00 01 00 01 ..ust net </pre> | | | | | | |

DNS Delay=TS(response)-TS(query)

For above it comes around 2ms

Diagnosing and Optimizing Performance

If network performance is slow, the following improvements can help:

1. Reducing TCP Handshake Delay

- Use **TCP Fast Open** to speed up handshakes.
- Reduce **packet loss** by ensuring a stable network.

2. Improving HTTP Performance

- Use a **Content Delivery Network (CDN)** to cache content closer to users.
- Enable **server-side caching and compression** to reduce load times.

3. Speeding Up DNS Resolution

- Switch to **faster DNS servers** like **Google DNS (8.8.8.8)**.
- Enable **DNS caching** to store previously resolved addresses.

Task3:

What filters can you apply in Wireshark to separate attack traffic from legitimate traffic, focusing on patterns like high SYN packets? How can you determine the type of DDoS attack (e.g., SYN flood, UDP flood, ICMP flood)? Lastly, how can you identify the attack source and analyze its impact on the network, looking for service degradation?

Solution:

In a DDoS attack, attackers overwhelm a network with excessive traffic, making services slow or unavailable for legitimate users. Using Wireshark, we can separate attack traffic from normal traffic by looking for unusual patterns, such as a high number of SYN packets (indicating a SYN flood) or excessive UDP and ICMP requests (suggesting a UDP or ICMP flood).

To detect a SYN flood, we filter packets where only the SYN flag is set, meaning the attacker is sending connection requests without completing the handshake. A UDP flood can be identified by tracking large volumes of UDP traffic sent to random ports, while an ICMP flood is evident when there are too many ping requests with very few responses.

Once suspicious traffic is found, we can analyze the source IP addresses to check if the attack is coming from a single system or a botnet. Sorting packets by source IP helps identify which addresses are sending excessive traffic. Finally, we assess the impact on the network by looking at packet loss, delayed responses, and failed connections, which indicate that the attack is degrading service quality.

By carefully filtering and analyzing captured traffic, we can not only detect a DDoS attack but also determine its type, source, and impact, helping to mitigate future threat

| tcp.flags.syn == 1 && tcp.flags.ack == 0 | | | | | |
|--|-----------|---------------|-----------------|----------|--|
| No. | Time | Source | Destination | Protocol | Length Info |
| 35 | 70.812282 | 192.168.1.2 | 147.137.21.94 | TCP | 62 2717 → 445 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM |
| 36 | 70.812610 | 192.168.1.2 | 147.137.21.94 | TCP | 62 2718 → 139 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM |
| 37 | 73.731185 | 192.168.1.2 | 147.137.21.94 | TCP | 62 [TCP Retransmission] 2718 → 139 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM |
| 38 | 73.731272 | 192.168.1.2 | 147.137.21.94 | TCP | 62 [TCP Retransmission] 2717 → 445 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM |
| 39 | 79.735812 | 192.168.1.2 | 147.137.21.122 | TCP | 62 2718 → 139 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM |
| 40 | 79.735895 | 192.168.1.2 | 147.137.21.94 | TCP | 62 [TCP Retransmission] 2717 → 445 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM |
| 44 | 94.039026 | 192.168.1.2 | 147.234.1.253 | TCP | 62 2720 → 21 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM |
| 63 | 94.286183 | 147.234.1.170 | 178.178.178.178 | TCP | 113 43698 → 43698 [SYN, PSYN, URG, CWR, Reserved] Seq=0 Win=43698 Urg=43698 Len=39 |
| 83 | 94.289910 | 192.168.1.6 | 147.234.1.253 | TCP | 62 2721 → 58999 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 |

This shows all **SYN packets** without an ACK response, which could indicate an attack.

| No. | Time | Source | Destination | Protocol | Length Info |
|-----|----------|-------------|---------------|----------|-----------------------------------|
| 1 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 2 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 3 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 4 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 5 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 6 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 7 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 8 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 9 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 10 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 11 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 12 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 13 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 14 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 15 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 16 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 17 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 18 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 19 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 20 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 21 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 22 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 23 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 24 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 25 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 26 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 27 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 28 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 29 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 30 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 31 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 32 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 33 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 34 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 35 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 36 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 37 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 38 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 39 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 40 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 41 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 42 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 43 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 44 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 45 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 46 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 47 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 48 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 49 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 50 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 51 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 52 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 53 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 54 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 55 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 56 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 57 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 58 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 59 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 60 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 61 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 62 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 63 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 64 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 65 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 66 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 67 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 68 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 69 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 70 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 71 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 72 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 73 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 74 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 75 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 76 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 77 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 78 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 79 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 80 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 81 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 82 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 83 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 84 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 85 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 86 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 87 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 88 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 89 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 90 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 91 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 92 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 93 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 94 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 95 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 96 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 97 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 98 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 99 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 100 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 101 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 102 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 103 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 104 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 105 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 106 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 107 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 108 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 109 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 110 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 111 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 112 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 113 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 114 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 115 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 116 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 117 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 118 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 119 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 120 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 121 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 122 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 123 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME<ic> |
| 124 | 0.000000 | 192.168.1.2 | 192.168.1.255 | ICMP | 92 Name query NB ECT_DOWNTIME< |

A **ping flood** sends a massive number of ICMP Echo Requests (ping requests) to overload the network.

To check for this, we can use this command: `icmp`

Things to keep in mind

- Look for a **high number of ICMP Echo Requests** (Type 8) without corresponding Echo Replies (Type 0).
- If most traffic comes from a single source, it may be a **direct attack**; if from multiple sources, it suggests a **botnet-based DDoS**.

To find the **IP addresses involved in the attack**:

- We can sort packets by **Source IP** and check which IPs are sending an excessive number of requests.

Using the filter:

```
ip.src == x.x.x.x
```

- (Replace `x.x.x.x` with a suspicious IP) to isolate and analyze its behavior.
- Cross-check against known **DDoS IP databases** to confirm if the IP is part of a botnet.

To determine how the attack affects network performance:

We can check **packet loss** using the filter:

```
Tcp.analysis.lost_segment
```

We can look for **high latency** in TCP handshakes with:

```
tcp.analysis.initial_rtt
```

- If **legitimate user connections** (e.g., HTTP requests) are getting delayed or dropped, this indicates **service degradation** due to the attack.