Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Компьютерные сети

Лабораторная работа 5

Выполнили:

Кривоносов Егор Дмитриевич Марков Петр Денисович

Группа: Р33111

Преподаватель:

Тропченко Андрей Александрович

2022 г.

Санкт-Петербург

Цель работы

Изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

Задание

- 1. Запустить Wireshark. В появившемся окне выбрать интерфейс, для которого необходимо осуществлять анализ проходящих через него пакетов. В качестве интерфейса, используемого для захвата трафика, выбрать физический адаптер, через который компьютер подключён к Интернету (обычно этот адаптер называется Local или "Подключение по локальной сети"). Если меню для выбора адаптера не появляется при запуске Wireshark, нужно запустить из "Меню" команду "Capture->Options". После выбора адаптеру, нужно запустить процесс захвата трафика (кнопка Start).
- 2. Инициировать процесс передачи трафика по сети (например, в браузере открыть сайт, заданный по варианту, или запустить соответствующую сетевую утилиту).
- 3. Установить значение "Фильтра", чтобы из всего множества перехватываемых пакетов Wireshark отобразил только те, которые имеют отношение к выполняемому заданию. Для корректного создания фильтра следует пользоваться всплывающими подсказками Wireshark, которые активизируются при наборе фильтра. В качестве альтернативного способа можно использовать интерактивный конструктор фильтра, нажав на кнопку "Expression" в правой части элемента "Фильтр"
- 4. Дождаться появления данных в списке захваченных пакетов и убедиться, что количество пакетов достаточно для выполнения задания.
- 5. Сохранить захваченный трафик в файл-трассу (рсар).
- 6. Описать в отчёте структуру наблюдаемых PDU (т.е. протокольных блоков данных: кадров, пакетов, сегментов) как для запросов, так и ответов. Указать название и назначение всех заголовков всех уровней OSI-модели в пакетах с учётом порядка инкапсуляции (для этого нужно раскрывать соответствующие значки «+» в поле с детальной информацией о выбранном пакете).
- 7. Написать в отчёте ответы на вопросы задания (для этого может потребоваться самостоятельно изучить назначение соответствующей заданию сетевой утилиты, использованной для создания трафика).
- 8. Поместить в отчёт скриншоты окна Wireshark, иллюстрирующие ответы из вышеуказанных п.6 и п.7.

URL - Адрес сайта, в названии которого лексически входит фамилия студента: https://krivonosov.ru/

Выполнение

Этап 1. Анализ трафика утилиты ping

Необходимо отследить и проанализировать трафик, создаваемый утилитой ping, запустив её следующим образом из командной строки:

```
"ping -l размер пакета адрес сайта по варианту".
```

В качестве "размера_пакета" необходимо поочерёдно использовать различные значения от 100 до 10000, самостоятельно выбрав шаг изменения.

for /L %i in (100, 500, 10000) do ping -l %i krivonosov.ru

```
C:\Users\Egor>ping -1 6600 krivonosov.ru

Обмен пакетами с krivonosov.ru [87.236.16.8] с 6600 байтами данных:
Ответ от 87.236.16.8: число байт=6600 время=20мс TTL=54

Статистика Ping для 87.236.16.8:
Пакетов: отправлено = 4, получено = 4, потеряно = 0
(0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 20мсек, Максимальное = 20 мсек, Среднее = 20 мсек
```

Вопросы

1. Имеет ли место фрагментация исходного пакета, какое поле на это указывает?

Да, так как с увеличением размера пакета, мы можем видеть, что на уровне IPпротокола появляются фрагменты "Fragmented IP protocol". Флаг More Fragments как раз указывает на наличие фрагментации исходного пакета.

```
Protocol Length Info
                                                                                                                                                                              Signal Stranght
     303 8.846657
                         87.236.16.8
                                                                                   1142 Echo (ping) reply
                                                                                                                 id=0x0001, seq=97/24832, ttl=54 (request .
    327 9.830047
328 9.849898
                                                                                   1142 Echo (ping) request id=0x0001, seq=98/25088, ttl=128 (reply i... 1142 Echo (ping) reply id=0x0001, seq=98/25088, ttl=54 (request ...
                         192.168.0.143
                                                 87.236.16.8
    368 10.833682
                        192,168,0,143
                                                 87.236.16.8
                                                                         TCMP
                                                                                    1142 Echo (ping) request id=0x0001, seq=99/25344, ttl=128 (reply i...
                                                                                    1142 Echo (ping) reply
                                                                                                                 id=0x0001, seq=99/25344, ttl=54 (request ...
     369 10.853558
                                                 192.168.0.143
                                                                                   1142 Echo (ping) request id=0x0001, seq=100/25600, ttl=128 (reply ...
1142 Echo (ping) reply id=0x0001, seq=100/25600, ttl=54 (request...
                        192.168.0.143
    391 11.837543
                                                 87.236.16.8
                                                                         ICMP
     396 11.857442
                                                 192.168.0.143
    397 11.878988
                        192,168,0,143
                                                 87,236,16,8
                                                                         IPv4
                                                                                   1506 Fragmented IP protocol (proto=ICMP 1, off=0, ID=0ad2) [Reassem...
                                                                                     170 Echo (ping) request
                                                                                                                 id=0x0001, seq=101/25856, ttl=128 (reply ...
                                                 192.168.0.143
                                                                                   1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8c57) [Reassem.
     400 11.899095
                         87.236.16.8
    401 11.899095
                         87.236.16.8
                                                 192.168.0.143
                                                                                    162 Echo (ping) reply id=0x0001, seq=101/25856, ttl=54 (request...
    453 12.881129
                         192.168.0.143
                                                 87.236.16.8
                                                                         IPv4
                                                                                   1506 Fragmented IP protocol (proto=ICMP 1, off=0, ID=0ad3) [Reassem.
                        192.168.0.143
    454 12.881129
                                                 87.236.16.8
                                                                                    170 Echo (ping) request id=0x0001, seq=102/26112, ttl=128 (reply ...
    455 12.901164
                                                 192.168.0.143
                                                                                   1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8d1b) [Reassem.
                         87.236.16.8
    456 12.901164
                        87.236.16.8
                                                 192.168.0.143
                                                                                   162 Echo (ping) reply id=0x0001, seq=102/26112, ttl=54 (request... 1506 Fragmented IP protocol (proto=ICMP 1, off=0, ID=0ad4) [Reassem...
    483 13.883563
                         192.168.0.143
                                                 87.236.16.8
                                                                         IPv4
    484 13.883563
                        192.168.0.143
                                                87.236.16.8
                                                                         TCMP
                                                                                    170 Echo (ping) request id=0x0001, seq=103/26368, ttl=128 (reply ...
    485 13.903699
                                                                                   1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8d9e) [Reassem...
                        87.236.16.8
                                                 192.168.0.143
    Destination: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4)
  > Source: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
     Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.0.143, Dst: 87.236.16.8
    0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 1492
Identification: 0x0ad2 (2770)
  > Flags: 0x20, More fragments
...0 0000 0000 0000 = Fragment Offset: 0
```

2. Какая информация указывает, является ли фрагмент пакета последним или промежуточным?

Установленный бит MF (More Fragments) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Соответственно, если флаг MF не установлен, то пакет является последним.

3. Чему равно количество фрагментов при передаче ping-пакетов?

