

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

**Комп'ютерні інформаційні мережі**

**ЛАБОРАТОРНА РОБОТА №8**

Виконав:

Ст. Лук'янчук Денис

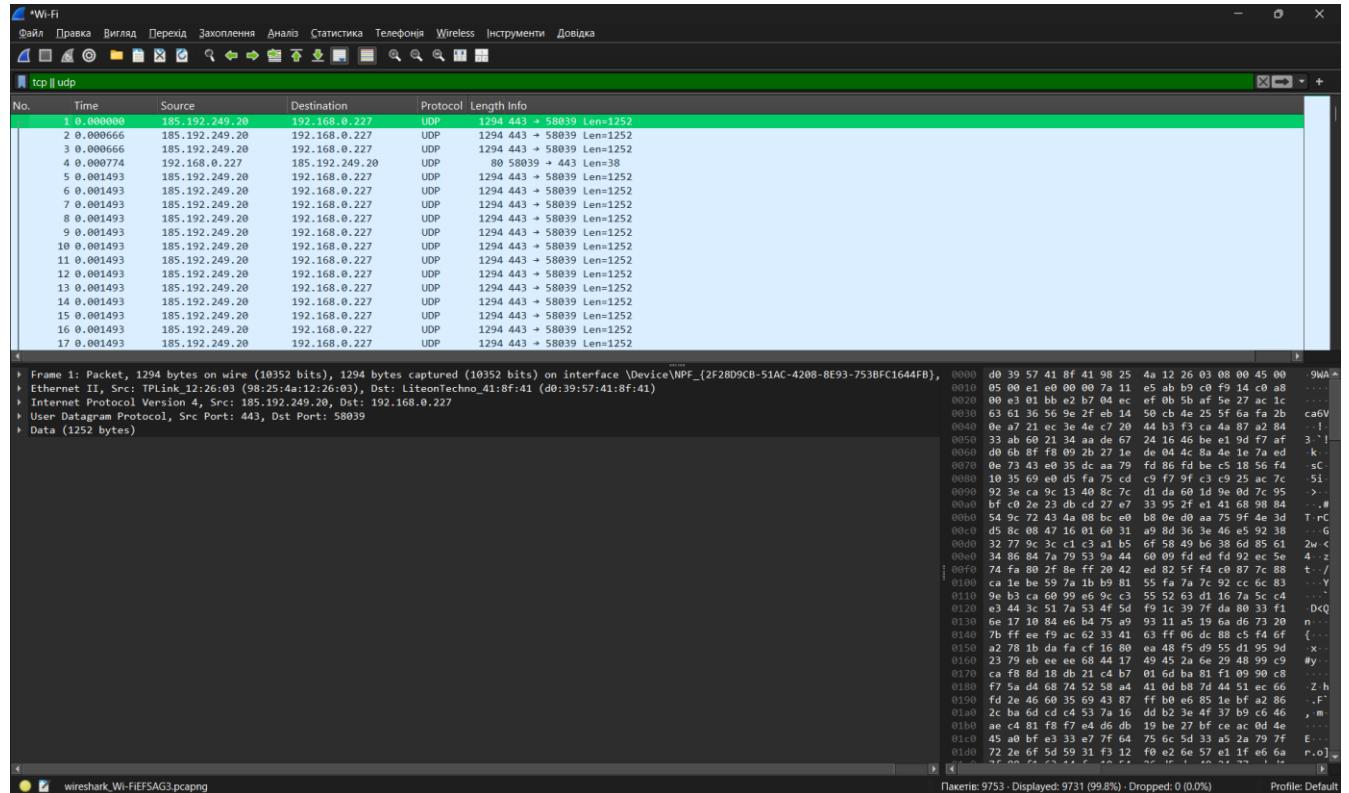
Група ПМі-33

## Тема: “Аналіз TCP-сегментів та UDP-датаграм засобами Wireshark”

**Мета роботи:** Здобути практичні навички з інтерпретації протокольних блоків даних транспортного рівня стеку TCP/IP.

### Хід роботи

1. Використовуючи Wireshark, захопив пакети, здійснюючи активність в браузері, а саме, на сайті, який працює за протоколом http. Також під час захоплення пакетів виконав завантаження файлу розміром понад 5 Мб.
2. Встановив фільтр `tcp || udp`. Пояснив, чому крім вказаних протоколів (TCP та UDP) відображаються і інші, такі як, DNS і HTTP:



DNS — перетворює доменні імена на IP-адреси і зазвичай працює через UDP-порт 53.

