Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Компьютерные сети

Лабораторная работа №3

Выполнил: Борисенко Е. А.

Группа: P33011

Преподаватель: Маркина Т. А.

г. Санкт-Петербург

2021

Цель работы: изучить структуру протокольных блоков данных, анализируя реальный трафик на компьютере студента с помощью бесплатно распространяемой утилиты Wireshark.

Сайт: dr-borisenko.ru

1. Анализ трафика утилиты ping

Структура PDU:

Ethernet II – длина заголовка = 14 байт

```
Ethernet II, Src: LiteonTe_b4:f8:65 (74:df:bf:b4:f8:65), Dst: NF
  Destination: NPKRotek_12:21:49 (dc:e3:05:12:21:49)
  Source: LiteonTe_b4:f8:65 (74:df:bf:b4:f8:65)
  Type: IPv4 (0x0800)
```

- Destination Address (6 байт) – MAC-адрес назначения
- Source Address (6 байт) – MAC-адрес источника
- Type (2 байт) – тип протокола определения адреса

IPv4 – длина заголовка = 20-60 байт

```
Internet Protocol Version 4, Src: 192.168.0.5, Dst: 31.31.198.59
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1228
  Identification: 0xd249 (53833)
  Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  Fragment Offset: 0
  Time to Live: 128
  Protocol: ICMP (1)
  Header Checksum: 0xbddf [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.0.5
  Destination Address: 31.31.198.59
```

- Version (нибл) – версия протокола (4 или 6)
- Header Length (нибл) – длина заголовка в 32-битных словах.
- Differentiated Services Field (1 байт) –
- Total Length (2 байт) – длина заголовка и данных в байтах
- Identification (2 байт) – идентификатор пакета, используемый для восстановления их порядка
- Flags (3 бита):
  - Reserved bit – зарезервирован, равен 0
  - DF (Don’t fragment) – если равен 0, то допускается фрагментация пакетов, если 1, то фрагментация не будет выполняться
  - MF (More fragments) – если равен 1, то после текущего пакета есть ещё, иначе пакет является последним.
- Fragment Offset (13 бит) – смещение относительно первого фрагмента
- Time to Live / TTL (1 байт) – время жизни пакета в хопах (максимальное количество пройденных узлов)
- Protocol (1 байт) – тип протокола транспортного уровня

- Header Checksum (2 байт) – контрольная сумма для заголовка
- Source Address (4 байт) – ip адрес отправителя
- Destination Address (4 байт) – ip адрес получателя

ICMP – длина заголовка = 8 байт

```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xed23 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 207 (0x00cf)
    Sequence Number (LE): 52992 (0xcf00)
```

- Type (1 байт) – число-тип сообщения ICMP (echo-request, echo-reply и т.п.)
- Code (1 байт) – дополнительная информация о типе сообщения
- Checksum (2 байт) – контрольная сумма
- Identifier (2 байт)
- Sequence Number (2 байт) – в последних 4 байтах предоставляется дополнительная информация, зависящая от типа сообщения

Вопросы:

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

Да, имеется фрагментация. Ее можно увидеть по флагу MF (More fragments), установленному в 1, по полям Total Length (меньше, чем количество отправленных байт) и Fragment Offset (не 0 для не первого пакета).

```
Internet Protocol Version 4, Src: 192.168.0.5, Dst: 31.31.198.59
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0xd213 (53779)
  v Flags: 0x20, More fragments
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..1. .... = More fragments: Set
    Fragment Offset: 0
    Time to Live: 128
Internet Protocol Version 4, Src: 192.168.0.5, Dst: 31.31.198.59
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 848
    Identification: 0xd213 (53779)
  v Flags: 0x00
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    Fragment Offset: 1480
    Time to Live: 128
```