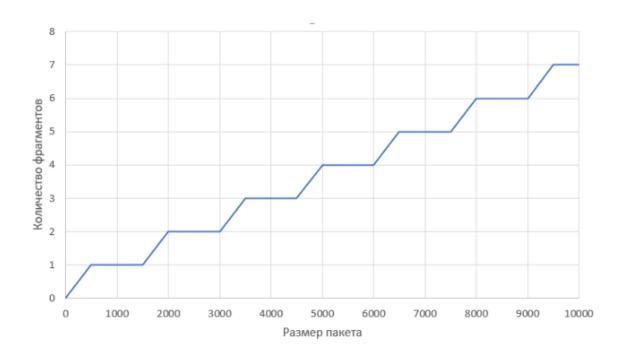
Количество фрагментов зависит от размера пакета и соответствующего соединению MTU (Maximum Transmission Unit – максимальный размер передаваемого блока), который обычно равен 1500 байт.

```
Total Length: 1500
```

Количество фрагментов = ceil(Длина сообщения (байт) / 1500 байт (фрагмент)) ceil() - округление вверх.

4. Построить график, в котором на оси абсцисс находится размер пакета, а по оси ординат – количество фрагментов, на которое был разделён каждый

ping-пакет.



5. Как изменить поле TTL с помощью утилиты ping?

Для изменения TTL нужно добавить ключ -i, его аргументом является срок жизни пакета в миллисекундах. (ping -i 15 krivonosov.ru)

6. Что содержится в поле данных ping-пакета?

В поле данных содержатся символы английского алфавита.

```
✓ Data (3600 bytes)

        Data: 6162636465666768696a6b6c6d6e6f70717273747576776162636465666768696a6b6c6d...
         [Length: 3600]
      00 00 c9 4b 00 01 00 77
                                  61 62 63 64 65 66 67 68
                                                                 ··K···w abcdefgh
0010
      69 6a 6b 6c 6d 6e 6f 70
                                  71 72 73 74 75 76 77 61
                                                                ijklmnop grstuvwa
      62 63 64 65 66 67 68 69  6a 6b 6c 6d 6e 6f 70 71
0020
                                                                bcdefghi jklmnopq
      72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 6a
                                                                rstuvwab cdefghij
      6b 6c 6d 6e 6f 70 71 72
                                 73 74 75 76 77 61 62 63
                                                                klmnopgr stuvwabo
0050
      64 65 66 67 68 69 6a 6b  6c 6d 6e 6f 70 71 72 73
                                                                defghijk lmnopqrs
      74 75 76 77 61 62 63 64 65 66 67 68 69 6a 6b 6c
0060
                                                                tuvwabcd efghijkl
      6d 6e 6f 70 71 72 73 74   75 76 77 61 62 63 64 65
0070
                                                                mnopqrst uvwabcde
0080
      66 67 68 69 6a 6b 6c 6d  6e 6f 70 71 72 73 74 75
                                                                fghijklm nopqrstu
      76 77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e
0090
                                                                vwabcdef ghijklmn
      6f 70 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70
00a0
                                                                opqrstuv wabcdefg
00b0
                                                                hijklmno pqrstuvw
                                                                abcdefgh ijklmnop
00c0
00d0
       71 72 73 74 75 76 77 61
                                 62 63 64 65 66 67 68 69
                                                                qrstuvwa bcdefghi
      6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61 62
63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72
00e0
                                                                jklmnopq rstuvwab
                                                                cdefghij klmnopqr
00f0
      73 74 75 76 77 61 62 63
                                                                stuvwabc defghij
0100
                                  64 65 66 67 68 69 6a 6b
          6d 6e 6f 70 71 72 73
                                  74 75 76 77 61 62 63 64
                                                                1mnopgrs tuvwabcd
0110
       6c
      65 66 67 68 69 6a 6b 6c
                                  6d 6e 6f 70 71 72 73 74
                                                                efghijkl mnopqrst
0120
0130
      75 76 77 61 62 63 64 65  66 67 68 69 6a 6b 6c 6d
                                                                uvwabcde fghijklm
```

Этап 2. Анализ трафика утилиты tracert (traceroute)

Необходимо отследить и проанализировать трафик, создаваемый утилитой tracert (или traceroute в Linux), запустив её следующим образом из командной строки:

"tracert -d адрес сайта по варианту"

```
C:\Users\Egor>tracert -d krivonosov.ru
Трассировка маршрута к krivonosov.ru [87.236.16.8]
 максимальным числом прыжков 30:
       <1 MC
                 <1 MC
                           <1 Mc 192.168.0.1
                           <1 MC 185.102.9.161
<1 MC 172.29.192.136
 2
3
       <1 MC
                 <1 MC
       <1 MC
                 <1 MC
                           <1 MC 172.29.193.253
 4
5
       <1 MC
                 <1 MC
       <1 MC
                 <1 MC
                           <1 mc 172.29.194.177
                           <1 MC 172.29.194.167
<1 MC 172.29.194.184
 6
7
8
                 <1 MC
       <1 MC
        1 ms
                 <1 MC
                           <1 mc 85.114.1.12
        1 ms
                 <1 MC
        1 ms
                 1 ms
                           1 ms 85.114.3.53
                           1 ms 85.114.3.56
1 ms 87.245.250.65
 10
        1 ms
                  1 ms
        4 ms
                  1 ms
12
                 17 ms
                           11 ms 87.245.233.89
       11 ms
       10 ms
                 10 ms
                            9 ms 87.245.229.69
 14
                                   Превышен интервал ожидания для запроса.
       19 ms
                 19 ms
                           19 ms 87.236.16.8
Грассировка завершена.
```

Вопросы

1. Сколько байт содержится в заголовке IP? Сколько байт содержится в поле данных?

В заголовке - 20 байт. В поле данных - 64 байта.

```
> Ethernet II, Src: Giga-Byt d8:9b:31 (18:c0:4d:d8:9b:31), Dst: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4)
Internet Protocol Version 4, Src: 192.168.0.143, Dst: 87.236.16.8
              = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 92
     Identification: 0x0b16 (2838)
  > Flags: 0x00
     ...0 0000 0000 0000 = Fragment Offset: 0
  > Time to Live: 1
     Protocol: ICMP (1)
     Header Checksum: 0x0000 [validation disabled]
     [Header checksum status: Unverified]
     Source Address: 192.168.0.143
     Destination Address: 87.236.16.8

▼ Internet Control Message Protocol

     Type: 8 (Echo (ping) request)
     Code: 0
     Checksum: 0xf755 [correct]
     [Checksum Status: Good]
     Identifier (BE): 1 (0x0001)
     Identifier (LE): 256 (0x0100)
     Sequence Number (BE): 169 (0x00a9)
     Sequence Number (LE): 43264 (0xa900)
    Data (64 bytes)
```

2. Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracert?

