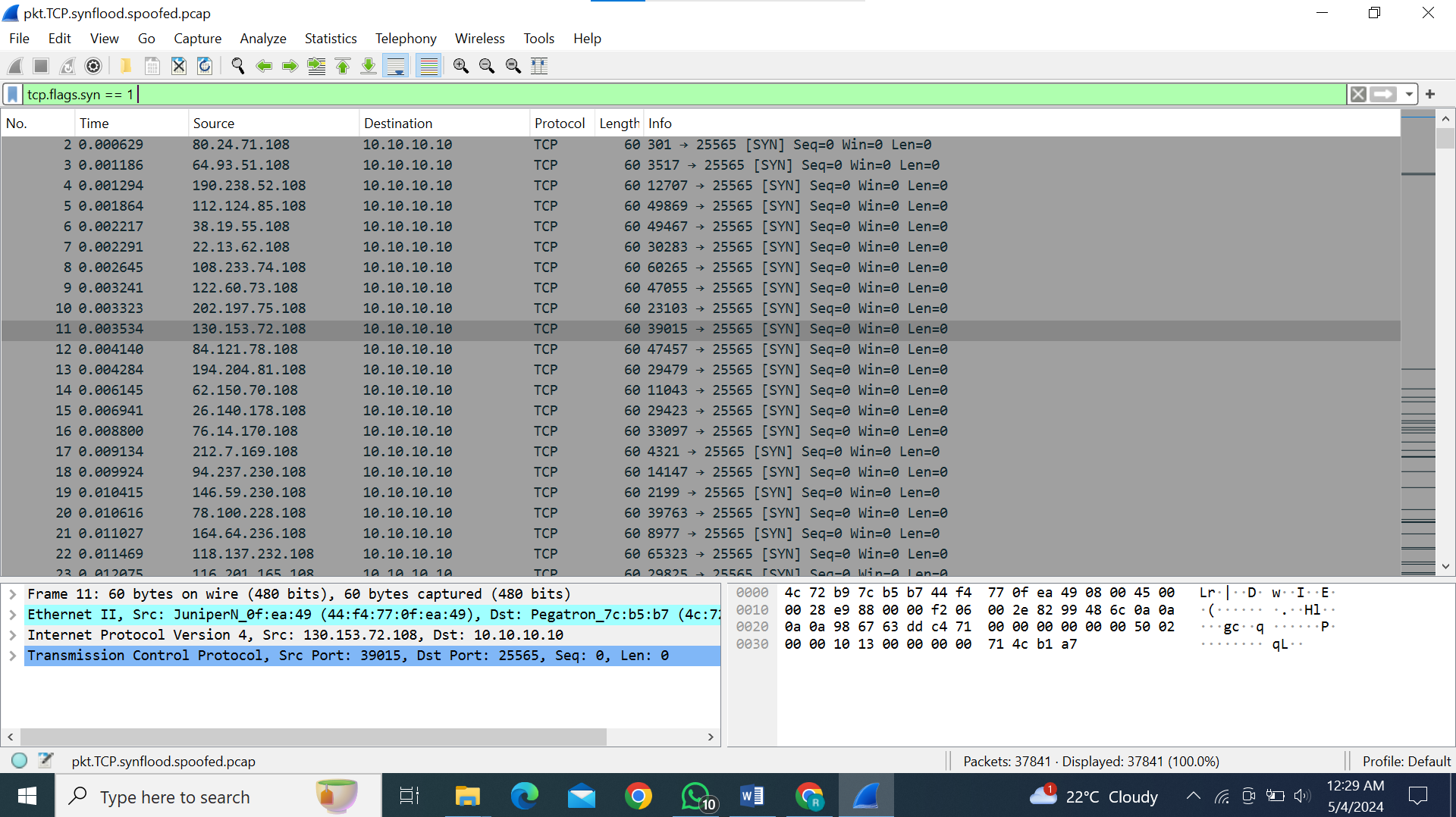
**DETAILED ANALYSIS OF 1ST PCAP FILE:**

After the analysis of the file, I examined the large number of half-open connections, where the handshake process (SYN-ACK) is initiated but not completed.

Then I studied what type of behavior is this and I found this to be **SYN FlOOD ATTACK**.

**The Signature of SYN FLOOD ATTACK:**

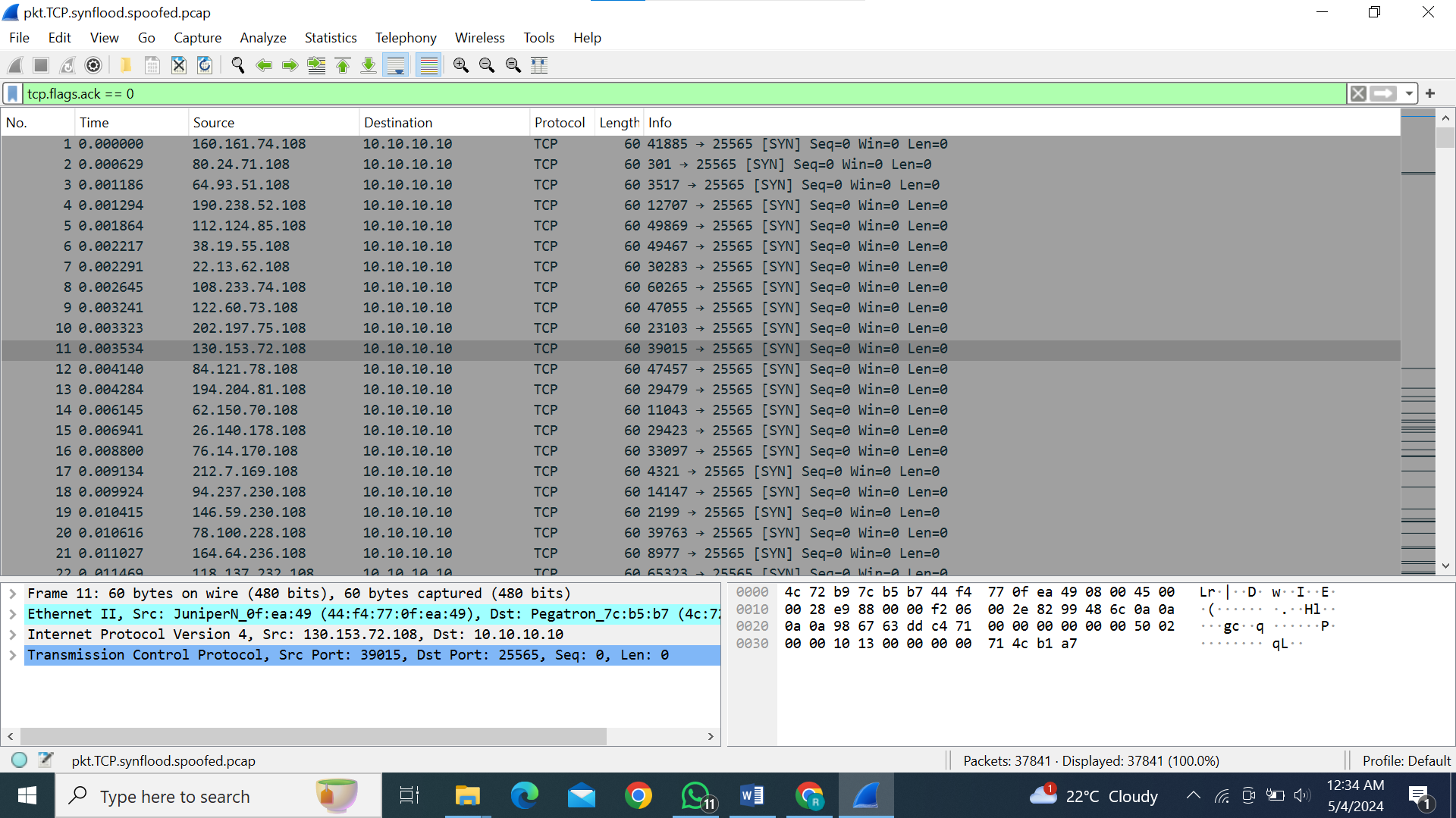
1. **Attack Initiation:** Flood of SYN packets sent to the target server.



