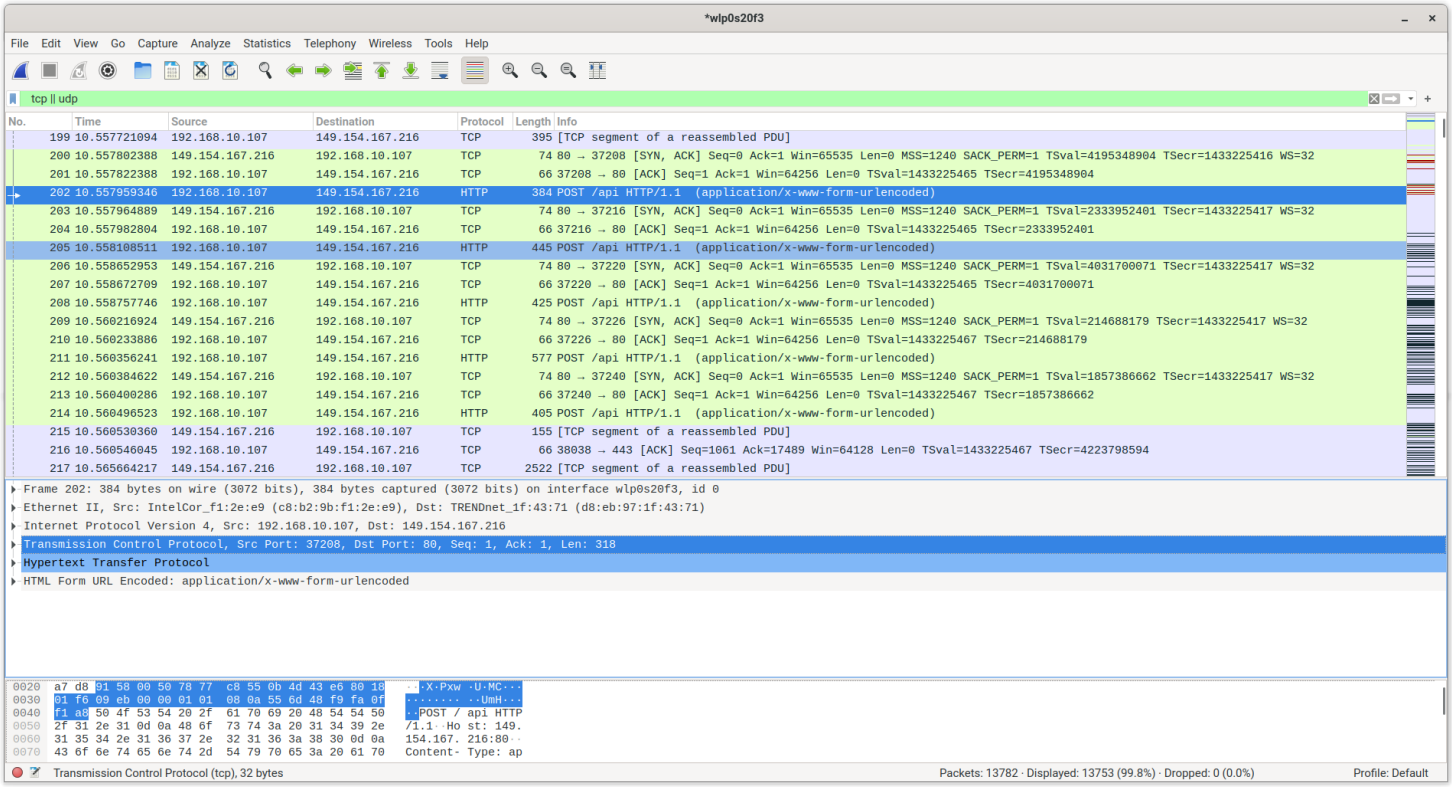
1. Порти

1. Захоплення проектів

Як http сайт брався: http://example.com/

---

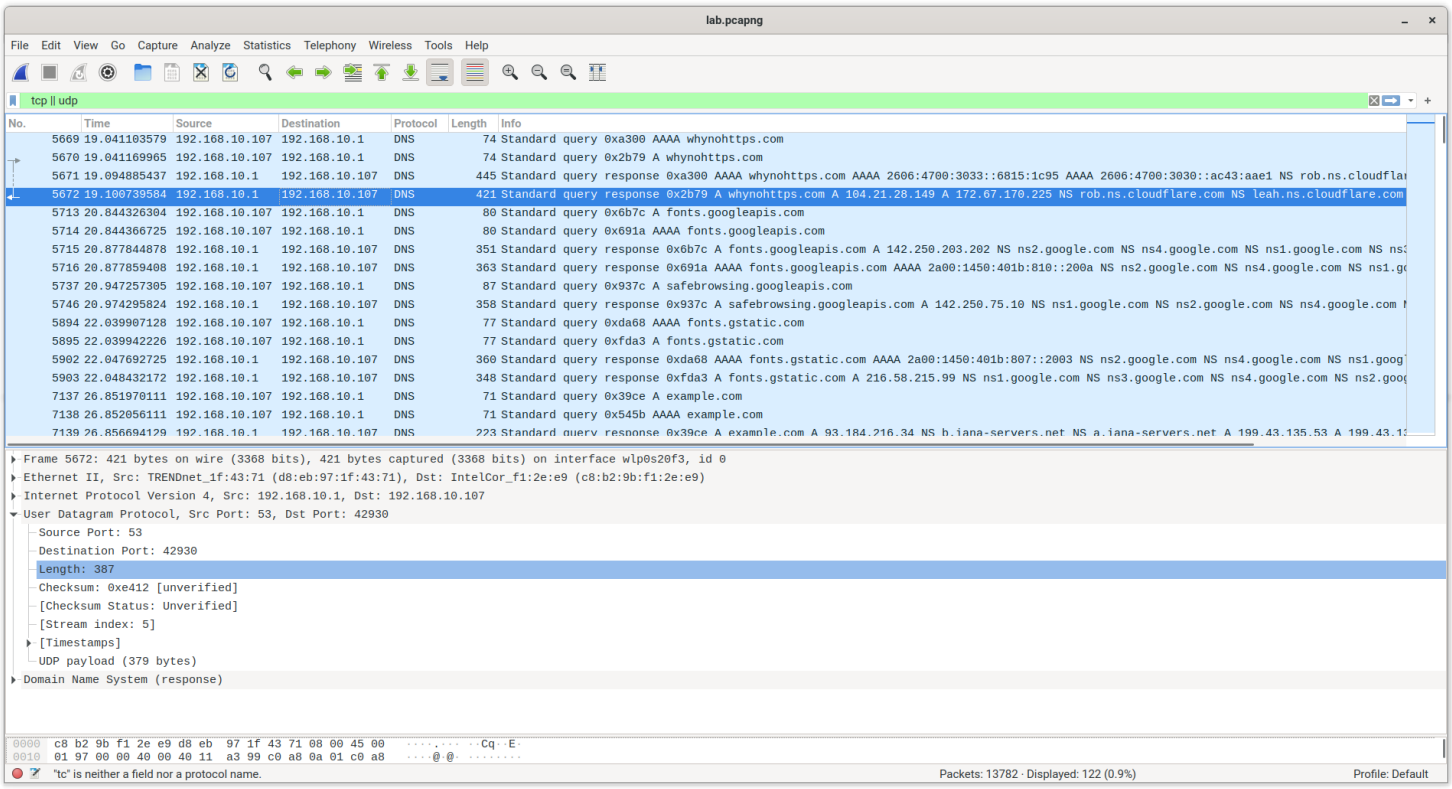
2. Фільтрація по tcp || udp



На скріншоті вище, ми бачимо приклад того, що нам відображаються і інші протокли окрім TCP і UDP - наприклад DNS, HTTP. Вони відображаються тому що, DNS використовує 53 порт UDP(або 53 порт TCP), HTTP - 80 порт TCP.