Утилита tracert отправляет первый пакет с TTL равным 1 и увеличивает значение TTL на 1 для каждого последующего отправляемого пока назначение не ответит или пока не будет достигнуто максимальное значение поля TTL. Поскольку каждый маршрутизатор на пути обязан уменьшить значение поля TTL пакета, по крайней мере на 1 перед дальнейшей пересылкой пакета, значение TTL по сути является эффективным счетчиком переходов.

```
106 Echo (ping) request id=0x0001, seq=171/43776, ttl=1 (no respo-
                                                                          134 Time-to-live exceeded (Time to live exceeded in transit)
                      192.168.0.1
     172 4.025229
                      185.102.9.161
                                           192.168.0.143
                                                                           70 Time-to-live exceeded (Time to live exceeded in transit)
                       192.168.0.143
                                                                          106 Echo (ping) request id=0x0001, seq=173/44288, ttl=2 (no
                                           87.236.16.8
                      185.102.9.161
                                           192.168.0.143
                                                                          70 Time-to-live exceeded (Time to live exceeded in transit)
     175 4 028624
                      192 168 0 143
                                           87.236.16.8
                                                                          106 Echo (ping) request id=0x0001, seq=174/44544, ttl=2 (no respo
                      185.102.9.161
                                           192.168.0.143
                                                                          70 Time-to-live exceeded (Time to live exceeded in transit)
     218 5.032638
                      192.168.0.143
                                           87.236.16.8
                                                                TCMP
                                                                          106 Echo (ping) request id=0x0001, seg=175/44800, ttl=3 (no respo
                                                                           70 Time-to-live exceeded (Time to live exceeded in transit)
    219 5.033493
                      172.29.192.136
                                           192.168.0.143
     220 5.034150
                      192.168.0.143
                                           87.236.16.8
                                                                ICMP
                                                                          106 Echo (ping) request id=0x0001, seq=176/45056, ttl=3 (no respo.
                                                                           70 Time-to-live exceeded (Time to live exceeded in transit)
    221 5.034793
                      172,29,192,136
                                           192.168.0.143
                                                                ICMP
     222 5.035740
                      192.168.0.143
                                           87.236.16.8
                                                                ICMP
                                                                          106 Echo (ping) request id=0x0001, seq=177/45312, ttl=3 (no respo
                                           192.168.0.143
                                                                           70 Time-to-live exceeded (Time to live exceeded in transit)
     223 5.036385
                      172.29.192.136
  Frame 118: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{89F00A98-943E-466C-B0D3-98F5C490689F}, id 0
  Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: la:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4)

▼ Internet Protocol Version 4, Src: 192.168.0.143, Dst: 87.236.16.8

     0100 .... = Version: 4
       ... 0101 = Header Length: 20 bytes (5)
   > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 92
     Identification: 0x0b18 (2840)
   > Flags: 0x00
      ...0 0000 0000 0000 = Fragment Offset: 0
   > Time to Live: 1
    114 3.021535
                     192.168.0.143
                                           87.236.16.8
                                                                          106 Echo (ping) request id=0x0001, seq=169/43264, ttl=1 (no respo
                                                                          134 Time-to-live exceeded (Time to live exceeded in transit)
                      192.168.0.1
                                           192.168.0.143
                                                                          106 Echo (ping) request id=0x0001, seq=170/43520, ttl=1 (no
    117 3.022211
                      192.168.0.1
                                           192.168.0.143
                                                                          134 Time-to-live exceeded (Time to live exceeded in transit)
                                                                          134 Time-to-live exceeded (Time to live exceeded in transit)
                      192.168.0.1
                                           192.168.0.143
    171 4.024367
                     192.168.0.143
                                           87.236.16.8
                                                                          106 Echo (ping) request id=0x0001, seq=172/44032, ttl=2 (no
    172 4.025229
                                                                           70 Time-to-live exceeded (Time to live exceeded in transit)
                      185.102.9.161
                                           192.168.0.143
                     192,168,0,143
                                                                          106 Echo (ping) request id=0x0001, seq=173/44288, ttl=2 (no res
                      185.102.9.161
                                           192.168.0.143
                                                                           70 Time-to-live exceeded (Time to live exceeded in transit)
    175 4.028624
                     192.168.0.143
                                           87,236,16,8
                                                                         106 Echo (ping) request id=0x0001, seq=174/44544, ttl=2 (no resp
                                           192.168.0.143
    176 4.029261
                      185.102.9.161
                                                                           70 Time-to-live exceeded (Time to live exceeded in transit)
    218 5.032638
                     192.168.0.143
                                           87.236.16.8
                                                                ICMP
                                                                          106 Echo (ping) request id=0x0001, seq=175/44800, ttl=3 (no response
                      172.29.192.136
                                           192.168.0.14
                                                                          70 Time-to-live exceeded (Time to live exceeded in transit)
    220 5.034150
                    192.168.0.143
                                          87.236.16.8
                                                                ICMP 106 Echo (ping) request id=0x0001, seq=176/45056, ttl=3 (no respo.
                      172.29.192.136
                                                                          70 Time-to-live exceeded (Time to live exceeded in transit)
    222 5.035740
                    192.168.0.143
                                          87.236.16.8
                                                                ICMP 106 Echo (ping) request id=0x0001, seq=177/45312, ttl=3 (no respo.
    223 5.036385
                     172.29.192.136
                                          192.168.0.143
                                                                ICMP
                                                                          70 Time-to-live exceeded (Time to live exceeded in transit)
 Frame 171: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{89F00A98-943E-466C-B0D3-98F5C490689F}, id 0
  Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4)
▼ Internet Protocol Version 4, Src: 192.168.0.143, Dst: 87.236.16.8
    0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 92
    Identification: 0x0b19 (2841)
  > Flags: 0x00
     ...0 0000 0000 0000 = Fragment Offset: 0
  > Time to Live: 2
```

3. Чем отличаются ICMP-пакеты, генерируемые утилитой tracert, от ICMPпакетов, генерируемых утилитой ping (см. предыдущее задание).

В отличие от пакетов, генерируемых утилитой ping, пакеты, генерируемые утилитой tracert, в поле данных содержат нули.

4. Чем отличаются полученные пакеты «ICMP reply» от «ICMP error» и зачем нужны оба этих типа ответов?

Пакеты «ICMP reply» указывают на получение ответного сообщения. Пакеты «ICMP error» указывают на то, что произошла ошибка. Они используются чтобы различать причину истечения TTL.

Значение поля TYPE = 0 соответствует сообщению Echo Reply

```
Tinternet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xff2a [correct]
    [Checksum Status: Good]
```

Значение поля TYPE = 11 соответствует сообщению Time Exceeded (истекло время)

```
V Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xf4ff [correct]
    [Checksum Status: Good]
```

5. Что изменится в работе tracert, если убрать ключ "-d"? Какой дополнительный трафик при этом будет генерироваться?

Ключ -d предотвращает попытки команды tracert разрешения IP-адресов промежуточных маршрутизаторов в имена, то есть в выводе будут отсутствовать имена хостов, через которые проходит IP-пакет.

Этап 3. Анализ НТТР-трафика

Необходимо отследить и проанализировать HTTP-трафик, создаваемый браузером при посещении Интернет-сайта, заданного по варианту. В списке захваченных пакетов необходимо проанализировать следующую пару HTTP-сообщений (запрос-ответ):

- GET-сообщение от клиента (браузера);
- ответ сервера.

Для этого в поле с детальной информацией о пакете нужно развернуть строку "HTTP". Затем необходимо обновить страницу в браузере так, чтобы вместо «HTTP GET» был сгенерирован «HTTP CONDITIONAL GET» (так называемый «условный GET»). Условные запросы GET содержат поля If-Modified-Since, If-Match, If-Range и подобные, которые позволяют при повторном запросе не передавать редко изменяемые данные. В ответ на условный GET тело запрашиваемого ресурса передается только в том случае, если этот ресурс изменялся после даты «If-Modified- Since». Если ресурс не изменялся, сервер вернет код статуса «304 Not Modified».