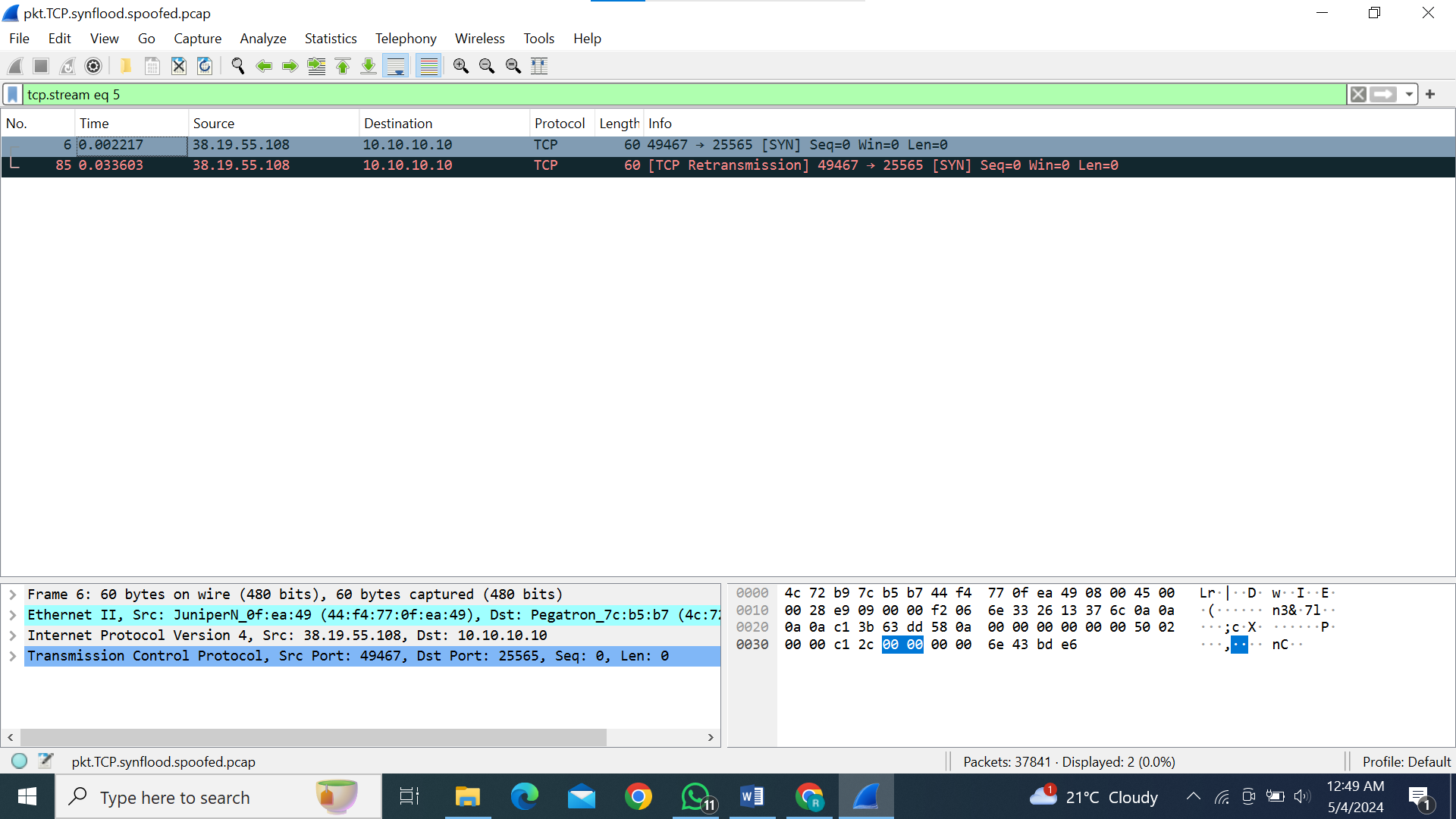
Through the above screenshot, we examine the bundle of SYN packets that are requested for connection.

1. **Half-Open Connections:** The server responds with SYN-ACK but connections remain incomplete.

When the TCP ACK (Acknowledgment) flag is set to zero, it means that the packet is not acknowledging any data.

In a SYN flood attack, the attacker sends numerous SYN packets with the ACK flag set to zero. These packets flood the target server, overwhelming its resources and preventing it from completing the TCP handshake process for legitimate connections.

1. **Retransmission of packets:**  I have found certain retransmission of packets in which still there is no payload in it

****When the length (len) field of a TCP packet is set to 0, and there is nothing in the payload, it typically indicates that the TCP packet is empty. In other words, it doesn't carry any application data.

An empty TCP header, particularly in the context of SYN flood attacks, typically refers to TCP packets with minimal or no payload and potentially spoofed or unusual header fields.

In the context of SYN flood attacks, empty TCP headers might be used to overwhelm a target server with a flood of connection requests, consuming its resources and causing denial of service to legitimate users.

It indicates that it is a DDos attack

**Results:**

* **Resource Depletion**: Server resources are exhausted due to maintaining half-openconnections.
* **Service Disruption**: Legitimate requests delayed or denied due to resource exhaustion.
* **Objective**: Overwhelm the server, causing a denial of service to legitimate users.

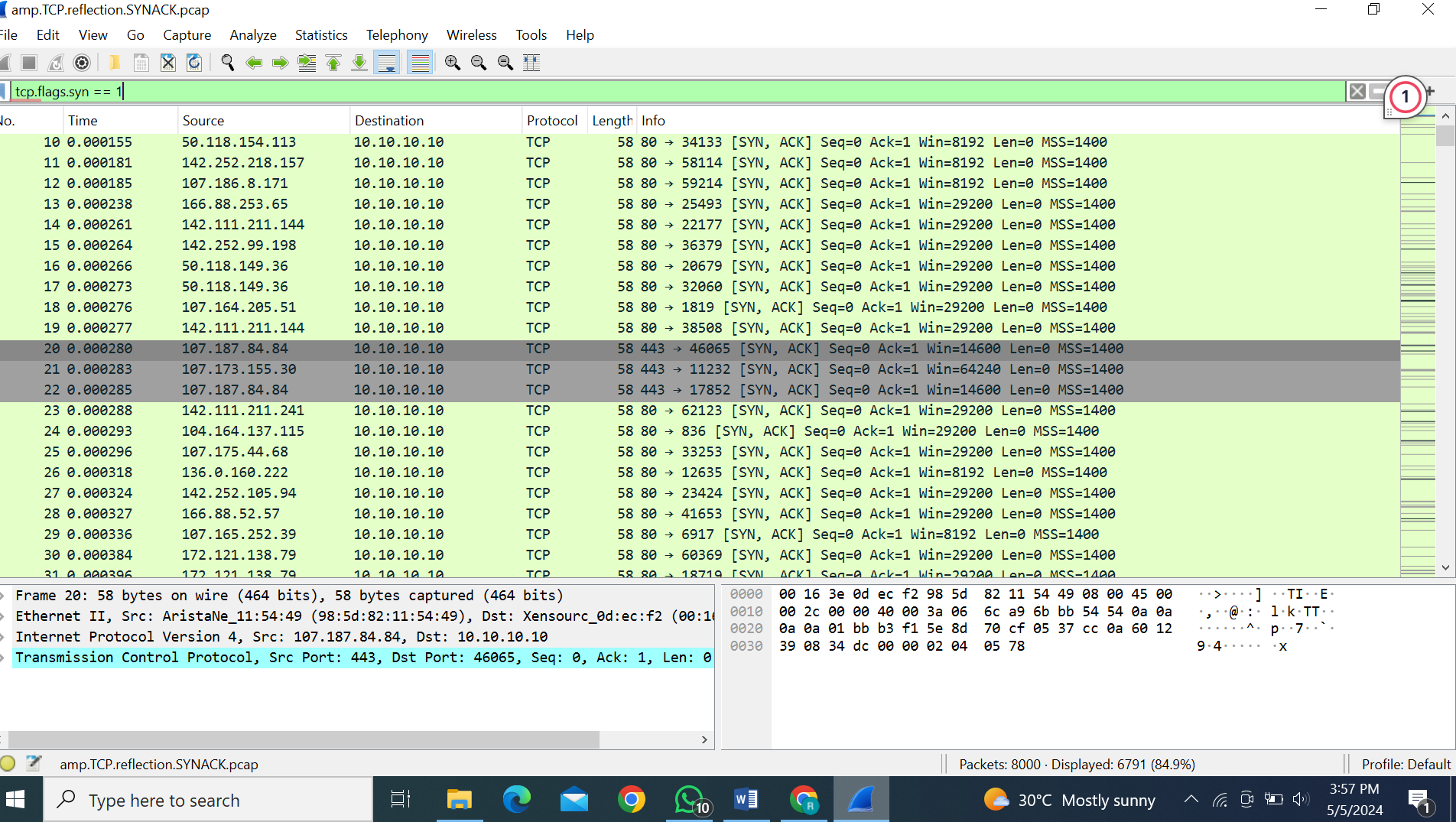
**DETAILED ANALYSIS OF 2nd PCAP FILE:**

After analyzing the file, I observed a significant number of half-open connections, characterized by the initiation of the SYN-ACK handshake process without completion. This pattern aligns with the behavior expected in an SYN-ACK reflection DDoS attack, where spoofed SYN packets trigger servers to respond with SYN-ACK packets, overwhelming the victim with incomplete connections.