---

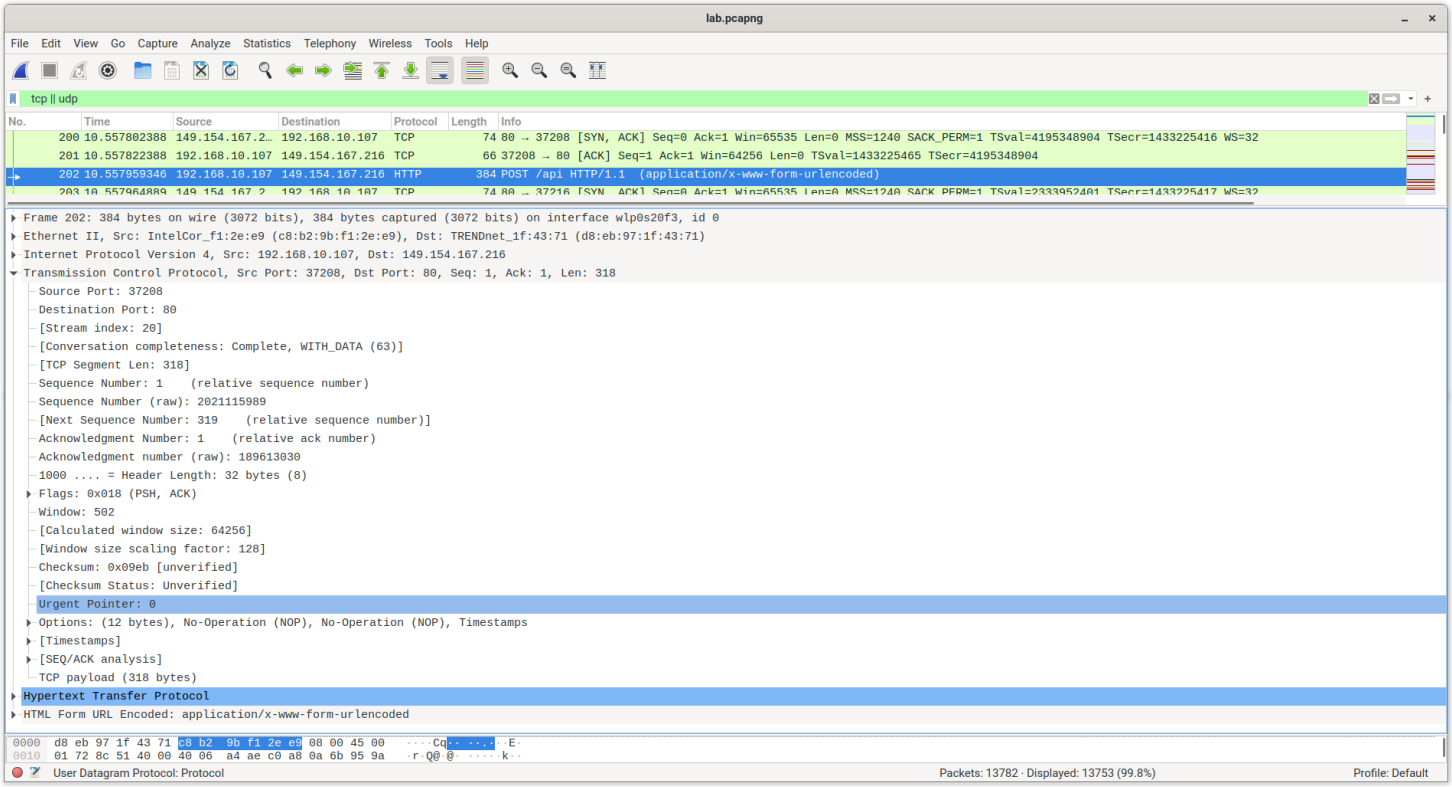
3. UDP



Порт відправника: **42930** (згенерований операційною системою)
Порт одержувача: **53** (закріплений за DNS)