Вопросы

По результатам анализа собранной трассы покажите, каким образом протокол HTTP передавал содержимое страницы при первичном посещении страницы и при вторичном запросе-обновлении от браузера (т.е. при различных видах GET-запросов).

Так как сайт, выбранный в нашем варианте, имеет динамическое содержимое, как и большинство современных сайтов, использующих реактивные технологии, то заголовка not modified там быть не может (только статические страницы/файлы могут его иметь). Также HTTPS имеет сайт из-за этого HTML поля нет т.к. передача происходит по TCP. Поэтому рассмотрим на примере другого сайта.

	ip.addr ==87.236.16.8						
No.		Time	Source	Destination	Protocol Le		
	418	5.236135	87.236.16.8	192.168.0.143	TCP		
	419	5.236135	87.236.16.8	192.168.0.143	TCP		
	420	5.236153	192.168.0.143	87.236.16.8	TCP		
	421	5.236211	87.236.16.8	192.168.0.143	TCP		
	422	5.236211	87.236.16.8	192.168.0.143	TCP		
	423	5.236225	192.168.0.143	87.236.16.8	TCP		
	424	5.236328	87.236.16.8	192.168.0.143	TCP		
	425	5.236328	87.236.16.8	192.168.0.143	TLSv1.3		
	426	5.236342	192.168.0.143	87.236.16.8	TCP		
	429	5.245798	87.236.16.8	192.168.0.143	TLSv1.3		
	430	5.288140	192.168.0.143	87.236.16.8	TCP		
	484	5.429730	192.168.0.143	87.236.16.8	TLSv1.3		
	494	5.447861	87.236.16.8	192.168.0.143	TCP		
	620	5.793710	87.236.16.8	192.168.0.143	TCP		
	621	5.793768	87.236.16.8	192.168.0.143	TCP		
	622	5.793768	87.236.16.8	192.168.0.143	TLSv1.3		
L	623	5.793789	192.168.0.143	87.236.16.8	TCP		

```
46382 4042.41366 192.168.1.102
                                     184.51.233.240
                                                        HTTP
                                                               GET / HTTP/1.1
                                                               HTTP/1.1 304 Not Modified
 46383 4042.46925 184.51.233.240
                                     192.168.1.102
                                                        HTTP

⊟ Hypertext Transfer Protocol

 GET / HTTP/1.1\r\n

⊕ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]

     Request Method: GET
     Request URI: /
     Request Version: HTTP/1.1
   Cache-Control: max-age = 3600\r\n
   Connection: Keep-Alive\r\n
   Accept: */*\r\n
   If-Modified-Since: Mon, 26 Jul 2021 16:20:55 GMT\r\n
   If-None-Match: "60fee0e7-2cd"\r\n
   User-Agent: Microsoft-CryptoAPI/10.0\r\n
   Host: x1.c.lencr.org\r\n
    \r\n
```

```
Internet Protocol, Src: 184.51.233.240 (184.51.233.240), Dst: 192.168.1.102 (192.168.1.102)
⊕ Transmission Control Protocol, Src Port: http (80), Dst Port: 57927 (57927), Seq: 1, Ack: 228, L
Hypertext Transfer Protocol
 □ HTTP/1.1 304 Not Modified\r\n

☐ [Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]

        [Message: HTTP/1.1 304 Not Modified\r\n]
        [Severity level: Chat]
       [Group: Sequence]
     Request Version: HTTP/1.1
     Response Code: 304
   Content-Type: application/pkix-crl\r\n
   Last-Modified: Mon, 26 Jul 2021 16:20:55 GMT\r\n
   ETag: "60fee0e7-2cd"\r\n
   cache-control: max-age=3600\r\n
   Expires: Thu, 05 May 2022 15:45:44 GMT\r\n
   Date: Thu, 05 May 2022 14:45:44 GMT\r\n
   Connection: keep-alive\r\n
   r\n
```

Заголовок «HTTP/1.1 304 Not Modified» означает, что не нужно отправлять тело вместе с ним (таким образом экономится пропускная способность). Как видно в 1 запросе, установлен Cache-Control (в секундах), соответственно содержимое страницы кэшируется и нет необходимости отправлять его снова, если изменений с последней загрузки не произошло.

Этап 4. Анализ DNS-трафика

Необходимо отследить и проанализировать трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

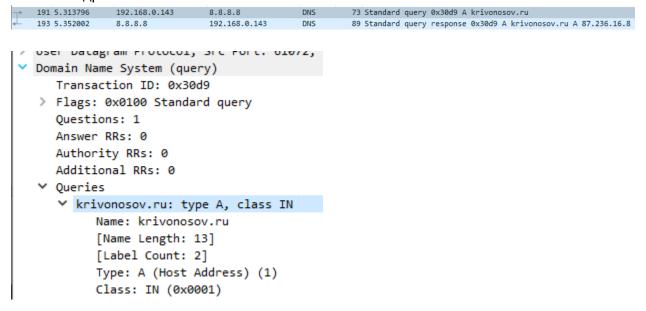
- настроить Wireshark-фильтр: "ip.addr == ваш_IP_адрес";
- очистить кэш DNS с помощью команды ipconfig в командной строке: ipconfig /flushdns
- очистить кэш браузера;
- зайти на Интернет-сайт, заданный по варианту.

```
C:\Users\Egor>ipconfig /flushdns
Настройка протокола IP для Windows
Кэш сопоставителя DNS успешно очищен.
```

Вопросы

1. Почему адрес, на который отправлен DNS-запрос, не совпадает с адресом посещаемого сайта?

Так как только что был очищен кэш, необходимого получить с DNS-сервера адрес запрашиваемого сайта. Там будет найдено совпадение доменного имени и сетевого адреса.



2. Какие бывают типы DNS-запросов?

Прямой: преобразование домена в IP-адрес.

Обратный: преобразование ІР-адреса в домен.

Рекурсивный: выполняется DNS-сервером, пока не будет найден домен или не будет получен ответ, что домен не существует. Рекурсия выполняется сервером. **Итеративный:** то же самое, что рекурсивный, но также допускается выполнение поиска клиентом.

3. В какой ситуации нужно выполнять независимые DNS-запросы для получения содержащихся на сайте изображений?

Если запрашиваемое изображение находится на другом сервере. Такое может

Этап 5. Анализ ARP-трафика

Необходимо отследить и проанализировать трафик протокола ARP, сгенерированный в результате выполнения следующих действий:

- очистить ARP-таблицу командой "netsh interface ip delete arpcache" (проверить очистилась ли таблица можно с помощью команды "arp -a", выводящей таблицу на экран);
- очистить кэш браузера;
- зайти на Интернет-сайт, заданный по варианту.

```
C:\Users\Egor>netsh interface ip delete arpcache
C:\Users\Egor>arp -a
Интерфейс: 192.168.0.143 --- 0х8
  адрес в Интернете Физический адрес
192.168.0.1 1a-e8-29-c3-c7-c4
                                             Тип
 динамический
                                            динамический
                                            статический
                                            статический
                     01-00-5e-00-00-16
01-00-5e-00-00-fb
 224.0.0.22
                                            статический
  224.0.0.251
                                            статический
 нтерфейс: 172.18.85.81 --- 0х13
  адрес в Интернете Физический адрес
                                             Тип
  172.18.85.95
                      ff-ff-ff-ff-ff
                                            статический
                     01-00-5e-00-00-16
 224.0.0.22
                                            статический
                   01-00-5e-00-00-fb
  224.0.0.251
                                            статический
  \Users\Egor>
```