Then I studied what type of behavior is this and I found this to be a **SYN-ACK reflection DDoS attack**.

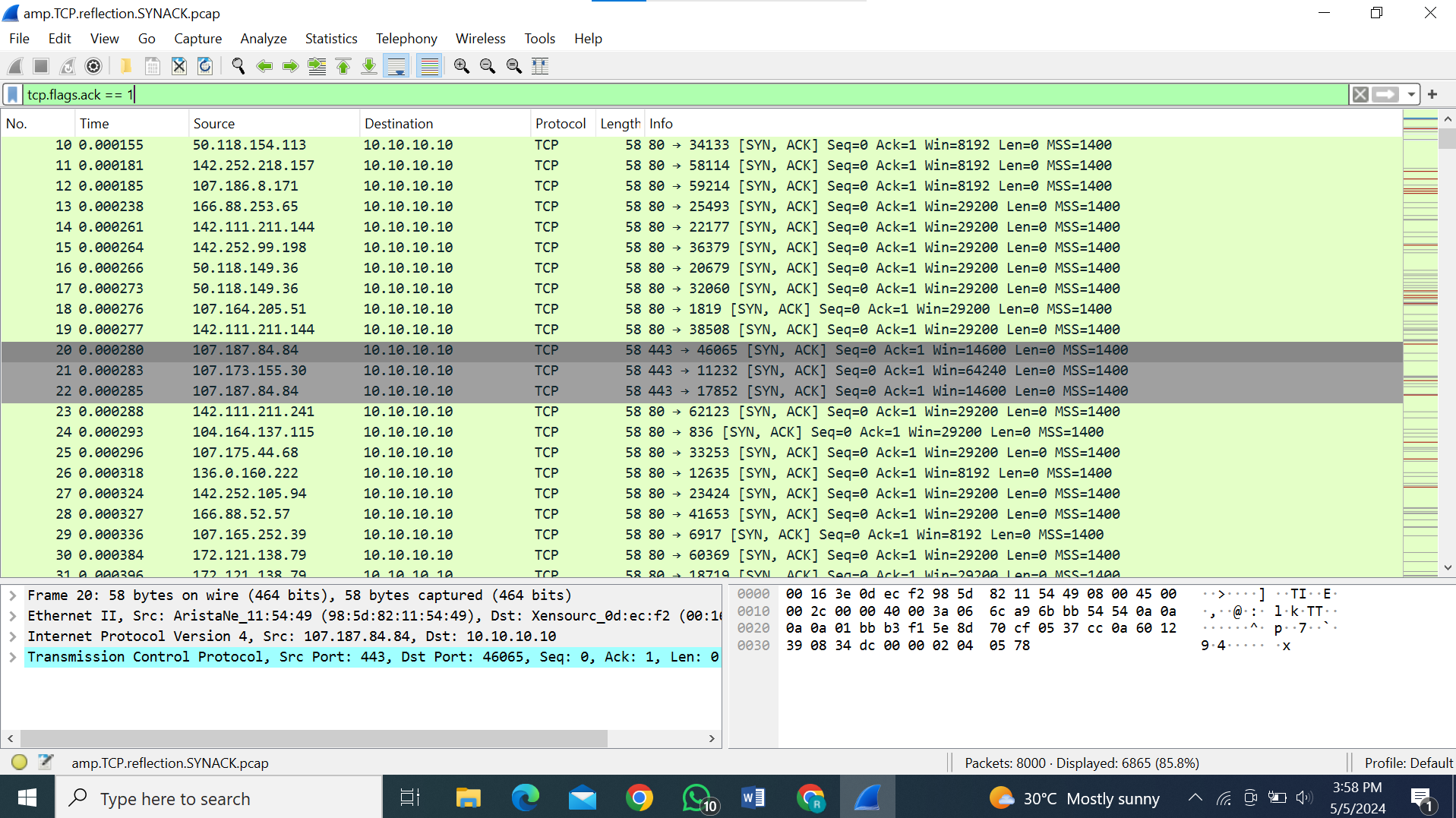
**Attack behavior:**

**Filter: tcp.flags.syn == 1**



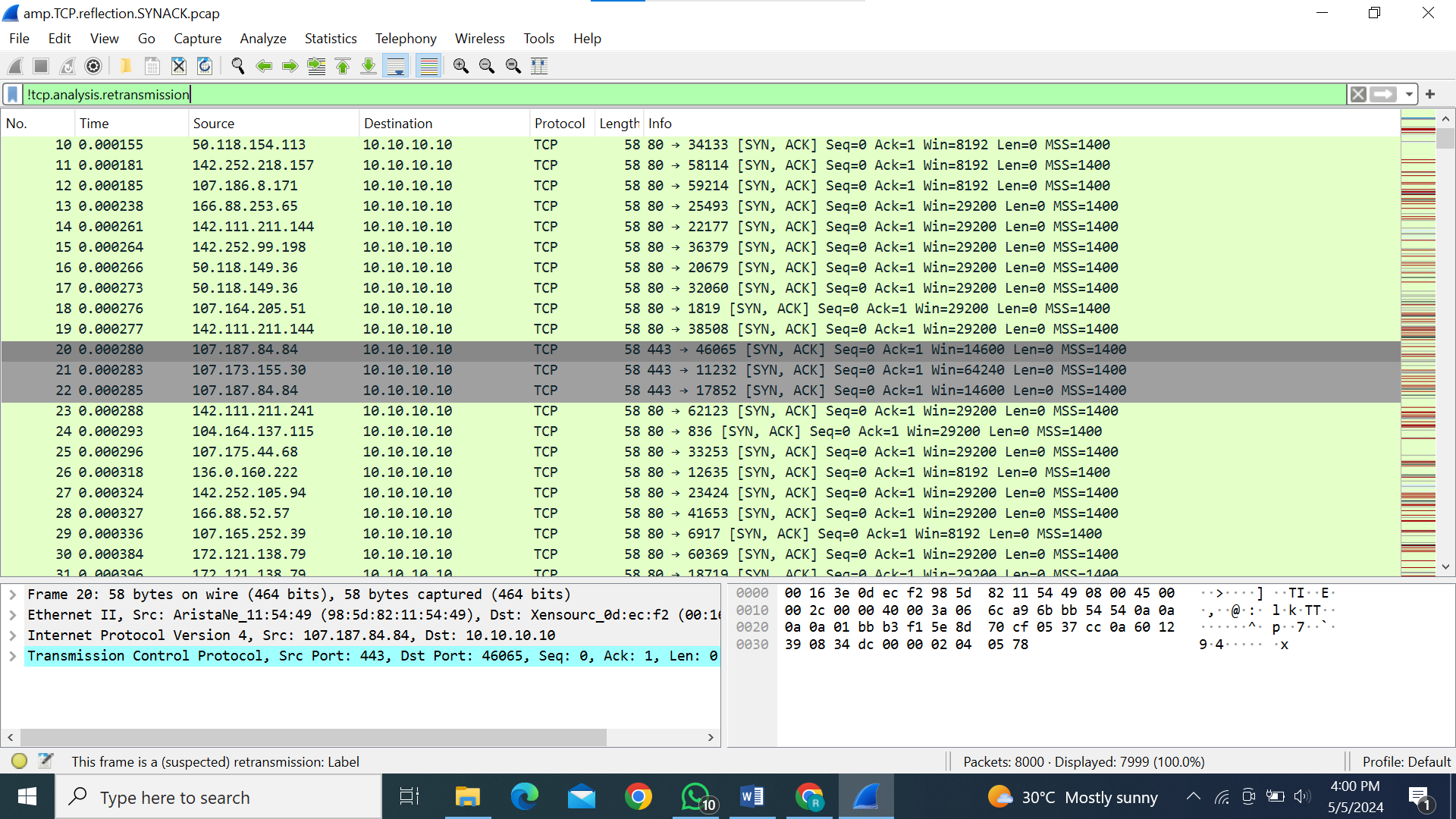
This filter captures packets with the SYN flag set. In a SYN-ACK reflection DDoS attack, attackers send SYN packets with spoofed source IP addresses to multiple servers, initiating the TCP handshake process.

**Filter: tcp.flags.ack == 1**



This filter captures packets with the ACK flag set. In the context of a SYN-ACK reflection DDoS attack, it helps to identify the SYN-ACK packets generated by servers in response to the spoofed SYN packets, as part of completing the TCP handshake.