---

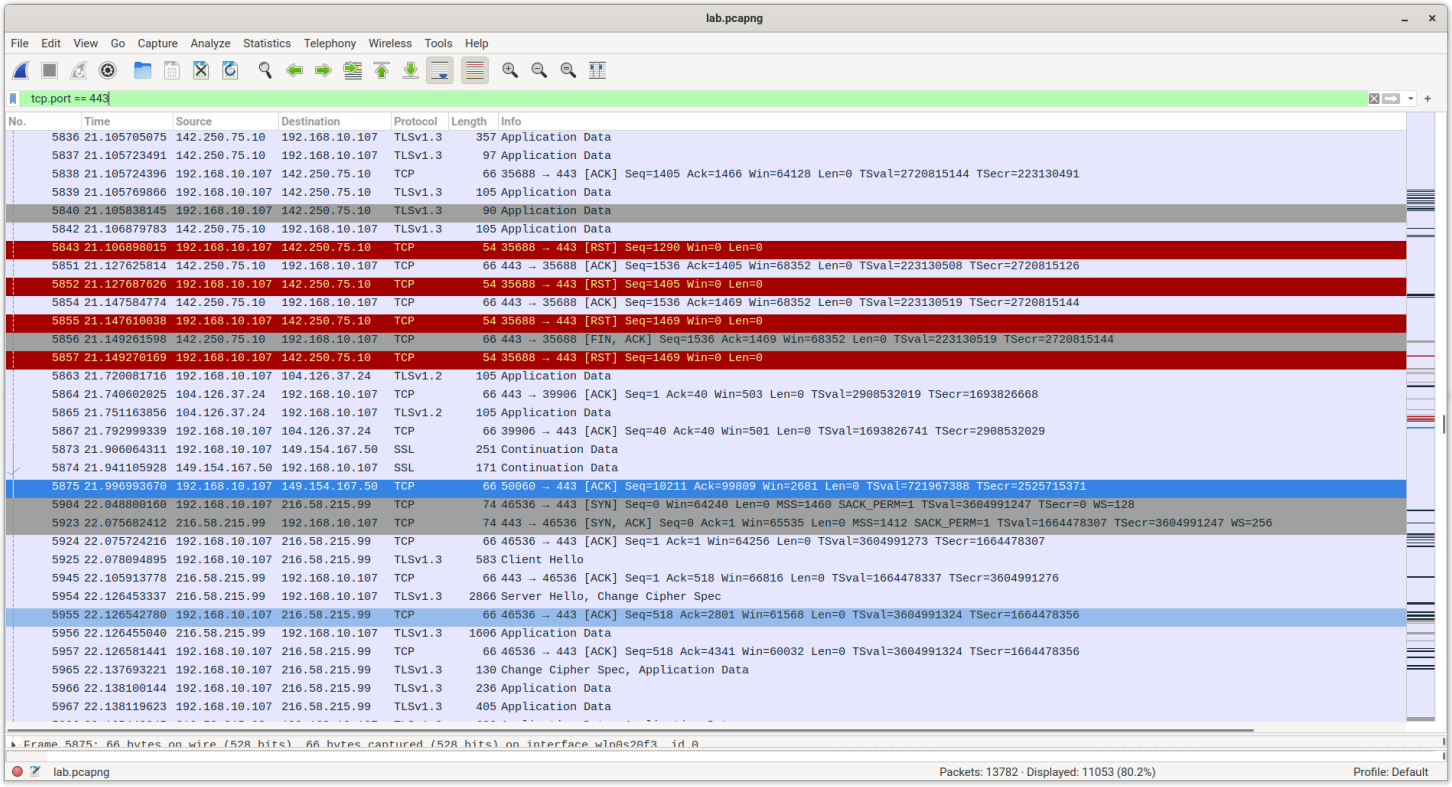
4. HTTP



Порт відправника: **37208** (згенерований операційною системою)
Порт одержувача: **80** (закріплений за HTTP)

---

5. HTTPS port



Пофільтруємо пакети за стандартним для HTTPS портом - 443. В стопці `Protocol`, замість HTTPS містяться такі протоколи, як TLSv1.3, TCP, SSL. Це тому, що HTTPS зашифрований у TCP, через SSL. І тому, що порт tcp == 443, ми підозрюємо, що цей пакет містить в собі HTTPS

---

2. TCP з'єднання

6. Потрійне рукостискання

```
1 0.000000000  192.168.10.108 93.184.216.34   TCP      74 39630 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2481362717 TSecr=0 WS=128
2 0.108194940  93.184.216.34  192.168.10.108 TCP      74 80 → 39630 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1452 SACK_PERM=1 TSval=188616514 TSecr=2481362717 WS=512
3 0.108259151  192.168.10.108 93.184.216.34   TCP      66 39630 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2481362825 TSecr=188616514
```

---

7. Опис полів

Порти:

- Порт відправника(ініціатора рукостикання): **39630** (згенерований системою)
- Порт одержувача: **80** (порт протоколу HTTP)

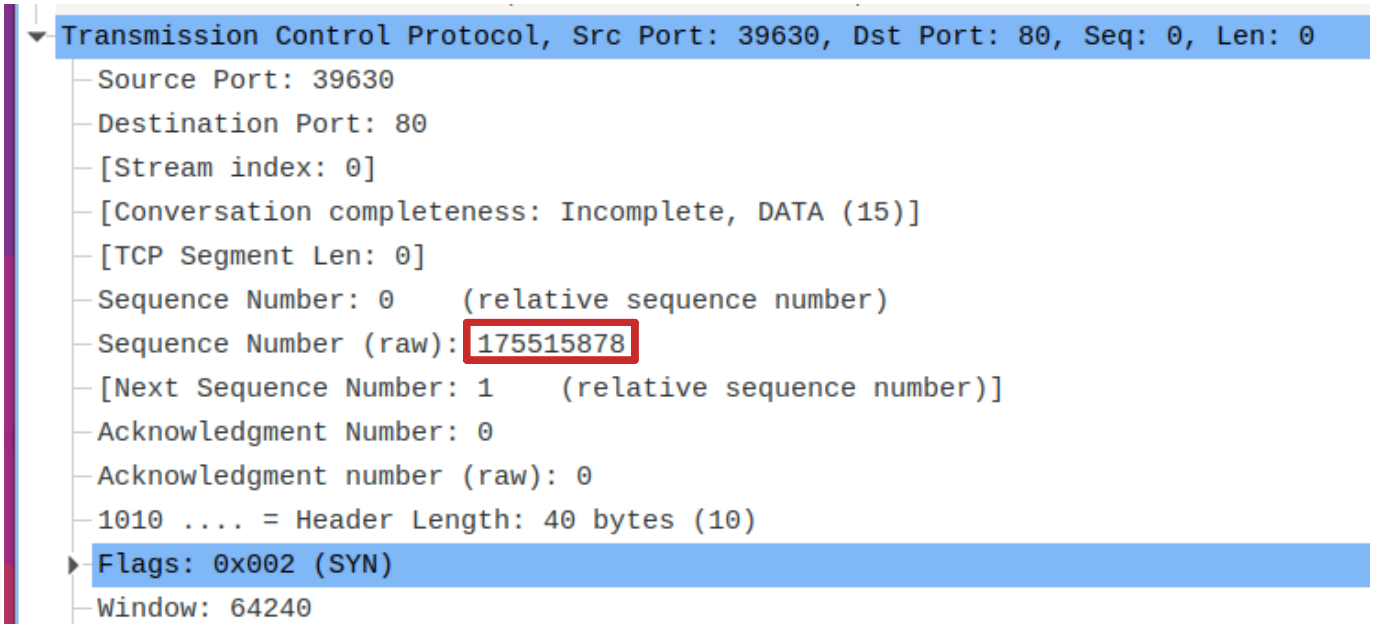
Прапорці:

- SYN(Synchronize): позначає ініціювання та встановлення з'єднання. Також допомагає синхронізувати порядкові номери між пристроями. Відправляється зі сторони клієнта

- ACK(Acknowledge): позначає підтвердження про надходження пакету, з прапорцем *SYN*. Відправляється зі сторони клієнта
- SYN, ACK: комбінація двох прапорців. Відпрвляється зі сторони сервера

Номери послідовностей

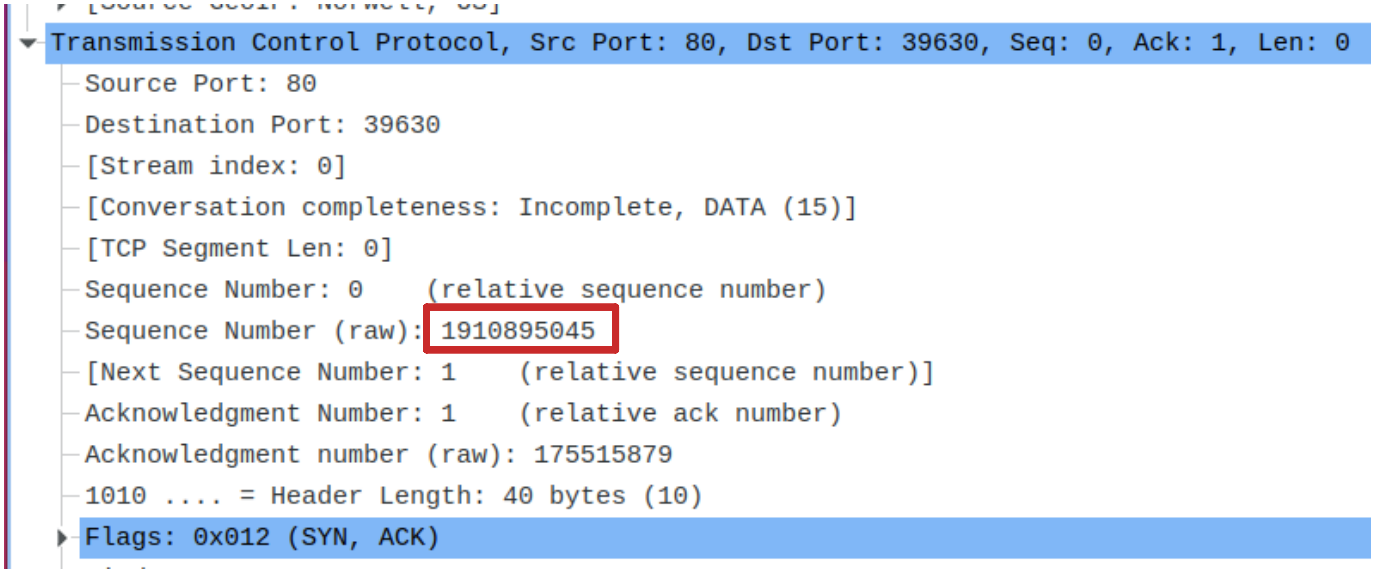
1 пакет



Номер послідовності - **175515878**. Це випадково згенероване число, що вказує на початок порядкових номерів для даних, які має передати

---

2 пакет



Номер послідовності - **1910895045**. Окрім випадково згенерованого `Sequence`

`number`, також містить `Acknowlegment number`, що рівне `Sequence number` першого пакету + 1