Вопросы

1. Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что означают эти адреса? Какие устройства они идентифицируют?

```
5965 14.705508 ASUSTekC da:14:25 Broadcast
                                                                     60 Who has 192.168.0.2? Tell 192.168.0.179
                                                         ARP
ARP
ARP
  42 Who has 192.168.0.2? Tell 192.168.0.143
                                                                        60 Who has 192.168.0.2? Tell 192.168.0.179
 6327 23.287481 Giga-Byt_d8:9b:31 Broadcast
                                                                      42 Who has 192.168.0.2? Tell 192.168.0.143
 6366 24.050260 ASUSTekC_da:14:25 Broadcast
6368 24.056565 Giga-Byt_d8:9b:31 Broadcast
6410 24.705195 ASUSTekC_da:14:25 Broadcast
                                                           ARP
                                                                     60 Who has 192.168.0.2? Tell 192.168.0.179
                                                           ARP
                                                                        42 Who has 192.168.0.2? Tell 192.168.0.143
                                                            ARP
                                                                        60 Who has 192.168.0.2? Tell 192.168.0.179
6422 25.056240 Giga-Byt_d8:9b:31 Broadcast
                                                         ARP 42 Who has 192.168.0.2? Tell 192.168.0.143
 6448 25.705281 ASUSTekC_da:14:25 Broadcast ARP
6838 31.465830 1a:e8:29:c3:c7:c4 Giga-Byt_d8:9b:31 ARP
                                                                        60 Who has 192.168.0.2? Tell 192.168.0.179
                                                                        60 Who has 192.168.0.143? Tell 192.168.0.1
6839 31.465840 Giga-Byt_d8:9b:31 1a:e8:29:c3:c7:c4 ARP 42 192.168.0.143 is at 18:c0:4d:d8:9b:31
```

```
Frame 6838: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{89F00A98-943E-466C-B0D3-98F5C490689F}, id 0
 > Ethernet II, Src: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4), Dst: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)

✓ Address Resolution Protocol (request)

      Hardware type: Ethernet (1)
      Protocol type: IPv4 (0x0800)
      Hardware size: 6
      Protocol size: 4
      Opcode: request (1)
      Sender MAC address: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4)
      Sender IP address: 192.168.0.1
      Target MAC address: 00:00:00 00:00:00 (00:00:00:00:00:00)
      Target IP address: 192.168.0.143
 > Frame 6839: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{89F00A98-943E-466C-80D3-98F5C490689F}, id 0
 > Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4)

✓ Address Resolution Protocol (reply)

      Hardware type: Ethernet (1)
      Protocol type: IPv4 (0x0800)
      Hardware size: 6
      Protocol size: 4
      Opcode: reply (2)
      Sender MAC address: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
      Sender TP address: 192,168,0,143
      Target MAC address: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4)
      Target IP address: 192.168.0.1
1a:e8:29:c3:c7:c4 - адрес отправителя (маршрутизатор) (ip = 192.168.0.1)
```

1a:e8:29:c3:c7:c4 - адрес отправителя (маршрутизатор) (ip = 192.168.0.1) **18:c0:4d:d8:9b:31** - адрес устройства получателя (компьютер, с которого производится запрос на сайт) (ip = 192.168.0.143) **00:00:00:00:00:00** - broadcast-agpec

2. Какие MAC-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Что означают эти адреса? Какие устройства они идентифицируют?

В HTTP-пакетах присутствуют те же самые MAC-адреса, что и в ARP запросе. Они используются для идентификации отправителя и получателя HTTP-пакета.

3. Для чего ARP-запрос содержит IP-адрес источника?

Чтобы узел-получатель мог добавить информацию об узле-отправителе в свою ARP- таблицу.

Этап 6. Анализ трафика утилиты nslookup

Необходимо отследить и проанализировать трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

- Настроить Wireshark-фильтр: "ip.addr == ваш IP адрес".
- Запустить в командной строке команду "nslookup адрес_сайта_по_варианту".
- Дождаться отправки трёх DNS-запросов и трёх DNS-ответов (в работе нужно использовать только последние из них, т.к. первые два набора запросов/ответов специфичны для nslookup и не генерируются другими сетевыми приложениями).

• Повторить предыдущие два шага, используя команду: "nslookup -type=NS имя сайта по варианту".

```
C:\Users\Egor>nslookup krivonosov.ru

¬xЁтхЁ: dns.google
Address: 8.8.8.8

Не заслуживающий доверия ответ:

Ь: krivonosov.ru
Address: 87.236.16.8

C:\Users\Egor>nslookup -type=NS krivonosov.ru

¬xЁтхЁ: dns.google
Address: 8.8.8.8

Не заслуживающий доверия ответ:
krivonosov.ru nameserver = ns1.beget.com
krivonosov.ru nameserver = ns2.beget.com
krivonosov.ru nameserver = ns1.beget.pro
krivonosov.ru nameserver = ns2.beget.pro
krivonosov.ru nameserver = ns2.beget.ru
krivonosov.ru nameserver = ns1.beget.ru
krivonosov.ru nameserver = ns2.beget.ru
```

Вопросы

1. Чем различается трасса трафика в п.2 и п.4, указанных выше?

```
94 2.339430
                       192.168.0.143
                                            8.8.8.8
                                                                  DNS
                                                                             73 Stand
     100 2.368881
                       8.8.8.8
                                            192.168.0.143
                                                                  DNS
                                                                            133 Stand
    2842 62.987333
                       192.168.0.143
                                                                  DNS
                                                                             80 Stand
                                            8.8.8.8
    2843 62.992276
                                                                  DNS
                                                                            104 Stand
                       8.8.8.8
                                            192.168.0.143
                       192.168.0.143
    2844 62.992832
                                            8.8.8.8
                                                                  DNS
                                                                             73 Stand
    2845 63.023106
                       8.8.8.8
                                            192.168.0.143
                                                                  DNS
                                                                            205 Stand
> Frame 94: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface
Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: 1a:e8:29:c3:c7:c4 (
> Internet Protocol Version 4, Src: 192.168.0.143, Dst: 8.8.8.8
> User Datagram Protocol, Src Port: 59232, Dst Port: 53

✓ Domain Name System (query)

     Transaction ID: 0x0003
   > Flags: 0x0100 Standard query
     Questions: 1
     Answer RRs: 0
     Authority RRs: 0
     Additional RRs: 0

✓ Oueries

      krivonosov.ru: type AAAA, class IN
           Name: krivonosov.ru
           [Name Length: 13]
           [Label Count: 2]
           Type: AAAA (IPv6 Address) (28)
           Class: IN (0x0001)
     100 2.368881
                                                                            133 St
                       8.8.8.8
                                            192.168.0.143
                                                                 DNS
    2842 62.987333
                       192.168.0.143
                                            8.8.8.8
                                                                 DNS
                                                                             80 St
    2843 62.992276
                       8.8.8.8
                                            192.168.0.143
                                                                 DNS
                                                                            104 St
    2844 62.992832
                       192.168.0.143
                                            8.8.8.8
                                                                 DNS
                                                                             73 St
    2845 63.023106
                       8.8.8.8
                                            192.168.0.143
                                                                 DNS
                                                                            205 St
> Frame 2844: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on inter
> Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: 1a:e8:29:c3:c7:c
> Internet Protocol Version 4, Src: 192.168.0.143, Dst: 8.8.8.8
> User Datagram Protocol, Src Port: 53177, Dst Port: 53

✓ Domain Name System (query)

     Transaction ID: 0x0002
   > Flags: 0x0100 Standard query
     Questions: 1
     Answer RRs: 0
     Authority RRs: 0
     Additional RRs: 0
   Queries
     krivonosov.ru: type NS, class IN
           Name: krivonosov.ru
           [Name Length: 13]
           [Label Count: 2]
           Type: NS (authoritative Name Server) (2)
           Class: IN (0x0001)
```

При запуске в п.2 утилита ищет IP-адрес хоста (запись типа A (IPv4) или AAAA (IPv6)).