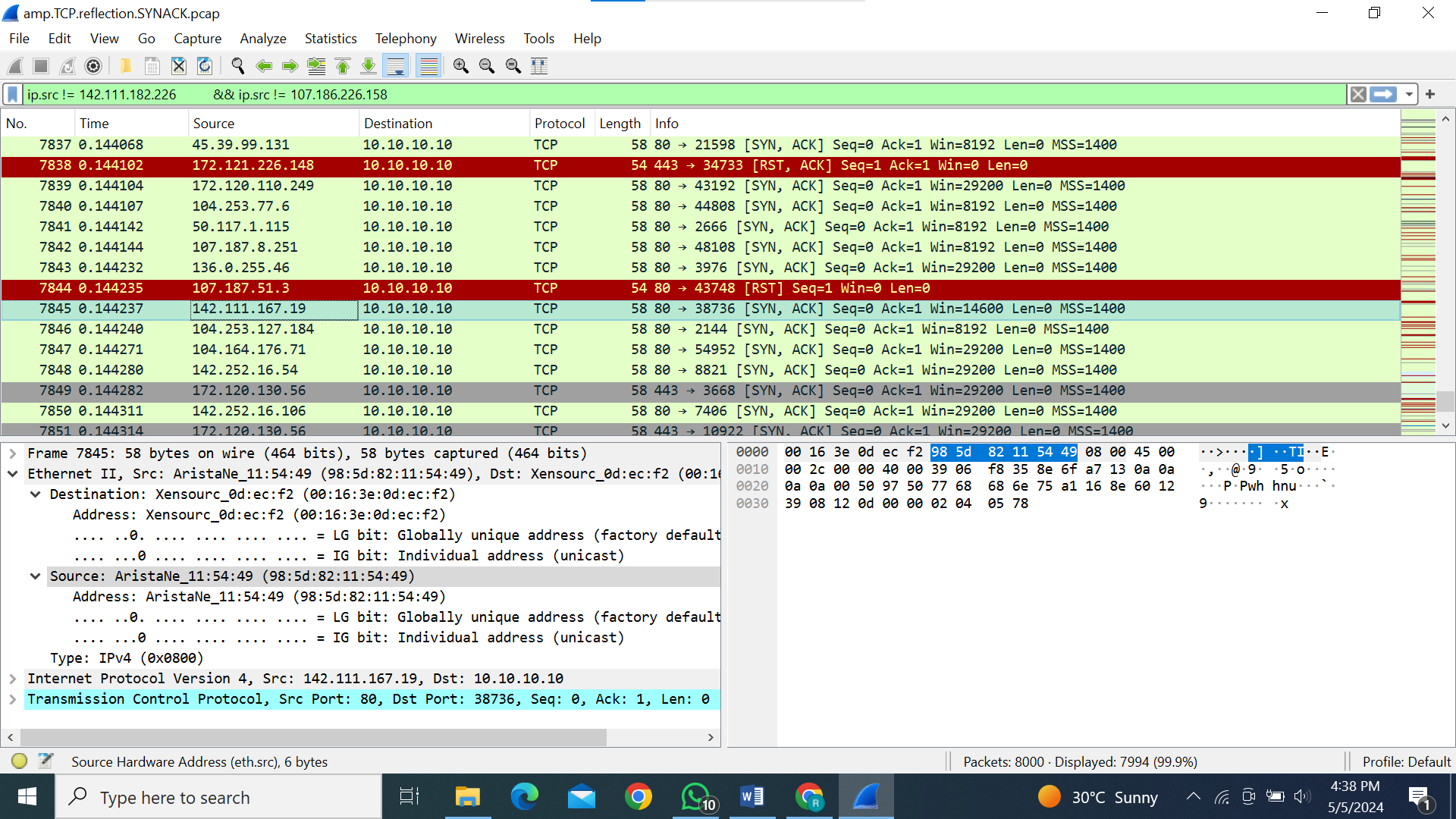
**Filter:! tcp.analysis.retransmission**



This filter excludes packets flagged as retransmissions. In a SYN-ACK reflection DDoS attack, it ensures that only original SYN and SYN-ACK packets are captured, filtering out any retransmitted packets, which may skew the analysis of the attack traffic.

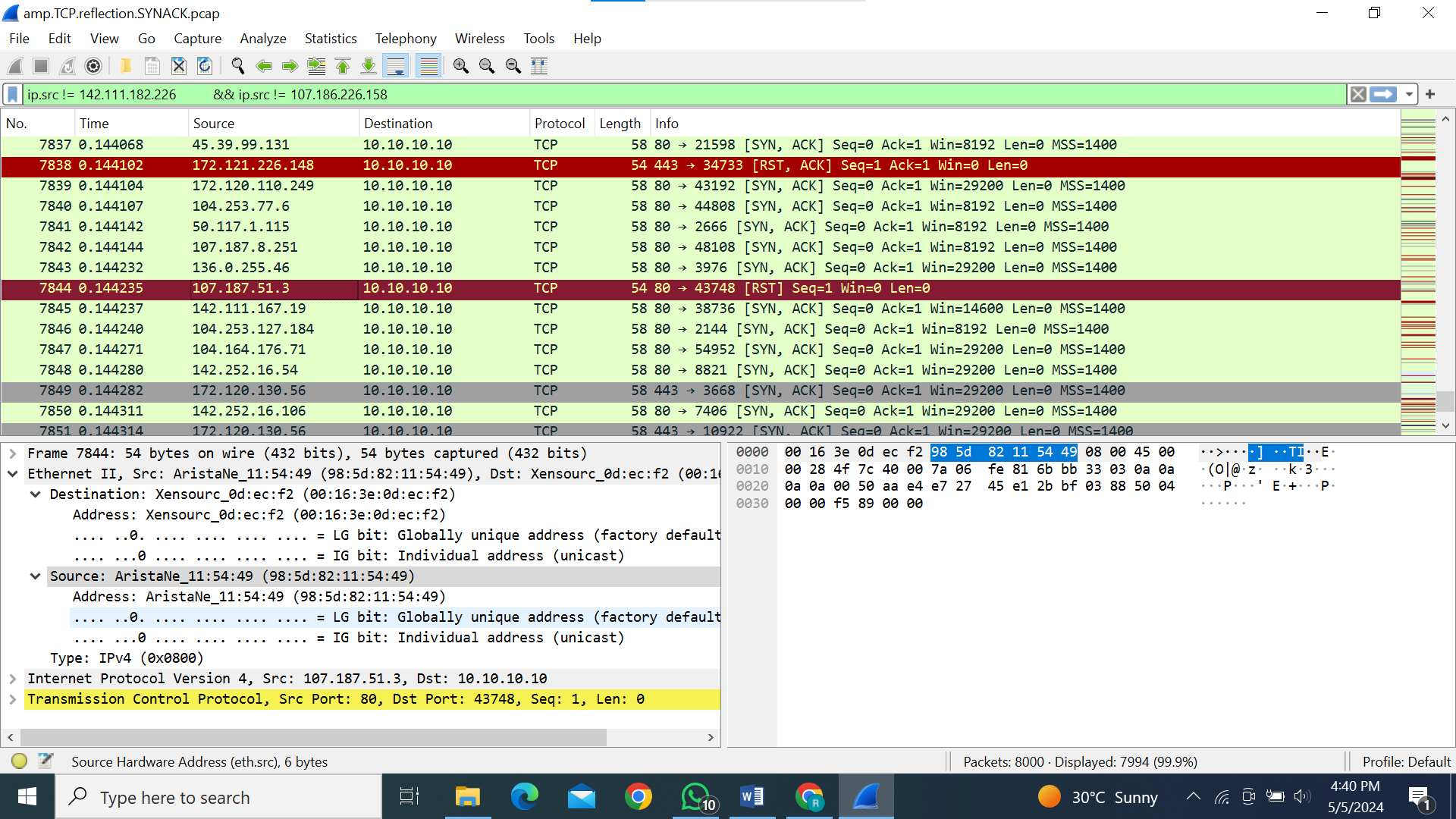
**IP spoofing:**

While examining the packets I noticed that the Mac is not changing while the Ip is changing in every packet, which means that the attacker is spoofing its IP and sending packets which indicates that these packets are not coming from a legitimate source.



