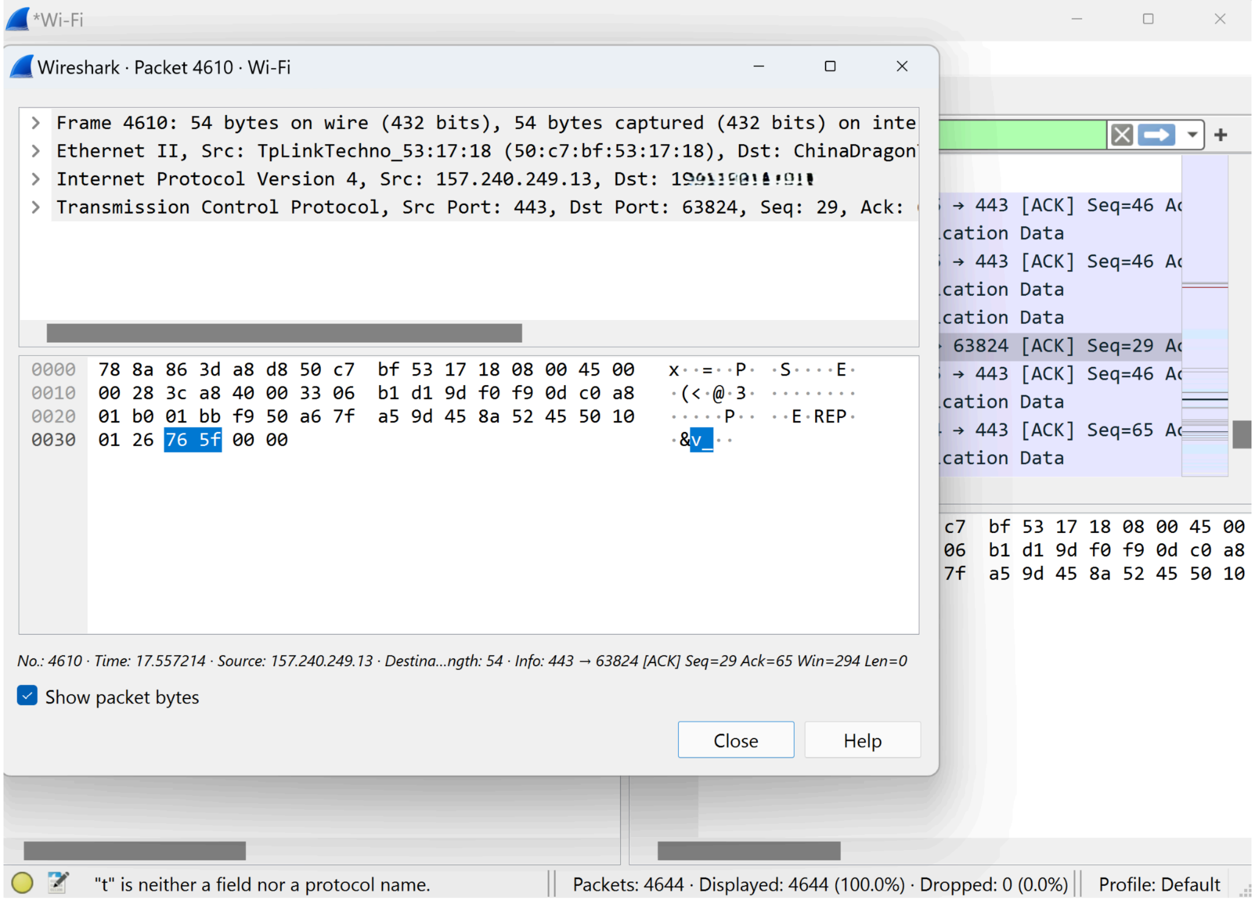

```
> Frame 4610: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on inte
> Ethernet II, Src: TpLinkTechno_53:17:18 (50:c7:bf:53:17:18), Dst: ChinaDragon
> Internet Protocol Version 4, Src: 157.240.249.13, Dst: 1[redacted]
> Transmission Control Protocol, Src Port: 443, Dst Port: 63824, Seq: 29, Ack:
```

```
0000  78 8a 86 3d a8 d8 50 c7  bf 53 17 18 08 00 45 00   x··=··P· ·S····E·
0010  00 28 3c a8 40 00 33 06  b1 d1 9d f0 f9 0d c0 a8   ·(<·@·3· ········
0020  01 b0 01 bb f9 50 a6 7f  a5 9d 45 8a 52 45 50 10   ·····P·· ··E·REP·
0030  01 26 76 5f 00 00                                  ·&v_··
```

```
→ 443 [ACK] Seq=46 Ac
cation Data
→ 443 [ACK] Seq=46 Ac
cation Data
cation Data
63824 [ACK] Seq=29 Ac
→ 443 [ACK] Seq=46 Ac
cation Data
→ 443 [ACK] Seq=65 Ac
cation Data

c7  bf 53 17 18 08 00 45 00
06  b1 d1 9d f0 f9 0d c0 a8
7f  a5 9d 45 8a 52 45 50 10
```