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

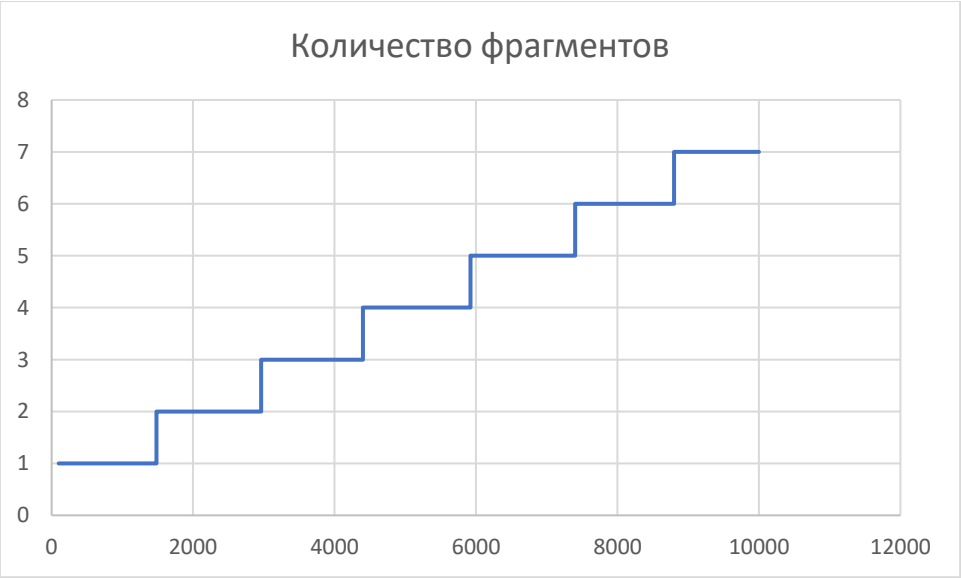
В предыдущем ответе взяты поля для исходного пакета размером 2300 байт (такой передается за 2 кадра). Если пакет первый или промежуточный, флаг MF равен 1, если он последний, то MF равен 0.

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

Количество_фрагментов = количество_пакетов/(размер_фрагмента-размер_заголовка-размер_инфы_фрагментации). В данном случае размер фрагмента=1500, размер заголовка=20 байт, размер информации о фрагментации=8 байт (хранится в последнем фрагменте).

```
Internet Protocol Version 4, Src: 192.168.0.5, Dst: 31.31.198.59
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1148
    Identification: 0xd232 (53810)
  v Flags: 0x04
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    Fragment Offset: 8880
    Time to Live: 128
    Protocol: ICMP (1)
    Header Checksum: 0xb9f0 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.0.5
    Destination Address: 31.31.198.59
  v [7 IPv4 Fragments (10008 bytes): #415(1480), #416(1480), #417(148
      [Frame: 415, payload: 0-1479 (1480 bytes)]
      [Frame: 416, payload: 1480-2959 (1480 bytes)]
      [Frame: 417, payload: 2960-4439 (1480 bytes)]
      [Frame: 418, payload: 4440-5919 (1480 bytes)]
      [Frame: 419, payload: 5920-7399 (1480 bytes)]
      [Frame: 420, payload: 7400-8879 (1480 bytes)]
      [Frame: 421, payload: 8880-10007 (1128 bytes)]
      [Fragment count: 7]
      [Reassembled IPv4 length: 10008]
      [Reassembled IPv4 data: 08006cb3000100b861626364656667686 96a6b
```

4. Построить график, в котором на оси абсцисс находится размер пакета, а по оси ординат – количество фрагментов, на которые был разделен каждый ping-пакет.



5. Как изменить поле TTL с помощью утилиты ping?

ping -i <TTL> - для Windows стандартно = 128.