HTTP — передає незашифровані веб-дані між клієнтом і сервером через TCP-порт 80.

TLS — забезпечує шифрування даних поверх TCP-порту 443.

QUIC — сучасний протокол Google, який поєднує функції TCP і TLS, але працює поверх UDP-порту 443.

3. Вибрали пакет, який використовує протокол UDP. Вказав порти відправника та одержувача:

```
▼ User Datagram Protocol, Src Port: 443, Dst Port: 58039
    Source Port: 443
    Destination Port: 58039
    Length: 1260
    Checksum: 0xf605 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Stream Packet Number: 5]
    ▶ [Timestamps]
    UDP payload (1252 bytes)
```

Який з цих портів згенерований автоматично операційною системою, а який закріплений за протоколом?

**Порт відправника:** 443 — це стандартний зарезервований порт для протоколів HTTPS / TLS / QUIC, який використовується сервером.

**Порт отримувача:** 58039 — це динамічний порт, автоматично згенерований операційною системою клієнта для встановлення з'єднання.

4. Вибрав пакет, який використовує протокол HTTP. Вказав порти відправника та одержувача:

```
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 59997, Seq: 1, Ack: 346, Len: 298
    Source Port: 80
    Destination Port: 59997
    [Stream index: 15]
    [Stream Packet Number: 7]
    ▶ [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 298]
    Sequence Number: 1 (relative sequence number)
    Sequence Number (raw): 55015916
    [Next Sequence Number: 299 (relative sequence number)]
    Acknowledgment Number: 346 (relative ack number)
    Acknowledgment number (raw): 3546293900
    0101 .... = Header Length: 20 bytes (5)
    ▶ Flags: 0x018 (PSH, ACK)
    Window: 1052
    [Calculated window size: 1052]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x89c2 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    ▶ [Timestamps]
    ▶ [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (298 bytes)
```

Який з цих портів згенерований автоматично операційною системою, а який закріплений за певною протоколом?

**Порт відправника 80** — це стандартний порт для протоколу HTTP, який використовується сервером.

**Порт отримувача 59997** — це динамічний порт, згенерований автоматично операційною системою клієнта для встановлення TCP-з'єднання з сервером.

5. Знаючи закріплений за *HTTPS* порт (443), знайдіть пакети цього протоколу. Як цей протокол відображає *Wireshark* в стовпці *Protocol*? Чому?

```
▼ User Datagram Protocol, Src Port: 443, Dst Port: 58039
  Source Port: 443
  Destination Port: 58039
  Length: 1260
  Checksum: 0xd9cd [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  [Stream Packet Number: 5305]
  ▶ [Timestamps]
  UDP payload (1252 bytes)
```

Цей пакет передає дані через порт 443, який зарезервований за протоколом *HTTPS* / *TLS* / *QUIC*, тобто зашифрованим варіантом *HTTP*. *Wireshark*, коли знаходить такі пакети, у стовпці *Protocol* відображає не *HTTPS*, а *TLS* або *QUIC*.

*Wireshark* не може бачити вміст зашифрованого трафіку *HTTPS*, тому замість *HTTPS* показує рівень шифрування (*TLS*) або протокол передачі (*QUIC*), який працює поверх *UDP*.

6. Відшукав послідовність пакетів процедурі “потрійного рукостискання”:

1353 5.499085	192.168.0.227	194.44.11.136	TCP	66 56858 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
1354 5.505760	194.44.11.136	192.168.0.227	TCP	66 80 → 56850 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
1355 5.505760	142.250.186.110	192.168.0.227	HTTP	597 HTTP/1.1 301 Moved Permanently
1356 5.505824	192.168.0.227	194.44.11.136	TCP	54 56850 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0

7. Опишіть вміст кожного сегменту з попереднього пункту (порти, пропорці, номери послідовностей):