In the above picture, we can see that the Mac address is

“**Source: AristaNe\_11:54:49 (98:5d:82:11:54:49)**” and the Ip address is “**142.111.167.19**”



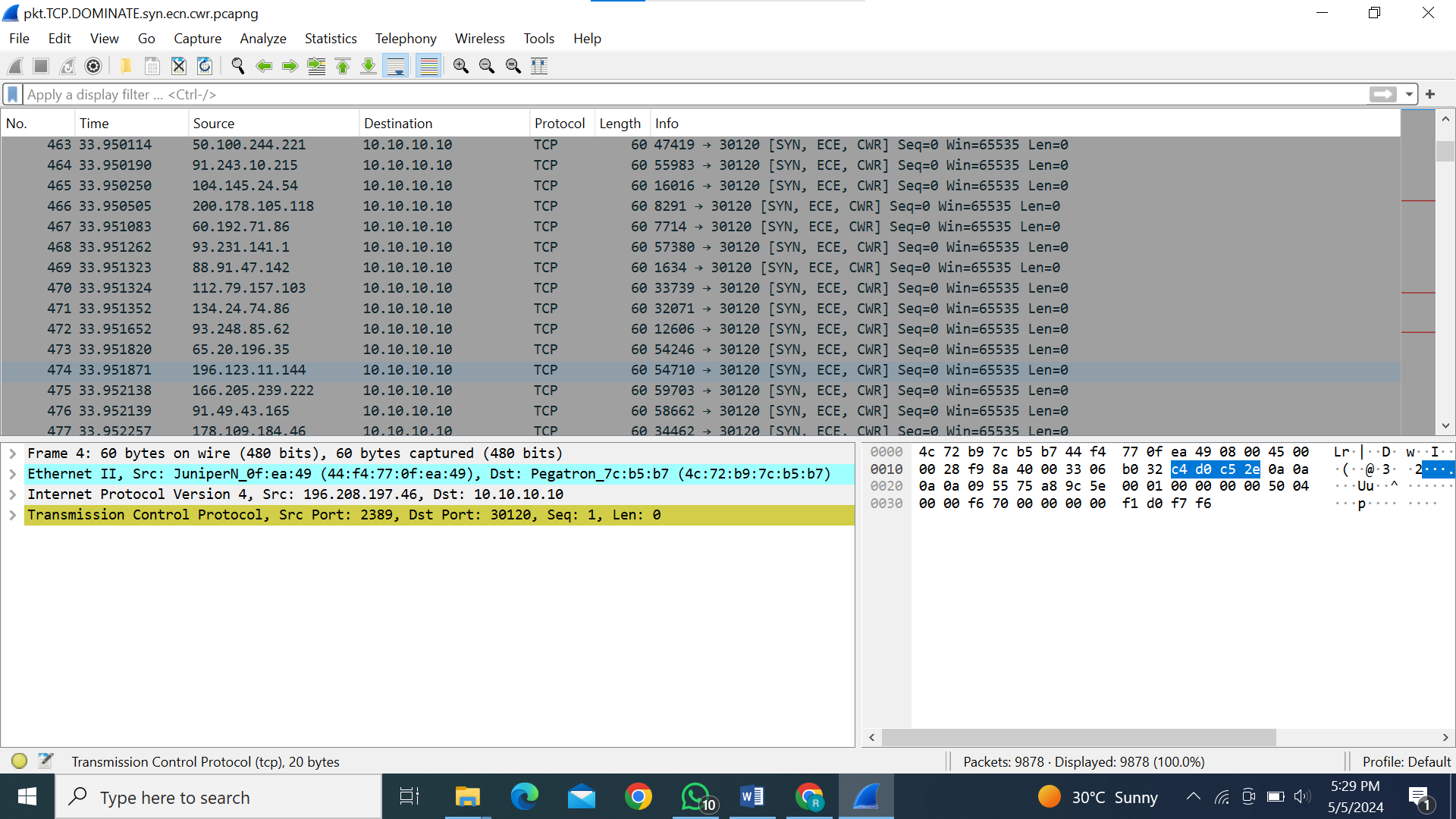
In this photo, we can see that the Mac address remains the same whereas the source IP changes to “**107.107.51.3**”.

It indicates that the IP is spoofed and it’s a **DDoS** attack.

**DETAILED ANALYSIS OF 3rd PCAP FILE:**

In a TCP Reset (RST) flood attack, attackers initiate TCP connections with the victim, setting flags indicating congestion. Then, they flood the victim with TCP RST packets, aiming to disrupt its connections. This flood of RST packets overwhelms the victim's resources, causing denial of service by abruptly terminating legitimate connections.

**Network Congestion:**



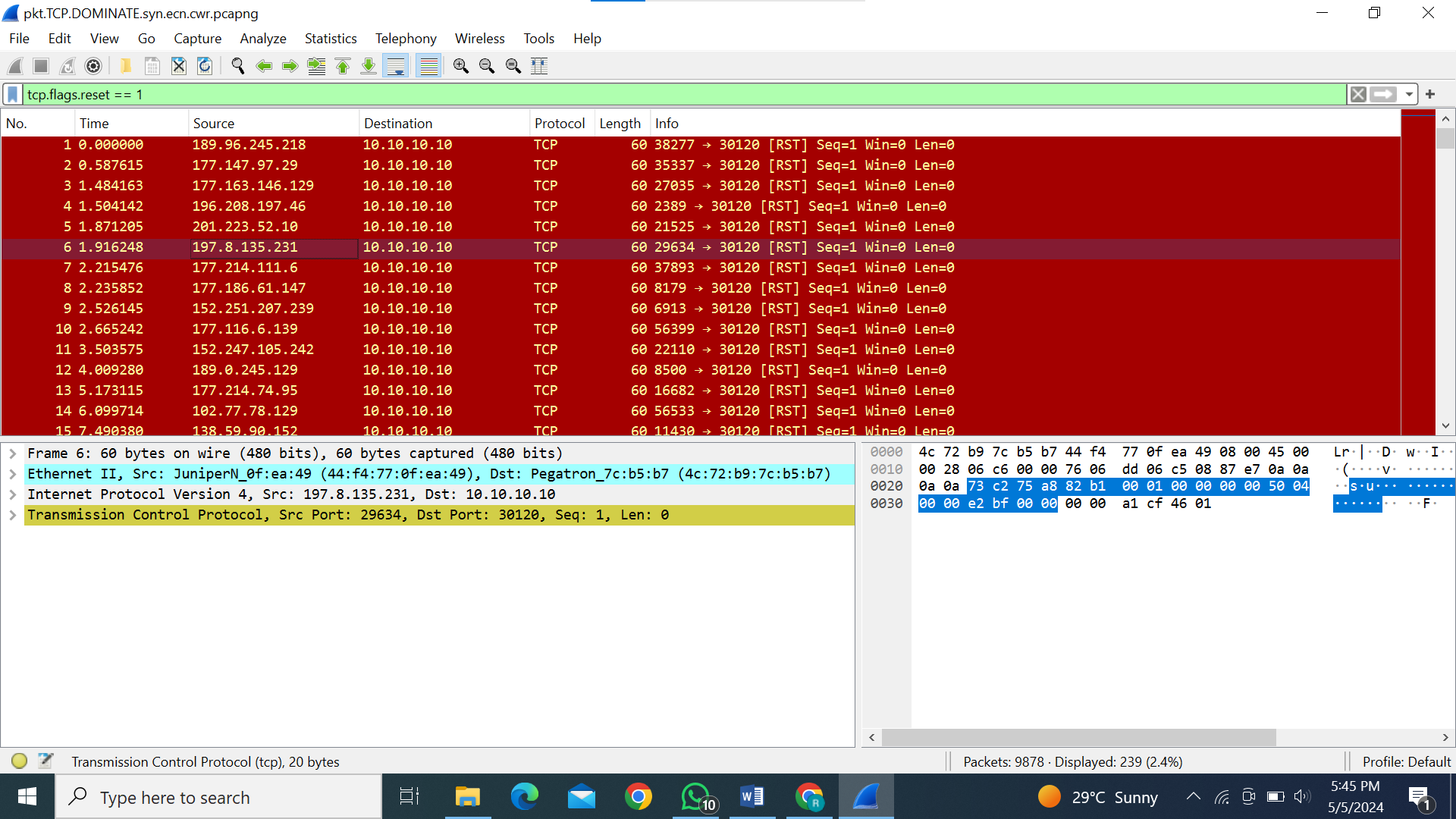
This packet indicates the initiation of a TCP connection

* "SYN" flag: The sender is requesting to establish a TCP connection.
* "ECE" (Explicit Congestion Notification Echo) flag: Indicates that the sender received an ECN (Explicit Congestion Notification) packet with the ECE flag set, suggesting possible network congestion.
* "CWR" (Congestion Window Reduced) flag: Indicates that the sender is reducing its congestion window size in response to congestion indications.