6. Что содержится в поле данных ping-пакета?

```
0020  52 5c 08 00 6d 23 00 01  00 48 61 62 63 64 65 66   R\··m#·· ·Habcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69 6a 6b 6c 6d 6e 6f   wabcdefg hijklmno
0050  70 71 72 73 74 75 76 77  61 62 63 64 65 66 67 68   pqrstuvw abcdefgh
0060  69 6a 6b 6c 6d 6e 6f 70  71 72 73 74 75 76 77 61   ijklmnop qrstuvwa
```

Коды латинских букв от a до w

2. Анализ трафика утилиты tracert (traceroute)

Вопросы:

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

20 байт в заголовке; 64 байта в поле данных.

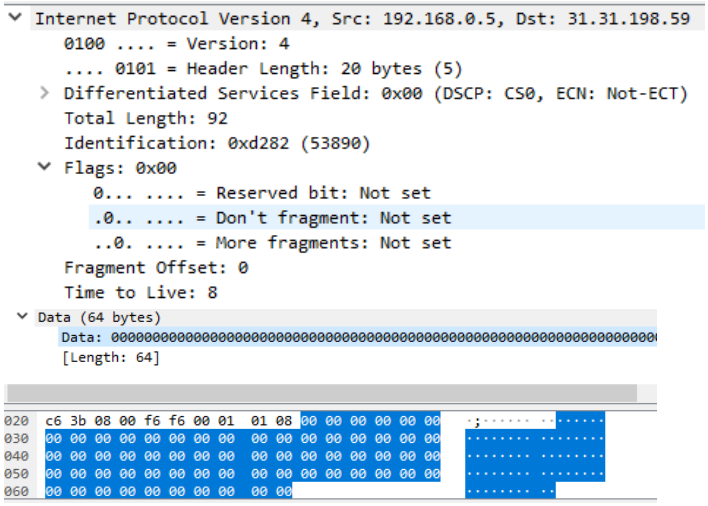
```
Internet Protocol Version 4, Src: 192.168.0.5, Dst: 31.31.198.59
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
   Total Length: 92
Data (64 bytes)
   Data: 00000000000000000000000000000000000000000000000000000000
   [Length: 64]
```

2. Как и почему изменяется поле TTL в следующих друг за другом ICMP-пакетах tracert? Для ответа на этот вопрос нужно проследить изменение TTL при передаче по маршруту, состоящему из более чем двух хопов.

При проходе по маршрутизаторам можно достичь максимально возможного числа хопов и последний маршрутизатор в цепочке сообщает об этом, посылая обратно отправителю сообщение об окончании жизни пакета (TTL=0). После этого TTL для повторной отправки увеличивается на 1.

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

Разные значения поля TTL (ping – стандартное для windows 128, tracert – постепенно увеличивающееся значение).
Данные ping заполняются кодами символов от “a” до “w”, данные tracert - нули.
Максимальная длина данных в пакете ping – 1472, tracert – 64.



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

ICMP-error имеет другой тип, неиспользованные 4 байта и содержит недошедший ICMP-request. ICMP-reply отправляется узлом в случае успешной доставки, ICMP-error – когда сообщение не дошло.

```
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xfef6 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 264 (0x0108)
    Sequence Number (LE): 2049 (0x0801)
    [Request frame: 50]
    [Response time: 16,447 ms]
  Data (64 bytes)
      Data: 0000000000000000000000000000000000000000
      [Length: 64]
```

```
Internet Control Message Protocol
  Type: 11 (Time-to-live exceeded)
  Code: 0 (Time to live exceeded in transit)
  Checksum: 0xf4ff [correct]
  [Checksum Status: Good]
  Unused: 00000000
Internet Protocol Version 4, Src: 192.168.0.5, Dst: 31.31.198.59
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 92
  Identification: 0xd26d (53869)
v Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  Fragment Offset: 0
> Time to Live: 1
  Protocol: ICMP (1)
  Header Checksum: 0x412c [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.0.5
  Destination Address: 31.31.198.59
> Internet Control Message Protocol
```