**SYN (перший сегмент):**

```
Transmission Control Protocol, Src Port: 56858, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 56858
  Destination Port: 80
  [Stream index: 14]
  [Stream Packet Number: 1]
  ▶ [Conversation completeness: Incomplete, CLIENT_ESTABLISHED (3)]
    [TCP Segment Len: 0]
    Sequence Number: 0 (relative sequence number)
    Sequence Number (raw): 490902561
    [Next Sequence Number: 1 (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  ▶ [Flags: 0x002 (SYN)]
    Window: 65535
    [Calculated window size: 65535]
    Checksum: 0x5002 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ▶ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  ▶ [Timestamps]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
```

- Клієнт: 192.168.0.227 → Сервер: 194.44.11.136

- Порти: 56858 → 80 (80 — стандартний порт HTTP).
- Прапорець: SYN = 1 — запит на встановлення з'єднання.
- Відносний номер Seq = 0, справжній номер (raw) = 499002561. Це випадкове 32-бітне число, яке TCP генерує як початковий номер послідовності клієнта.

## SYN+ACK (другий сегмент)

```
Transmission Control Protocol, Src Port: 80, Dst Port: 56850, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 56850
  [Stream index: 6]
  [Stream Packet Number: 2]
  ▶ [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0 (relative sequence number)
    Sequence Number (raw): 2899453764
    [Next Sequence Number: 1 (relative sequence number)]
    Acknowledgment Number: 1 (relative ack number)
    Acknowledgment number (raw): 3497690188
    1000 .... = Header Length: 32 bytes (8)
  ▶ Flags: 0x012 (SYN, ACK)
  Window: 64240
  [Calculated window size: 64240]
  Checksum: 0xdd8f [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
  ▶ [Timestamps]
  ▶ [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
```

- Сервер: 194.44.11.136 → Клієнт: 192.168.0.227
- Порти: 80 → 56858.
- Прапорці: SYN = 1, ACK = 1 — підтвердження запиту клієнта й сигнал готовності.
- Seq (відносний) = 0, Ack (відносний) = 1.
- Справжні номери: Seq (raw) = 2899453764, Ack (raw) = 3497690188. Сервер також генерує своє випадкове 32-бітне число як початковий номер послідовності.

## ACK (третій сегмент)

```
Transmission Control Protocol, Src Port: 56850, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
  Source Port: 56850
  Destination Port: 80
  [Stream index: 6]
  [Stream Packet Number: 3]
  ▶ [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 1 (relative sequence number)
    Sequence Number (raw): 3497690188
    [Next Sequence Number: 1 (relative sequence number)]
    Acknowledgment Number: 1 (relative ack number)
    Acknowledgment number (raw): 2899453765
    0101 .... = Header Length: 20 bytes (5)
  ▶ Flags: 0x010 (ACK)
  Window: 255
  [Calculated window size: 65280]
  [Window size scaling factor: 256]
  Checksum: 0x1854 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  ▶ [Timestamps]
  ▶ [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
```

- Клієнт: 192.168.0.227 → Сервер: 194.44.11.136
- Порти: 56858 → 80.
- Пропорець: ACK = 1 — підтвердження встановлення з'єднання.
- Seq (відносний) = 1, Ack (відносний) = 1.
- Справжні номери: Seq (raw) = 3497690188, Ack (raw) = 2899453765. Клієнт підтверджує отримання пакета SYN+ACK від сервера.

8. Використовуючи фільтр tls, отримав пакети криптографічного протоколу TLS.