При запуске в п.4 утилита ищет Name Server для запрашиваемого хоста.

2. Что содержится в поле «Answers» DNS-ответа?

Данные запрашиваемого типа DNS-записи: для A - IPv4-адрес, для NS - список authoritative Name Server.

```
Queries

✓ krivonosov.ru: type NS, class IN
         Name: krivonosov.ru
         [Name Length: 13]
         [Label Count: 2]
         Type: NS (authoritative Name Server) (2)
         Class: IN (0x0001)
 Answers
    Name: krivonosov.ru
         Type: NS (authoritative Name Server) (2)
         Class: IN (0x0001)
         Time to live: 300 (5 minutes)
         Data length: 15
         Name Server: ns1.beget.com

▼ krivonosov.ru: type NS, class IN, ns ns2.beget.com
         Name: krivonosov.ru
         Type: NS (authoritative Name Server) (2)
         Class: IN (0x0001)
        Time to live: 300 (5 minutes)
         Data length: 6
         Name Server: ns2.beget.com

▼ krivonosov.ru: type NS, class IN, ns ns1.beget.pro

         Name: krivonosov.ru
         Type: NS (authoritative Name Server) (2)
         Class: IN (0x0001)
         Time to live: 300 (5 minutes)
         Data length: 15
         Name Server: ns1.beget.pro
NAME — имя хоста.
ТҮРЕ — тип ресурсной записи. Определяет формат и назначение данной
CLASS — класс ресурсной записи. Обычно IN для Internet (Код 0х0001)
```

ресурсной записи.

TTL — (Time To Live) — допустимое время хранения данной ресурсной записи в кэше неответственного DNS-сервера.

RDLENGTH — длина поля данных.

RDATA — поле данных, формат и содержание которого зависит от типа записи.

3. Каковы имена серверов, возвращающих авторитативный (authoritative) отклик?

```
Answers
  krivonosov.ru: type NS, class IN, ns ns1.beget.com
  > krivonosov.ru: type NS, class IN, ns ns2.beget.com
  > krivonosov.ru: type NS, class IN, ns ns1.beget.pro
  > krivonosov.ru: type NS, class IN, ns ns2.beget.pro
  > krivonosov.ru: type NS, class IN, ns ns1.beget.ru
  > krivonosov.ru: type NS, class IN, ns ns2.beget.ru
```

Авторитативный отклик возвращают серверы, которые являются ответственными за зону, в которой описана информация, необходимая DNS-клиенту.

Этап 7. Анализ FTP-трафика

Необходимо отследить и проанализировать трафик протокола FTP, сгенерированный в результате выполнения следующих действий:

- настроить Wireshark-фильтр «ftp || ftp-data»;
- скачать в браузере небольшой файл с соответствующего варианту FTP-сервера в Интернете.

Вопросы

1. Сколько байт данных содержится в пакете FTP-DATA?

Размер может быть любы, но не больше MTU. В данном случае 409 байт.

```
30143 405.908243 89.111.47.130
                                                                              463 FTP Data: 409 bytes (PASV) (LIST)
                                       192.168.0.143
                                                               FTP-DATA
  30149 405.921729 89.111.47.130
                                                                               78 Response: 226 Directory send OK.
                                          192.168.0.143
 Frame 30143: 463 bytes on wire (3704 bits), 463 bytes captured (3704 bits) on interface \Device\NPF {89F00A98-943E-466C-B0D3-98F
> Ethernet II, Src: 1a:e8:29:c3:c7:c4 (1a:e8:29:c3:c7:c4), Dst: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
> Internet Protocol Version 4, Src: 89.111.47.130, Dst: 192.168.0.143
> Transmission Control Protocol, Src Port: 50553, Dst Port: 56583, Seq: 1, Ack: 1, Len: 409
  FTP Data (409 bytes data)
  [Setup frame: 30135]
  [Setup method: PASV]
  [Command: LIST]
  Command frame: 30140
  [Current working directory: /]
> Line-based text data (6 lines)
```

2. Как выбирается порт транспортного уровня, который используется для передачи FTP-пакетов?

Для потока управления на сервере используется порт 21. Для передачи данных используется порт 20, если передача идет в активном режиме, либо с любого порта клиента к любому порту сервера в пассивном режиме.

```
▼ Transmission Control Protocol, Src Port: 21, Dst Port: 56640, Seq: 232, Ack: 58, Len: 24
     Source Port: 21
     Destination Port: 56640
     [Stream index: 8]
     [Conversation completeness: Incomplete, DATA (15)]
     [TCP Segment Len: 24]
     Sequence Number: 232 (relative sequence number)
     Sequence Number (raw): 3219732088
     [Next Sequence Number: 256 (relative sequence number)]
     Acknowledgment Number: 58 (relative ack number)
     Acknowledgment number (raw): 1227646999
     0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 83
     [Calculated window size: 42496]
     [Window size scaling factor: 512]
     Checksum: 0x1ac4 [unverified]
     [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
     TCP payload (24 bytes)
```

3. Чем отличаются пакеты FTP от FTP-DATA?

FTP используется для выполнения команд (request/response), а FTP-DATA работает с файлами.

Этап 8. Анализ DHCP-трафика

Необходимо отследить и проанализировать трафик протокола DHCP, сгенерированный в результате выполнения следующих действий:

- Убедиться, что для назначения IP-адреса на компьютере был использован DHCP и что компьютеру был назначен IP-адрес.
- Настроить Wireshark-фильтр «bootp» (во время защиты УИР следует объяснить, почему именно такой фильтр используется для анализа DHCP-трафика).
- Сбросить текущий IP-адрес, выданный накануне перед этим DHCP-сервером, с помощью команды: "ipconfig /release".
- Запросить новый IP-адрес с помощью команды: "ipconfig /renew".
- Повторить п.3 и п.4.

```
C:\Users\Egor>ipconfig /release
Настройка протокола IP для Windows
Адаптер Ethernet Ethernet 3:
  DNS-суффикс подключения . . . . :
  Локальный IPv6-адрес канала . . . : fe80::d9f5:8153:a96f:61aa%8
  Основной шлюз. . . . . . . . :
Адаптер Ethernet vEthernet (Default Switch):
  DNS-суффикс подключения . . . . :
  Локальный IPv6-адрес канала . . . : fe80::d0ee:6f32:6212:7a14%19
  IPv4-адрес. . . . . . . . . . . . : 172.18.85.81
  Маска подсети . . . . . . . . : 255.255.255.240
  Основной шлюз. . . . . . . . :
C:\Users\Egor>ipconfig /renew
Настройка протокола IP для Windows
Адаптер Ethernet Ethernet 3:
  DNS-суффикс подключения . . . . . :
  Локальный IPv6-адрес канала . . . : fe80::d9f5:8153:a96f:61aa%8
  IPv4-адрес. . . . . . . . . . : 192.168.0.143
  Основной шлюз. . . . . . . : 192.168.0.1
Адаптер Ethernet vEthernet (Default Switch):
  DNS-суффикс подключения . . . . :
  Локальный IPv6-адрес канала . . . : fe80::d0ee:6f32:6212:7a14%19
  Маска подсети . . . . . . . . : 255.255.255.240
  Основной шлюз. . . . . . . . . .
```

Вопросы

1. Чем различаются пакеты «DHCP Discover» и «DHCP Request»?