**Reset Flag:**

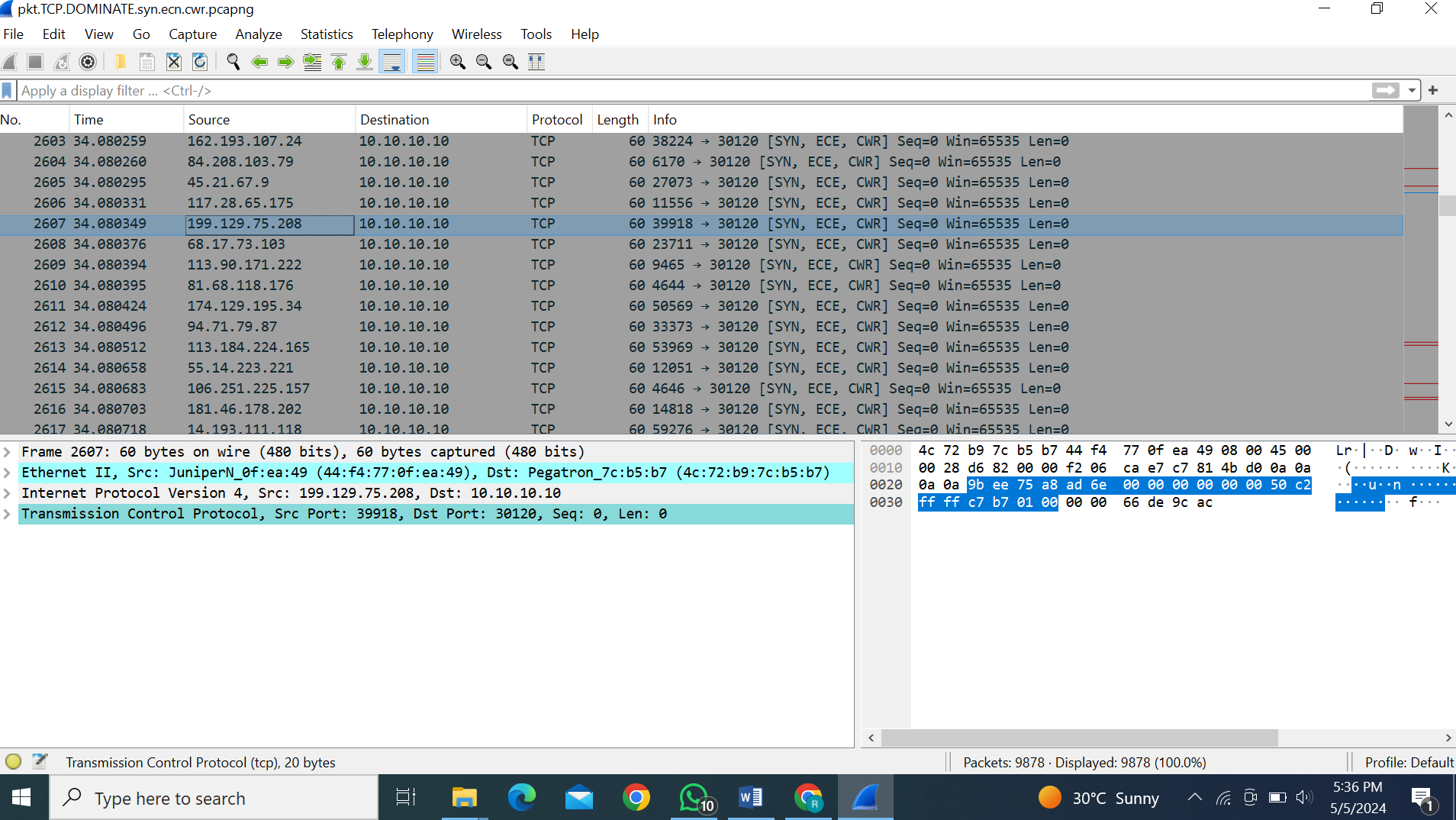
**Filter: tcp.flags.reset == 1**

This filter will display all TCP packets with the RST flag set, allowing you to observe if there is an unusually high volume of RST packets indicative of a flood attack.

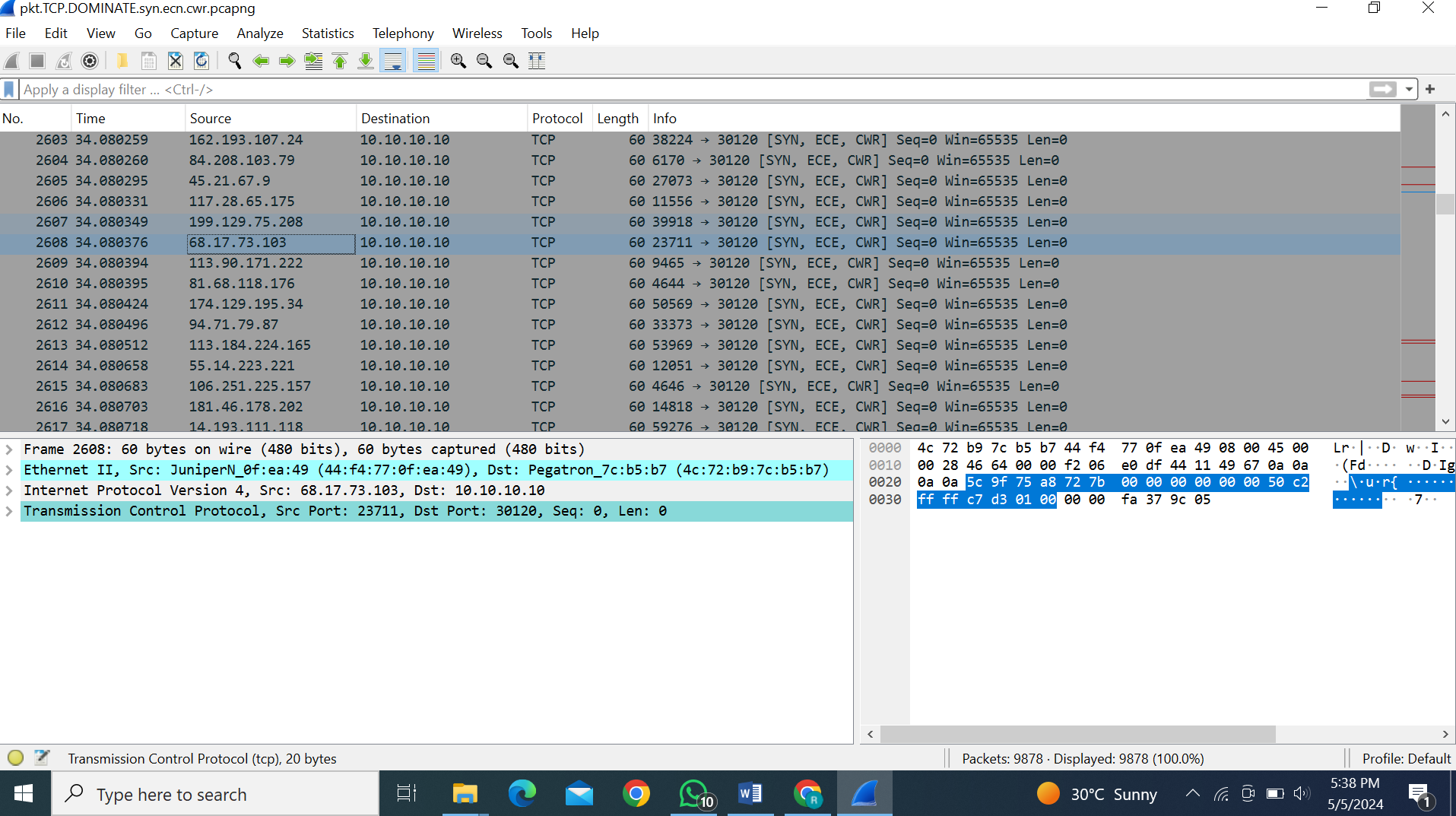


This packet is a TCP RST packet sent from the source IP address 197.8.135.231 to the destination IP address 10.10.10.10. It has the RST (Reset) flag set, which indicates an attempt to abruptly terminate a connection. Adverse effects of such packets include disrupting ongoing communication, leading to potential **denial of service (DoS) by terminating legitimate connections.**

**Spoofed Ip:**



* **MAC address:**
  + Source: JuniperN\_0f:ea:49 (44:f4:77:0f:ea:49)
  + Destination: Pegatron\_7c:b5:b7 (4c:72:b9:7c:b5:b7)
* **IP addresses:**
  + Source: 199.129.75.208
  + Destination: 10.10.10.10



* **MAC addresses:**
  + Source: JuniperN\_0f:ea:49 (44:f4:77:0f:ea:49)
  + Destination: Pegatron\_7c:b5:b7 (4c:72:b9:7c:b5:b7)
* **IP addresses:**
  + Source: 68.17.73.103
  + Destination: 10.10.10.10