---

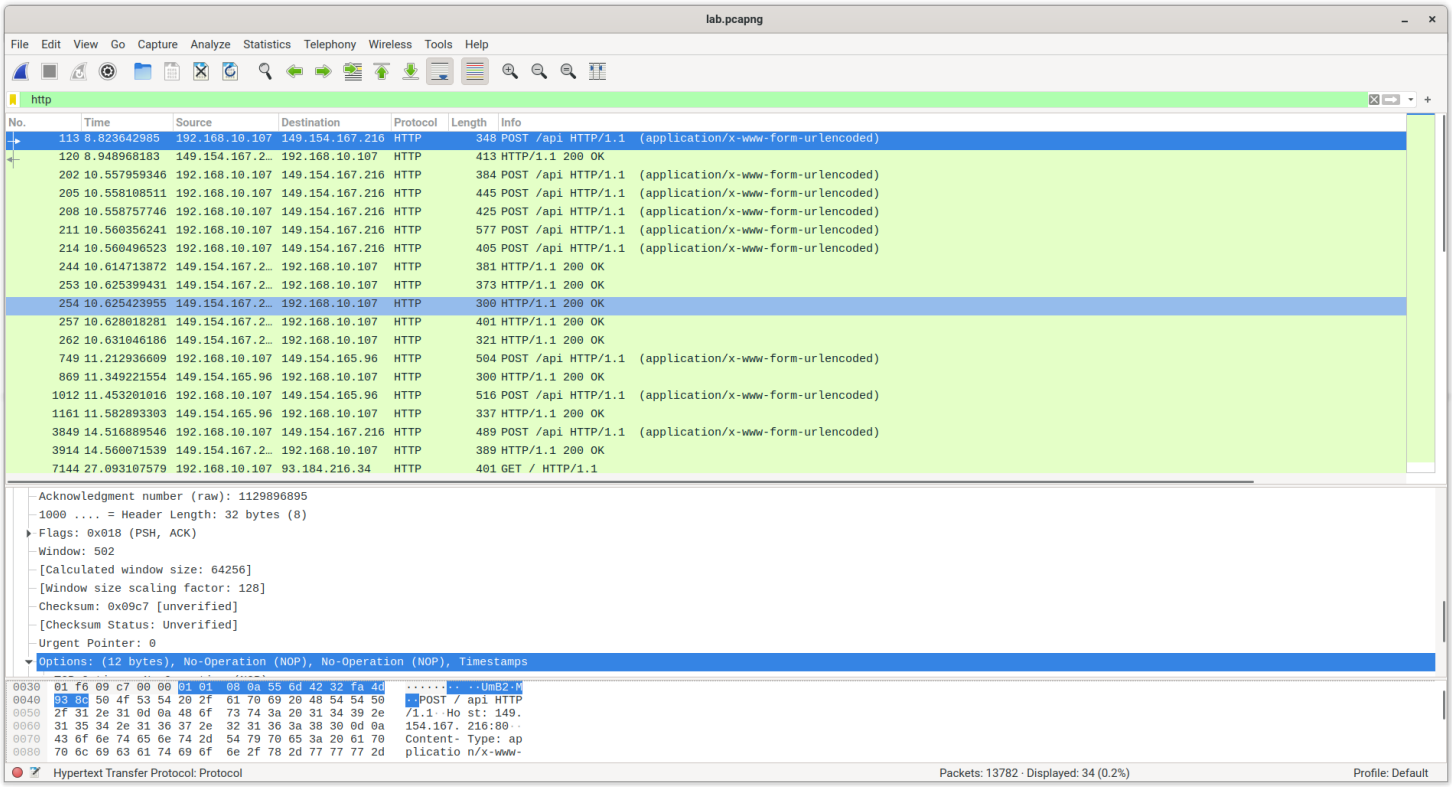
3 пакет

```
Transmission Control Protocol, Src Port: 39630, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 39630
    Destination Port: 80
    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 175515879
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1910895046
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x010 (ACK)
    Window: 502
```