Оба запроса выполняются клиентом, DHCP Discover ищет DHCP-сервер в своей канальной среде, а DHCP Request принимает предлагаемый адрес и уведомляет DHCP-сервер об этом.

```
bootp
No.
                        Source
                                             Destination
                                                                    Protocol
                                                                                 Length Info
                                                                                                                                                           Char
                       192.168.0.143
     617 12.283284
                                             192.168.0.1
                                                                   DHCP
                                                                                   342 DHCP Release - Transaction ID 0xa4d8eaba
                                                                                   344 DHCP Discover - Transaction ID 0x58444234
     793 20.411248
                                             255.255.255.255
                                                                    DHCP
                       0.0.0.0
> Frame 793: 344 bytes on wire (2752 bits), 344 bytes captured (2752 bits) on interface \Device\NPF_{89F00A98-943E-466C-B0D3-98F5C490689F}, id 0
  Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: Broadcast (ff:ff:ff:ff:ff)
  Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 68, Dst Port: 67

→ Dynamic Host Configuration Protocol (Discover)

     Message type: Boot Request (1)
     Hardware type: Ethernet (0x01)
     Hardware address length: 6
     Hops: 0
     Transaction ID: 0x58444234
     Seconds elapsed: 0
     Bootp flags: 0x0000 (Unicast)
     Client IP address: 0.0.0.0
     Your (client) IP address: 0.0.0.0
     Next server IP address: 0.0.0.0
     Relay agent IP address: 0.0.0.0
     Client MAC address: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
     Client hardware address padding: 000000000000000000000
     Server host name not given
     Boot file name not given
     Magic cookie: DHCP
     Option: (53) DHCP Message Type (Discover)

→ Option: (61) Client identifier
        Length: 7
        Hardware type: Ethernet (0x01)
        Client MAC address: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
   > Option: (50) Requested IP Address (192.168.0.143)
     Option: (12) Host Name
     Option: (60) Vendor class identifier
     Option: (55) Parameter Request List
   > Option: (255) End
         Time
                                               Destination
                                                                      Protocol
                        Source
                                                                                    Length Info
```

```
bootp
No.
    617 12.283284
                    192.168.0.143
                                         192.168.0.1
                                                             DHCP
                                                                           342 DHCP Release - Transaction ID 0xa4d8eaba
                                                                           344 DHCP Discover - Transaction ID 0x58444234
    793 20.411248
                     0.0.0.0
                                         255.255.255.255
                                                             DHCP
> Frame 617: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface \Device\NPF_{89F00A98-943E-466C-B0D3-98F5C490689F}, id 0
  Ethernet II, Src: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31), Dst: la:e8:29:c3:c7:c4 (la:e8:29:c3:c7:c4)
  Internet Protocol Version 4, Src: 192.168.0.143, Dst: 192.168.0.1
> User Datagram Protocol, Src Port: 68, Dst Port: 67

▼ Dynamic Host Configuration Protocol (Release)

    Message type: Boot Request (1)
     Hardware type: Ethernet (0x01)
     Hardware address length: 6
     Hops: 0
     Transaction ID: 0xa4d8eaba
     Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
     Client IP address: 192.168.0.143
     Your (client) IP address: 0.0.0.0
     Next server IP address: 0.0.0.0
     Relay agent IP address: 0.0.0.0
     Client MAC address: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)
     Client hardware address padding: 00000000000000000000
     Server host name not given
     Boot file name not given
     Magic cookie: DHCP

✓ Option: (53) DHCP Message Type (Release)

       Length: 1
       DHCP: Release (7)

→ Option: (54) DHCP Server Identifier (192.168.0.1)
       Length: 4
       DHCP Server Identifier: 192.168.0.1

→ Option: (61) Client identifier

       Length: 7
       Hardware type: Ethernet (0x01)
       Client MAC address: Giga-Byt_d8:9b:31 (18:c0:4d:d8:9b:31)

✓ Option: (255) End

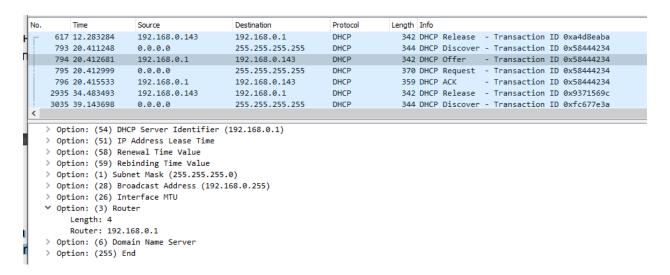
       Option End: 255
```

2. Как и почему менялись MAC- и IP-адреса источника и назначения в переданных DHCP-пакетах.

В качестве МАС-адреса источника клиент изначально подставил свой МАС-адрес, а МАС-адрес сервера он не знает, поэтому использует широковещательный МАС-адрес. Соответственно, в заголовке IP-пакета в качестве адреса источника клиент использовал 0.0.0.0. При отправке Offer или АСК пакетов, адреса источника соответствуют адресам DHCP-сервера, адреса назначения широковещательные.

3. Каков IP-адрес DHCP-сервера?

192.168.0.1 - адрес роутера



4. Что произойдёт, если очистить использованный фильтр "bootp"?

Будут отображаться все пакеты.

Этап 9. Анализ Telegram-трафика

Необходимо отследить и проанализировать трафик Telegram, сгенерированный в результате выполнения следующих действий:

- отправить текстовое сообщение и получить ответ;
- осуществить короткий сеанс аудио-общения;
- осуществить короткий сеанс видео-общения.

Вопросы

1. Чем различаются пакета разных видов Telegram-трафика (текст, аудио, видео)?

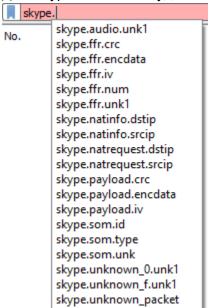
Текстовые данные передаются с помощью **TSL**При загрузке аудио и видео используются **TCP** и **SSL v2**Во время аудио-звонков используется **UDP**