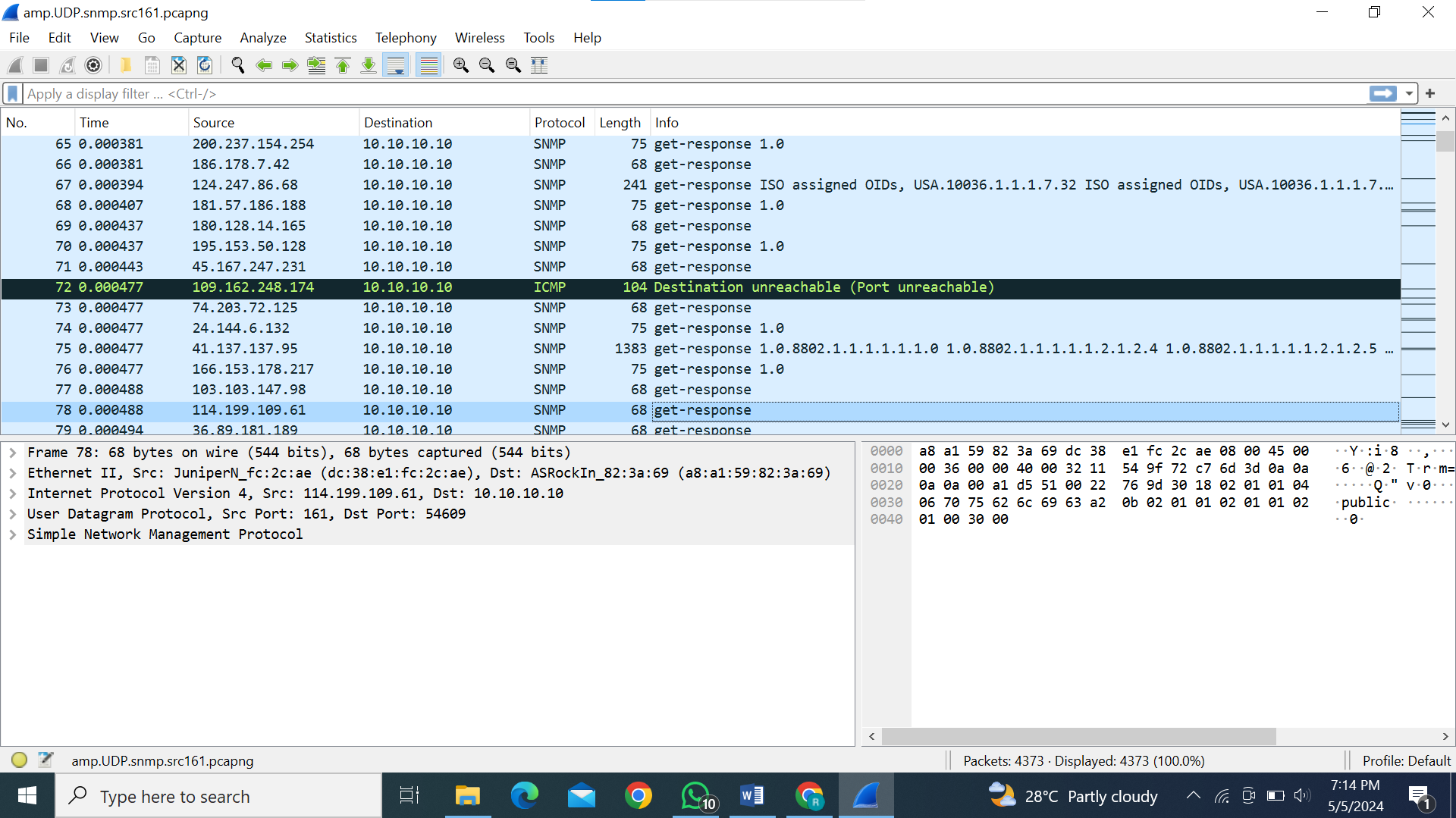
The series of packet captures reveal a coordinated attack against a single destination IP address (10.10.10.10), where each packet contains a TCP RST flag indicating an attempt to forcibly terminate connections. The changing source IP addresses, coupled with the consistent destination IP, suggest a distributed effort to overwhelm the victim's resources with unsolicited reset packets. This behavior aligns with the characteristics of a TCP Reset (RST) flood attack, where attackers flood the victim with RST packets to disrupt legitimate connections, leading to a **denial of service attack.**

**DETAILED ANALYSIS OF 4th PCAP FILE:**

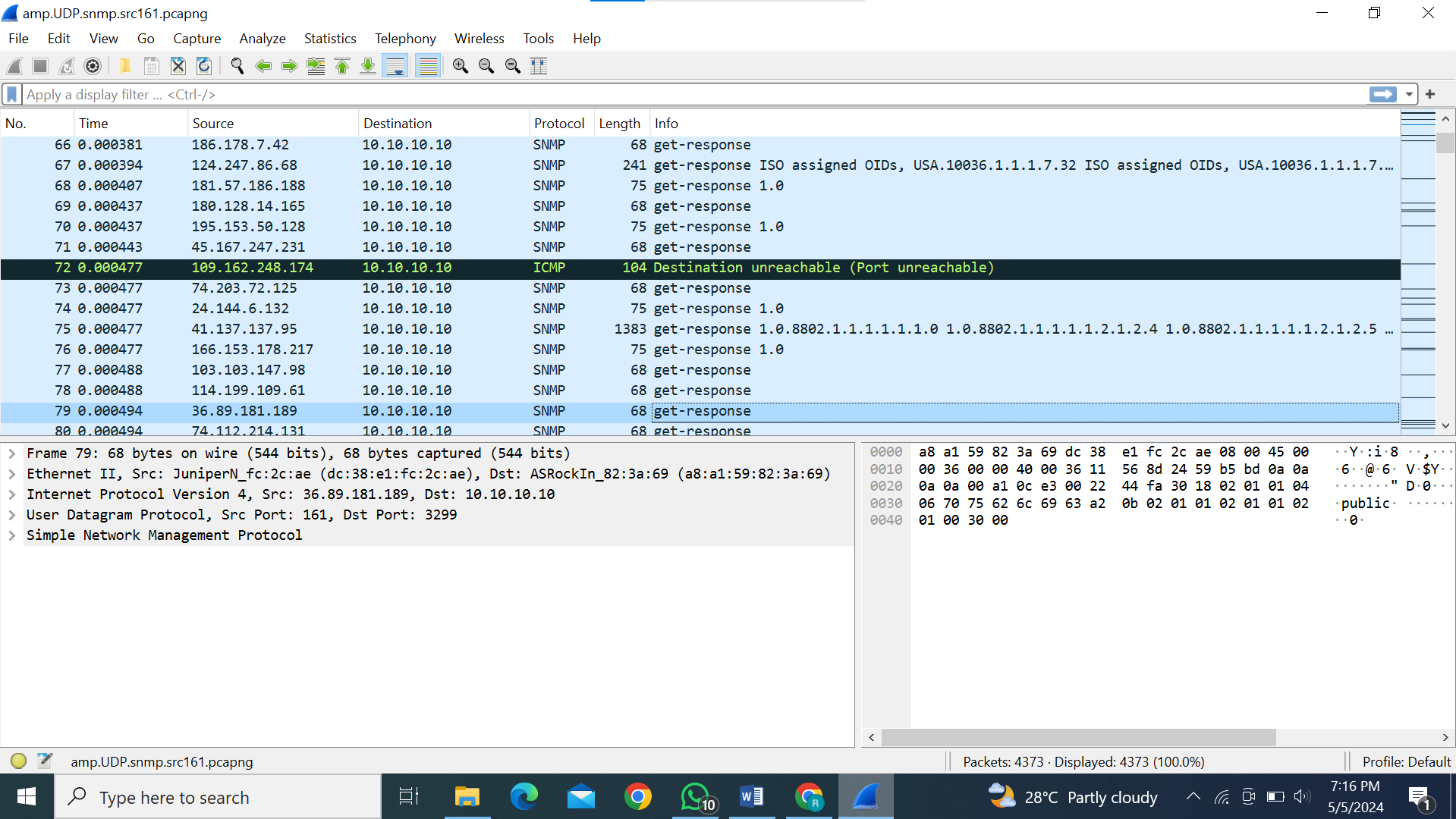
The captured traffic indicates a large volume of SNMP "get-response" messages, suggesting a potential SNMP amplification DDoS attack with spoofed source IP addresses. However, further analysis reveals a pattern of half-open connections characteristic of an SYN-ACK reflection DDoS attack, where spoofed SYN packets overwhelm the victim with incomplete connections.