Номер послідовності - **175515879**. Рівне `Sequence number` першого пакету + 1

---

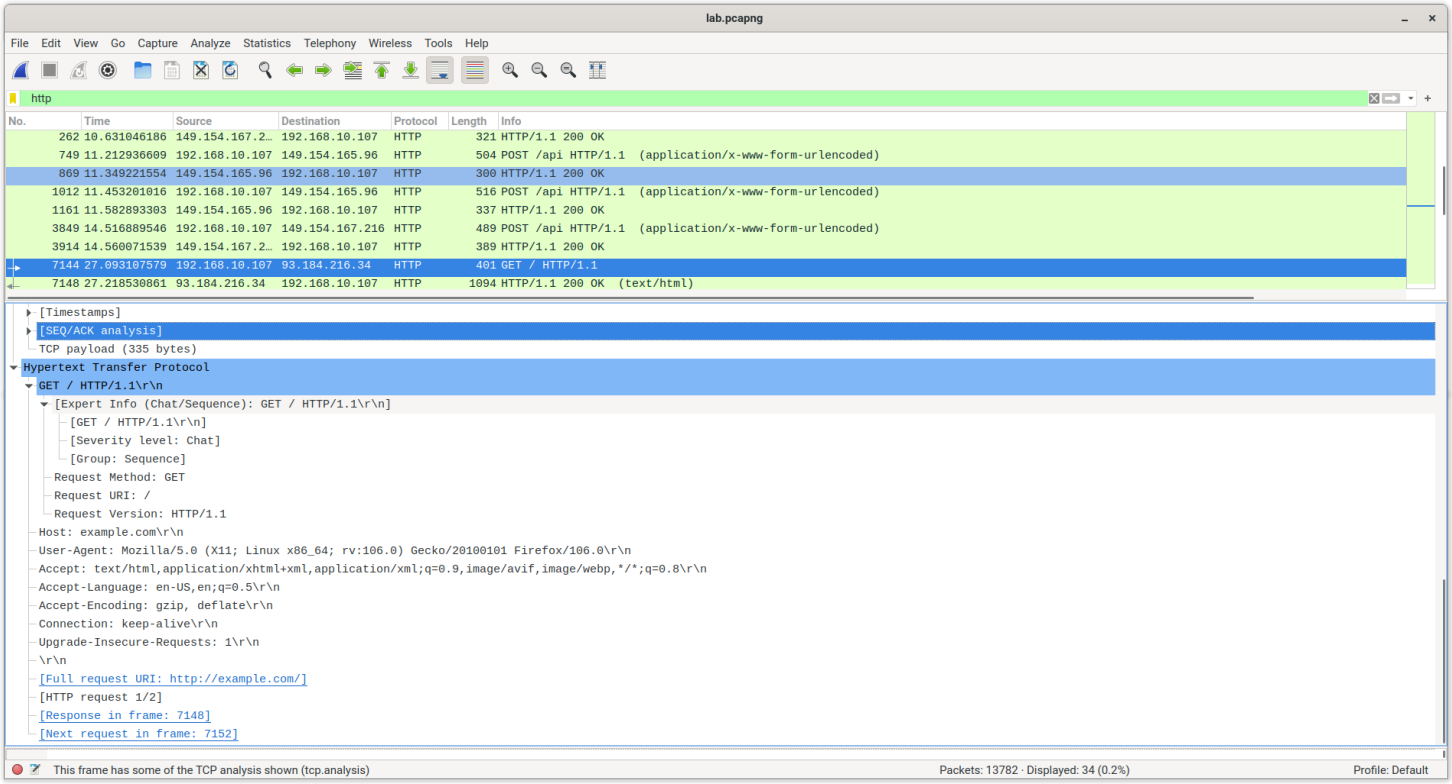
8. HTTP пакети



---

9. Текст запитів

Запит



`GET /HTTP/1.1` - інформація про метод

`Expert info`

`Severity level` - рівень важливості (`chat` == звичайний пакет)

`Group: Sequence` - порядковий номер протоколу був підозрілим, наприклад, він не був безперервним або була виявлена повторна передача.

`Request method` - використовуваний HTTP метод

`Uri` - ідентифікатор ресурсу по сайту

`Version` - версія HTTP

`Host` - хост сайту, на який надіслано запит

`User-Agent` - інформація про програму, операційну система та версію запитуючого агента клієнта