No.	Time	Source	Destination	Protocol	Length Info
1207 0.904897	192.168.0.227	142.250.185.110		TLSv1.3	542 Client Hello (SNI=www.youtube.com)
1209 0.946936	142.250.185.110	192.168.0.227		TLSv1.3	2878 Server Hello, Change Cipher Spec
1212 0.947098	142.250.185.110	192.168.0.227		TLSv1.3	838 Application Data
1214 0.949340	192.168.0.227	142.250.185.110		TLSv1.3	118 Change Cipher Spec, Application Data
1215 0.950024	192.168.0.227	142.250.185.110		TLSv1.3	146 Application Data
1218 0.950123	192.168.0.227	142.250.185.110		TLSv1.3	282 Application Data
1219 0.983666	142.250.185.110	192.168.0.227		TLSv1.3	670 Application Data, Application Data
1220 0.983666	142.250.185.110	192.168.0.227		TLSv1.3	85 Application Data
1222 0.984790	192.168.0.227	142.250.185.110		TLSv1.3	85 Application Data
1224 1.001878	142.250.185.110	192.168.0.227		TLSv1.3	317 Application Data
1225 1.001878	142.250.185.110	192.168.0.227		TLSv1.3	85 Application Data
1227 1.002764	142.250.185.110	192.168.0.227		TLSv1.3	93 Application Data
1228 1.002871	192.168.0.227	142.250.185.110		TLSv1.3	93 Application Data
1231 1.765837	192.168.0.227	52.112.100.68		TLSv1.2	110 Application Data
1232 1.776570	52.112.100.68	192.168.0.227		TLSv1.2	99 Application Data
1234 2.881208	192.168.0.227	146.75.117.91		TLSv1.2	652 Application Data
1236 3.116647	146.75.117.91	192.168.0.227		TLSv1.2	498 Application Data
1237 3.119775	192.168.0.227	146.75.117.91		TLSv1.2	655 Application Data
1239 3.329151	146.75.117.91	192.168.0.227		TLSv1.2	391 Application Data
1240 3.334591	192.168.0.227	146.75.117.91		TLSv1.2	664 Application Data

9. Відшукав пакети, які стосуються процедури TLS-рукостискання та описав їх:

9665 22.742615	192.168.0.227	194.44.11.136	TLSv1.3	494 Client Hello (SNI=www.nbuv.gov.ua)
9671 22.770537	194.44.11.136	192.168.0.227	TLSv1.3	1514 Server Hello, Change Cipher Spec, Application Data
9672 22.770537	194.44.11.136	192.168.0.227	TLSv1.3	1514 Application Data
9673 22.770537	194.44.11.136	192.168.0.227	TLSv1.3	276 Application Data, Application Data
9675 22.771631	192.168.0.227	194.44.11.136	TLSv1.3	78 Application Data

**Client Hello** — клієнт ініціює захищене з'єднання з сайтом і надсилає версію TLS і параметри шифрування.

**Server Hello** — сервер відповідає, вибирає параметри шифрування та переходить у зашифрований режим.

**Application Data** — далі передаються зашифровані HTTPS-дані, які Wireshark не може розшифрувати.

10. Вибравши пакет з даними, переконався, що вони зашифровані:

```

Frame 6087: Packet, 368 bytes on wire (2944 bits), 368 bytes captured (2944 bits) on interface \Device\NPF_{2F28D9CB-51AC-4208-8E93-753BFC1644FB}, 
  ▶ Ethernet II, Src: LiteonTechno_41:8f:41 (d0:39:57:41:8f:41), Dst: TPLink_12:26:03 (98:25:4a:12:26:03)
  ▶ Internet Protocol Version 4, Src: 192.168.0.227, Dst: 151.101.66.137
  ▶ Transmission Control Protocol, Src Port: 56862, Dst Port: 443, Seq: 2057, Ack: 4962, Len: 314
  ▶ Transport Layer Security
    [Stream index: 6]
      ▶ TLSv1.3 Record Layer: Application Protocol: Hypertext Transfer Protocol
        Opaque Type: Application Data (23)
        Version: TLS 1.2 (0x0303)
        Length: 309
        Encrypted Application Data [...]: f57c1bcf93577880365d3c93477fdf77a9269b852e7c31fc8eaa3ed6bee81d35fd71b51fc4cb1fa9e5c4bba910e8b3ec43917b5309b49
        [Application Data Protocol: Hypertext Transfer Protocol]

```

**Висновок:** У ході виконання лабораторної роботи я отримав знання та практичні навички з аналізу транспортних протоколів TCP і UDP у програмі Wireshark, дослідив роботу протоколів HTTP, HTTPS (TLS), DNS та QUIC, навчився визначати порти відправника й одержувача, спостерігати процес встановлення TCP-з'єднання (“потрійне рукостискання”), а також проаналізував етапи TLS-рукостискання та переконався, що передача даних у захищених з'єднаннях здійснюється у зашифрованому вигляді.