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

```
C:\Users\1>tracert -d dr-borisenko.ru

Трассировка маршрута к dr-borisenko.ru [31.31.198.59]
с максимальным числом прыжков 30:

  1     6 ms     1 ms     3 ms  192.168.0.1
  2     4 ms     3 ms     6 ms  10.189.204.1
  3     5 ms    12 ms     7 ms  5.18.4.250
  4     8 ms     6 ms     6 ms  178.18.224.152
  5    15 ms    14 ms    16 ms  178.18.226.192
  6     *        *        *     Превышен интервал ожидания для запроса.
  7    17 ms   180 ms    16 ms  172.16.71.11
  8    16 ms    16 ms    17 ms  31.31.198.59
```

```
C:\Users\1>tracert dr-borisenko.ru

Трассировка маршрута к dr-borisenko.ru [31.31.198.59]
с максимальным числом прыжков 30:

  1     6 ms     1 ms     3 ms  192.168.0.1
  2    23 ms     5 ms     4 ms  10.189.204.1
  3     5 ms     5 ms   113 ms  5.18.4.250
  4    12 ms     6 ms     6 ms  as9049.ix.dataix.ru [178.18.224.152]
  5    22 ms    19 ms    16 ms  as39134.ix.dataix.ru [178.18.226.192]
  6     *        *        *     Превышен интервал ожидания для запроса.
  7    22 ms    16 ms   120 ms  172.16.71.11
  8     *      211 ms    21 ms  server252.hosting.reg.ru [31.31.198.59]
```

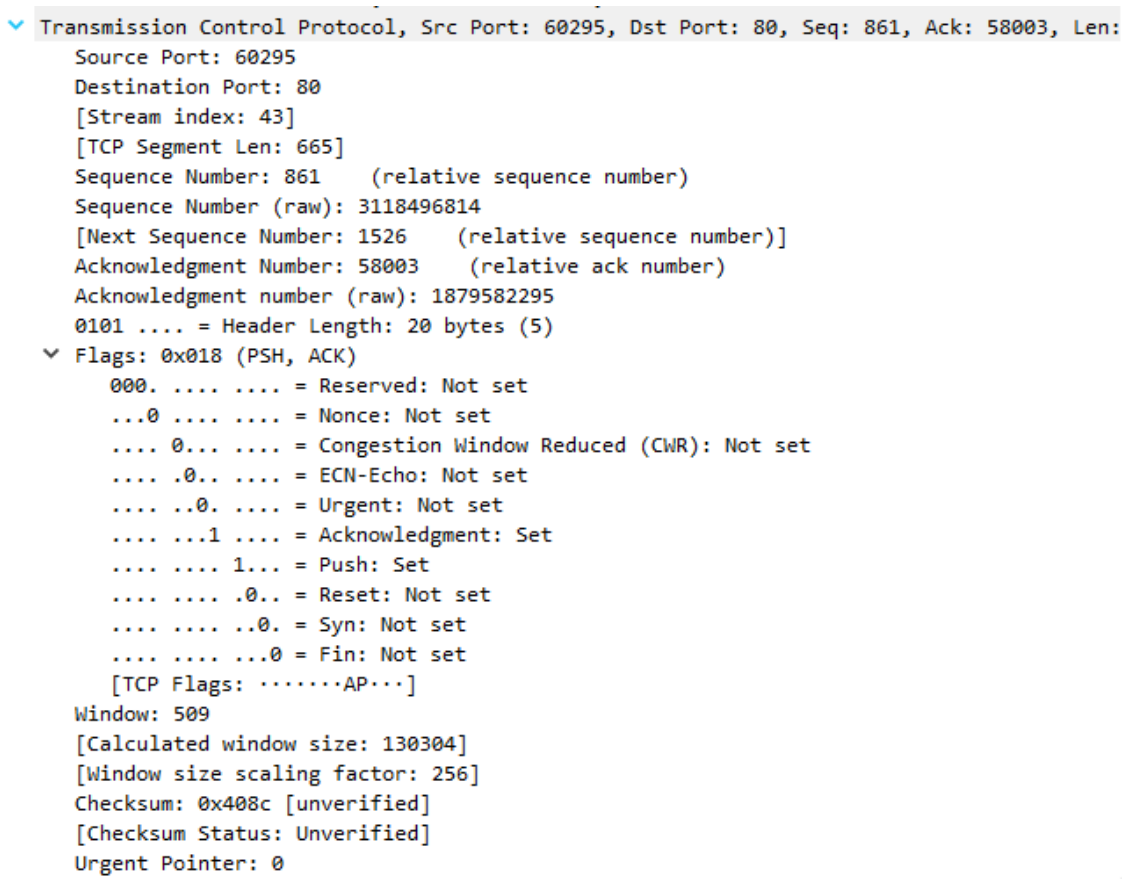
Ключ “d” предотвращает попытки команды tracert разрешения IP-адресов промежуточных маршрутизаторов в имена, что ускоряет скорость вывода результатов.