`Accept` - можливі формати, які можуть бути в тілі запиту

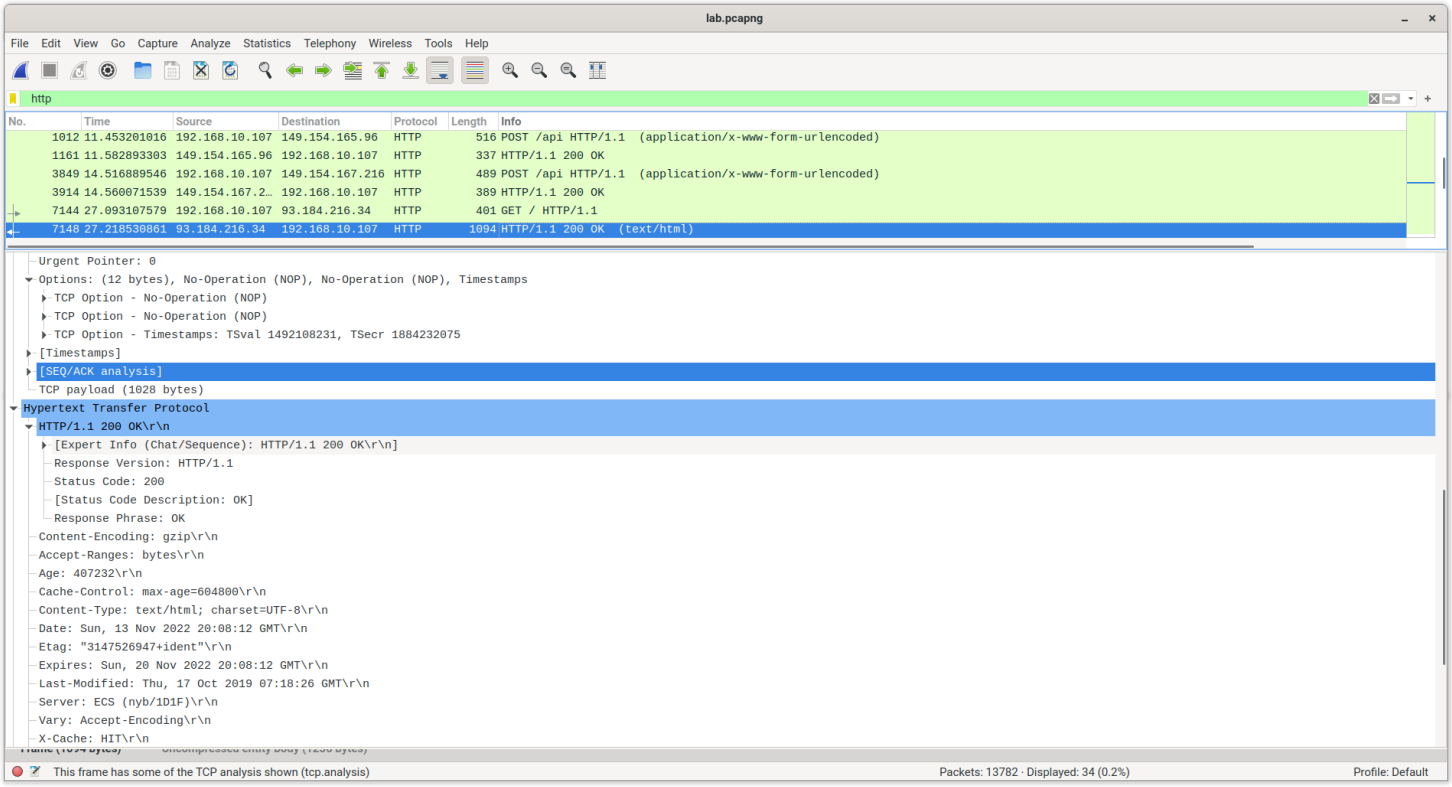
`Accept-Language` - мова, якою клієнт хоче отримувати інформацію

`Accept-Encoding` - кодування вмісту, зазвичай алгоритми стистення

`Connection` - вказує, чи залишиться з'єднання дійсним після завершення транзакції

`Upgrage-Insecure-Requests` - міситить інформацію про те, чи клієнт надає перевагу зашифрованій відповіді. Тобто - чи підвищувати рівень безпеки до HTTPS

Відповідь



`HTTP/1.1 200 OK` - інформація про метод

`Expert info`

`Severity level` - рівень важливості (`chat` == звичайний пакет)

`Group: Sequence` - порядковий номер протоколу був підозрілим, наприклад, він не був безперервним або була виявлена повторна передача.

`Request method` - використовуваний HTTP метод

`Uri` - ідентифікатор ресурсу по сайту

`Version` - версія HTTP

`Status Code` - статус відповіді запиту

`Response phrase` -

`Content-Encoding` - список всіх типів кодувань, що використовувались для вмісту

`Age` - час в секундах, коли відповідь була в кеші проксі

`Cache-Control` - містить інструкції, що керують кешуванням браузера

`Content-Type` - оригінальний тип ресурсу(до кодування вмісту)

`Date` - дата та час відповіді

`Etag` - ідентифікатором певної версії ресурсу

`Expires` - дата та час, після яких відповідь вважається простроченою.

`Last-Modified` - дата й час, коли вихідний сервер вважає, що ресурс було востаннє змінено

`Server` - сервер, який згенерував відповідь

`Vary` - частини повідомлення запиту(окрім методу та адреси) які вплинули на вміст відповіді

---

10 Типи запитів

Окрім GET методів був тільки POST (який ймовірно виник при пошуку повідомлення в програмі "телеграм"). Інших методів не було, бо я не робив відповідних дій - модифікування даних або їхнє видалення.