Then I studied what type of behavior is this and I found this to be an **SNMP amplification DDoS attack.**

**Spoofed IP:**

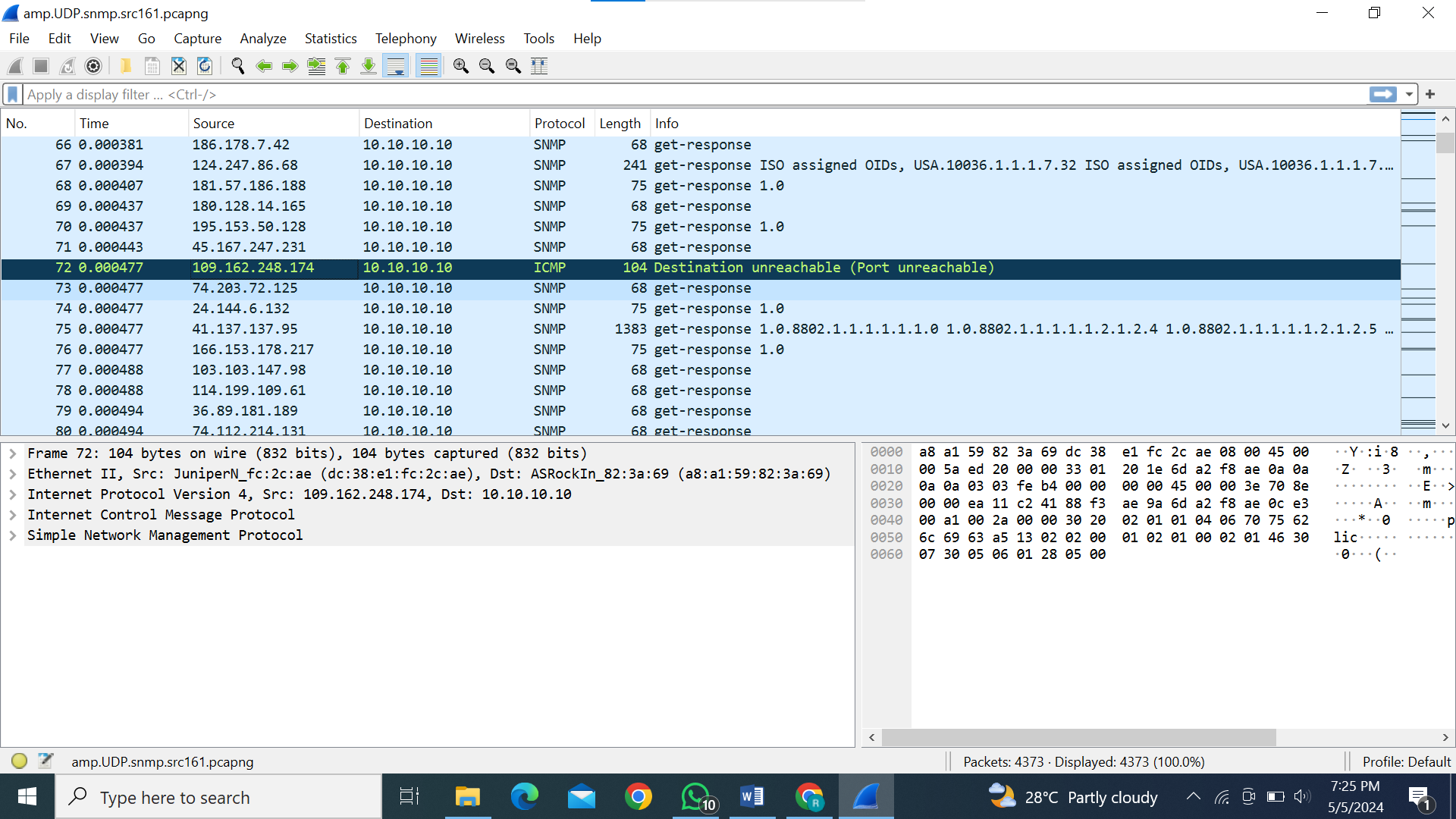


The IP address changes in the next packet while the Mac address remains the same in the next packet.



The Mac address is the same as the previous but the Ip address changes.

**Firewall/Security implications:**



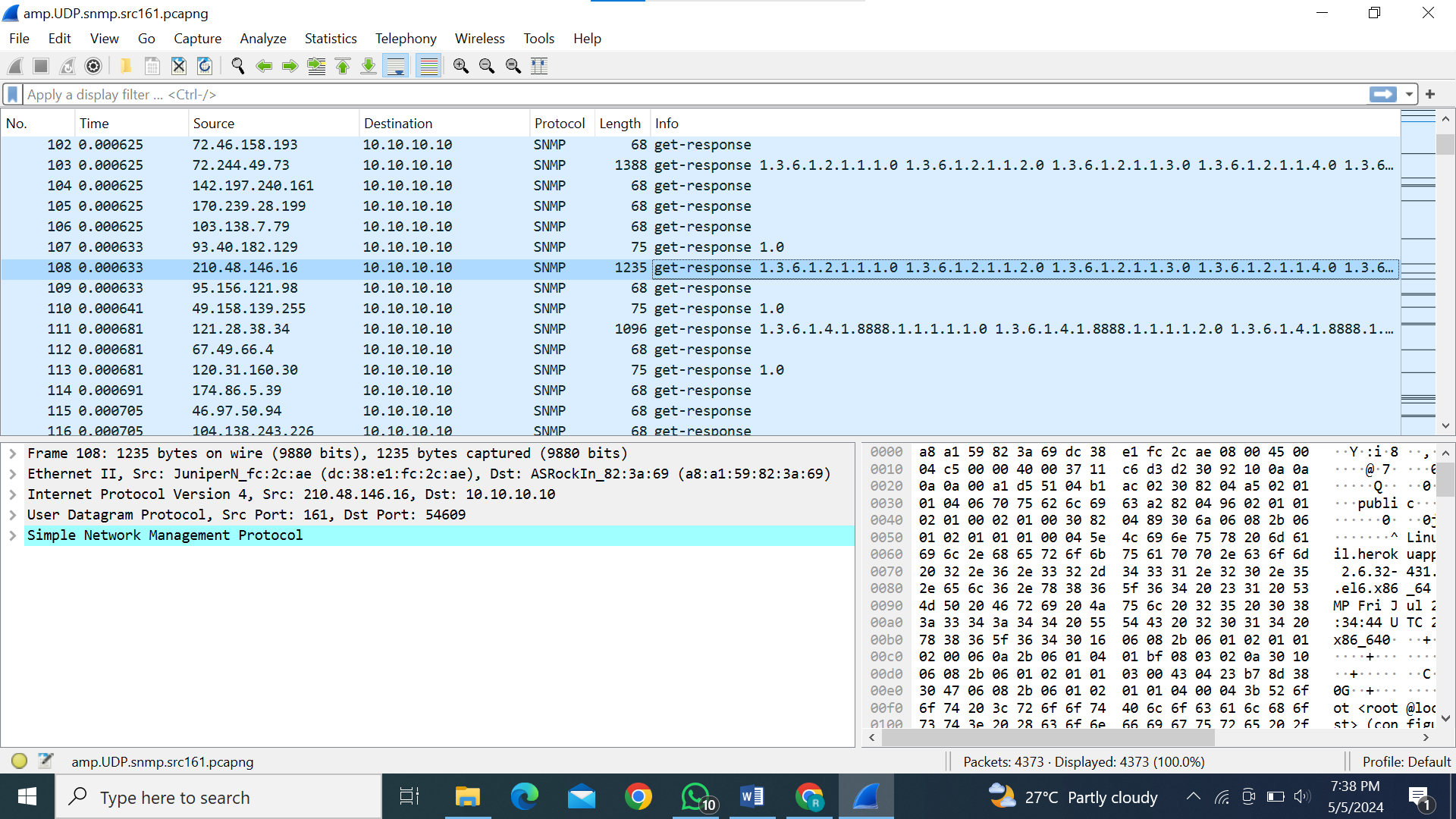
It is unreachable on ICMP, at the same Mac address it is accessing SNMP. It means that the port is unavailable or blocked through a firewall or some security implications.

**Time to live issues:**



Whenever a user tries to send the packet through ICMP, it gives two errors one is explained above and the other is “time to live”. It usually occurs when the port is unable or out of reach for spoofed Ip.

**Monitoring and gaining information through SNMP:**



This packet capture shows an SNMP "get-response" message with a significant number of OID (Object Identifier) requests. The SNMP agent located at IP address 10.10.10.10 is responding to a management query initiated by the SNMP manager at IP address 72.244.49.73. The OID requests typically retrieve various system and network information, such as system description, interface details, and network statistics, enabling the SNMP manager to monitor and manage the device's configuration and performance.

The pcap contains around 5000 packets and around 4950 SNMP packets.

If a file contains mostly SNMP packets with spoofed IP addresses and includes responses like "get-response," it suggests the possibility of an **SNMP amplification attack**. In an SNMP amplification attack, the attacker sends forged requests with the victim's IP address as the source to multiple SNMP devices. These devices then respond with large amounts of data to the victim, overwhelming its network resources. The use of spoofed IP addresses makes it difficult to trace the origin of the attack, increasing its effectiveness.

The analysis ends up summing up the result that it is an **SNMP amplification attack.**