3. Анализ HTTP-трафика

Сайт: bears.com

Структура PDU:

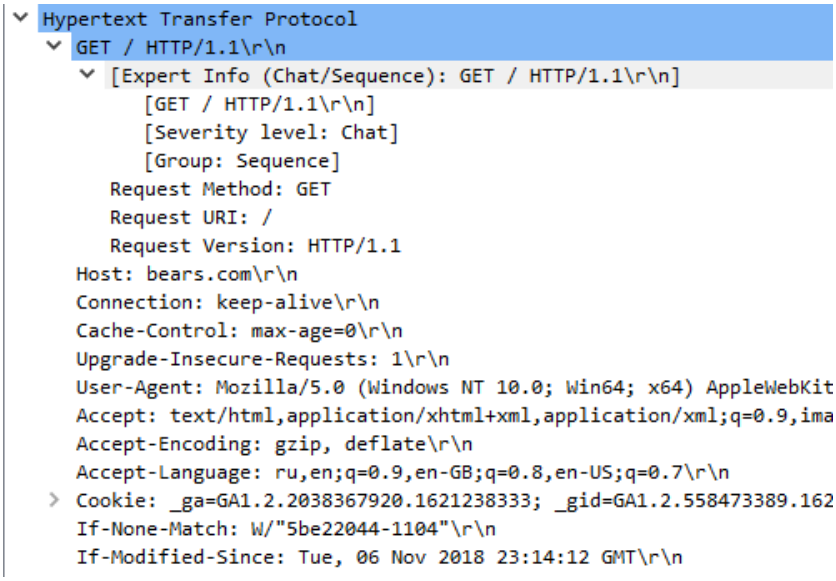
TCP



- Source Port (2 байт) – порт отправителя
- Destination Port (2 байт) – порт получателя
- Sequence Number (4 байт) – порядковый номер для гарантирования правильного и в необходимом порядке получения сегментов
- Acknowledgment Number (4 байт) – номер подтверждения. Если ACK=1, то это порядковый номер октета, который отправитель хочет получит (все предыдущие октеты были получены).
- Header Length (нибл) - длина заголовка в 32-битных словах. Минимально равно 5.
- Flags (12 бит):
    - Reserved (3 бит) – установлено в 0
    - Nonce
    - CWR (Congestion Window Reduced)
    - ECN-Echo
    - URG (Urgent) – указывает, что сегмент содержит срочные данные
    - ACK (Acknowledgment) – указывает, что сегмент содержит номер подтверждения
    - PSH (Push) – указывает, что данные нужно протолкнуть к получающему пользователю (в приложении)
    - RST (Reset) – сбрасывает соединение
    - SYN – используется для синхронизации порядковых номеров
    - FIN – указывает конец данных и завершение соединения
- Window (2 байт) – «окно», объем данных, который может принять получатель
- Checksum (2 байт) – контрольная сумма
- Urgent Pointer (2 байт) – указатель срочности, который сообщает порядковый номер для октета, следующего за срочными данными.
- Опции (переменная длина, кратная 8 бит)