No.	Time	Source	Destination	Protocol	Length Info
	8621 64.001151	149.154.167.41	192.168.0.143	TCP	60 443 → 51781 [ACK] Seq=1645 Ack=1966070 Win=65535 Len=0
	8622 64.001709	149.154.167.41	192.168.0.143	TCP	60 443 → 51781 [ACK] Seq=1645 Ack=1968550 Win=65535 Len=0
	8624 64.008901	162.159.130.234	192.168.0.143	TLSv1.2	211 Application Data
	8625 64.020964	162.159.130.234	192.168.0.143	TLSv1.2	97 Application Data
	8626 64.021000	192.168.0.143	162.159.130.234	TCP	54 60990 → 443 [ACK] Seq=55 Ack=189292 Win=1028 Len=0
	8627 64.032435	149.154.167.41	192.168.0.143	TCP	60 443 → 51781 [ACK] Seq=1645 Ack=1970962 Win=65535 Len=0
	8628 64.043270	149.154.167.41	192.168.0.143	TCP	143 443 → 51777 [PSH, ACK] Seq=1686 Ack=2102238 Win=65535 Len=89
	8629 64.045749	192.168.0.143	149.154.167.41	SSLv2	63294 Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Da
	8630 64.045781	192.168.0.143	149.154.167.41	TCP	1294 51777 → 443 [ACK] Seq=2165478 Ack=1775 Win=63768 Len=1240 [7
	8631 64.054320	162.159.130.234	192.168.0.143	TLSv1.2	226 Application Data
	8632 64.072042	149.154.167.41	192.168.0.143	TCP	143 [TCP segment of a reassembled PDU]
	8633 64.074213	192.168.0.143	149.154.167.41	SSLv2	63294 Encrypted Data, Encrypted Data, Encrypted Data, Encrypted Da
	8634 64.074244	192.168.0.143	149.154.167.41	TCP	1294 [TCP segment of a reassembled PDU]
	8635 64.079679	149.154.167.41	192.168.0.143	TCP	60 443 → 51777 [ACK] Seq=1775 Ack=2110918 Win=65535 Len=0
	8636 64.079749	192.168.0.143	149.154.167.41	SSLv2	8734 Encrypted Data, Encrypted Data, Encrypted Data
	8637 64.079902	149.154.167.41	192.168.0.143	TCP	60 443 → 51777 [ACK] Seq=1775 Ack=2118358 Win=65535 Len=0
	8638 64.079913	192.168.0.143	149.154.167.41	SSLv2	7494 Encrypted Data [TCP segment of a reassembled PDU]
	8639 64.080110	149.154.167.41	192.168.0.143	TCP	60 443 → 51777 [ACK] Seq=1775 Ack=2125798 Win=65535 Len=0
	8640 64.080118	192.168.0.143	149.154.167.41	SSLv2	7494 Encrypted Data [TCP segment of a reassembled PDU]
	8641 64.080445	149.154.167.41	192.168.0.143	TCP	60 443 → 51777 [ACK] Seq=1775 Ack=2131998 Win=65535 Len=0
	8642 64.080451	192.168.0.143	149.154.167.41	SSLv2	6254 Encrypted Data [TCP segment of a reassembled PDU]
	8643 64.080690	149.154.167.41	192.168.0.143	TCP	60 443 → 51777 [ACK] Seq=1775 Ack=2141918 Win=65535 Len=0
	8644 64.080695	192.168.0.143	149.154.167.41	TCP	9974 51777 → 443 [ACK] Seq=2196478 Ack=1775 Win=63768 Len=9920 [7
	8645 64.081134	149.154.167.41	192.168.0.143	TCP	60 443 → 51777 [ACK] Seq=1775 Ack=2153078 Win=65535 Len=0
	8646 64.081145	192.168.0.143	149.154.167.41	TCP	11214 51777 → 443 [ACK] Seq=2206398 Ack=1775 Win=63768 Len=11160 [
	8647 64.081495	149.154.167.41	192.168.0.143	TCP	60 443 → 51777 [ACK] Seq=1775 Ack=2160518 Win=65535 Len=0
	8648 64.081511	192.168.0.143	149.154.167.41	SSLv2	7494 Encrypted Data [TCP segment of a reassembled PDU]
	8649 64.081816	149.154.167.41	192.168.0.143	TCP	60 443 → 51777 [ACK] Seq=1775 Ack=2165478 Win=65535 Len=0
	8650 64.081830	192.168.0.143	149.154.167.41	TCP	5014 51777 → 443 [ACK] Seq=2224998 Ack=1775 Win=63768 Len=4960 [T
2					
Fr	rame 6258: 100 by	vtes on wire (800 bit	s), 100 bytes capture	ed (800 bits)	on interface \Device\NPF {89F00A98-943E-466C-B0D3-98F5C490689F}, id
					d8:9b:31 (18:c0:4d:d8:9b:31)
			73.100.196, Dst: 192		,
		•	rt: 443, Dst Port: 517		9, Ack: 1381, Len: 46
	Source Port: 44		,,		• • • • • • • • • • • • • • • • • • • •
	Destination Por				

2. Какой Wireshark-фильтр следует использовать для независимой идентификации Telegram-трафика разных видов (текст, аудио, видео)?

B Wireshark можно установить фильтр по протоколу, соответственно нужно установить tsl для текстовых данных, tcp для просмотра загрузки аудио/видео, udp для аудио.

Для Skype есть более удобные фильтры в wireshark

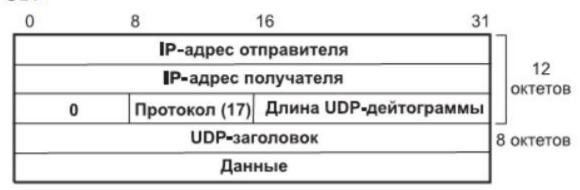


Структура наблюдаемых пакетов

ARP

0	8	16	24	31	
Ти	п оборудовани	Я	Тип протокола		
HA-Len	HA-Len PA-Len		Код операции		
	Аппаратный адрес отправителя (октеты 03)				
Адрес от	равителя (окте	ты 4,5) ІР-ад	ІР-адрес отправителя (октеты 0,1)		
ІР-адрес от	гправителя (окт	еты 2,3) Аппа	Аппаратный адрес адресата (0,1)		
Аппаратный адрес адресата (октеты 2,5)					
IP-адрес адресата (октеты 0-3)					

UDP



ICMP



DNS

Идентификатор	Флаги		
Количество вопросов	Количество ответных RR-записей	12 байт	
Количество RR-залисей полномочного источника	Количество дополнительных RR-записей		
Вог (количество вопр	Имя, поля типов для запроса		
От (количество RR-за	RR-записи в ответ на запрос		
Полномочн (количество RR-за	Записи для полномочного сервера		
Дополнительн (количество RR-эа	Дополнительная «полезная» информация		

DHCP

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31							
Op (1)	htype (1)	hlen (1)	Шаги (1)				
	xid (4)						
s	ecs (2)	Флаг	и (2)				
ciaddr (4)							
	yiaddr (4)						
	siaddr (4)						
	giaddr (4)						
	chaddr (16)						
sname (64)							
Файл (128)							
Опции (длина переменная)							

Вывод

В ходе лабораторной работы был проанализирован сетевой трафик с помощью программы Wireshark. Мы изучили, какие пакеты передаются при работе утилит ping, и tracert и какую информацию они содержат. Также был проведен анализ трафика HTTP-запросов и влияние на него кэширования данных. Кэширование также влияет на работу DNS, во время выполнения работы нам необходимо было очистить кэши и посмотреть на работу DNS-запросов. Далее был рассмотрен трафик при выполнении агр-запросов, для этого нужно было очистить агр-таблицу. Кроме того был проанализирован трафик при работе с FTP, самой сложной частью этого задания оказалось найти действующий ftp-сервер. Для анализа DHCP пришлось вспомнить 2 лабораторную и работу этого протокола, чтобы описать типы запросов и данные, содержащиеся в соответствующих пакетах. И последним был рассмотрен трафик Telegram, мы узнали, какие протоколы используются в нем для передачи разных типов данных.