No.: 4610 · Time: 17.557214 · Source: 157.240.249.13 · Destina...ngth: 54 · Info: 443 → 63824 [ACK] Seq=29 Ack=65 Win=294 Len=0

☑ Show packet bytes

"t" is neither a field nor a protocol name. | Packets: 4644 · Displayed: 4644 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

In the screenshot I've provided, I'm looking at a detailed view of a single packet captured using Wireshark. Here's a breakdown of what I am seeing:

Overview of the Packet:

- **Frame 4610**: Indicates that this is the 4610th packet captured during my session.
- **54 bytes on wire (432 bits)**: The packet is 54 bytes in size as it was captured from the network.
- **Ethernet II**: This is the data link layer protocol, showing the source and destination MAC addresses.
  - **Src:** `TplinkTechno_53:17:18` (Source MAC Address).
  - **Dst:** `ChinaDragon_...` (Destination MAC Address).

Network Layer Information:

- **Internet Protocol Version 4 (IPv4)**:
  - **Source IP Address:** `157.240.249.13`

    - **Destination IP Address:** `BLURRED OUT FOR PRIVACY`
    - These addresses represent the endpoints of the packet's journey over the network. The source IP is typically the server I am communicating with, while the destination IP is my local machine.

Transport Layer Information:

- **Transmission Control Protocol (TCP)**:
    - **Src Port:** `443` (Source Port)
    - **Dst Port:** `63824` (Destination Port)
    - **Seq:** `29`, **Ack:** `65`, **Win:** `294`
    - **Port 443** is the default port for HTTPS traffic, indicating that this packet is part of a secure web session. The destination port (`63824`) is a dynamically allocated port on my local machine.
    - **Seq** and **Ack** numbers are part of the TCP header, used to ensure reliable communication.

Packet Details (Hexadecimal and ASCII Representation):

- The bottom pane shows the raw data of the packet in both hexadecimal and ASCII formats:
    - **Hexadecimal View:** Displays the actual bytes of the packet. This is the data as it appears on the wire.
    - **ASCII View:** Displays the corresponding ASCII characters where applicable. The `.` represents non-printable characters.

Summary:

This packet is part of a secure web session (HTTPS), specifically sent from a server to my local machine. The data in the packet is encrypted, which is why the ASCII representation doesn’t contain human-readable text, except for a few visible characters. If I'm analyzing web traffic, I would typically focus on finding the initial HTTP request and the corresponding response before encryption, which would appear before the data is encrypted. However, since this is HTTPS, I will generally see the encryption details and not be able to decrypt unless I have the ability to decrypt the traffic (which requires access to the server’s private key or other methods, like a man-in-the-middle setup in a controlled environment).