HTTP

HTTP-запрос



Стартовая строка (Starting line):

- Request Method (HTTP-команда) – информирует сервер о цели запроса клиента
- Request URI – уникальный идентификатор ресурса на исходном сервере/шлюзе
- Request Version - номер версии HTTP

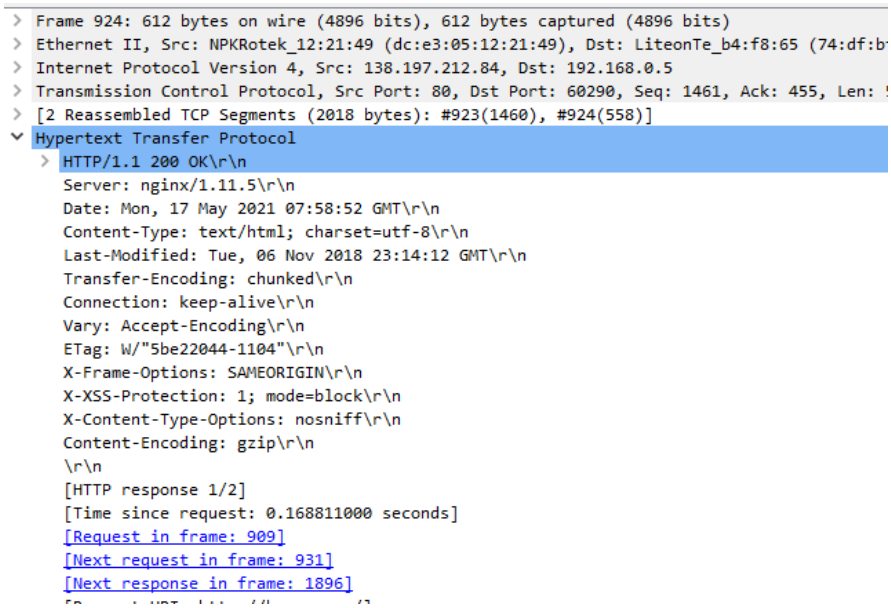
Основные заголовки (General Headers):

- Connection – параметры, требуемые для этого соединения
- Cache-Control - определение директив, которые должны выполняться всеми механизмами кэширования в цепочке запросов

Заголовки запроса (Request Headers):

- Host – хост и номер порта запрашиваемого ресурса
- User-Agent – информация о пользовательском устройстве, создавшем запрос
- Accept – типы, приемлемые для ответа
- Accept-Encoding – ограничивает допустимые в ответе кодировки содержимого
- Accept-Language – ограничивает набор естественных языков для ответа на запрос, выбор сервером ресурсов идет с учетом Accept-заголовков
- Cookie – список куки
- If-None-Match – сервер отправляет ресурс, только если он не соответствует одному из перечисленных тегов
- If-Modified-Since – установлена дата, сервер отправляет ресурс, только если он был изменен с этой даты

HTTP-ответ



Стартовая строка:

- Response Version – версия протокола, обычно HTTP/1.1
- Status Code – код состояния, показывающий, был ли запрос успешным
- Response Phrase - текстовое короткое пояснение к коду ответа для пользователя, не влияет на сообщение и не является обязательным

Основные заголовки:

- Connection – параметры, требуемые для этого соединения
- Date – дата и время, когда было отправлено сообщение

Заголовки ответа (Response Headers):

- Server - информация о ПО, используемом исходным сервером для обработки запроса
- ETag - текущее значение тега объекта для запрошенного варианта
- Vary – указывает набор полей заголовка запроса, который определяет, разрешено ли кэшу использовать response для ответа на последующие запросы без повторной проверки

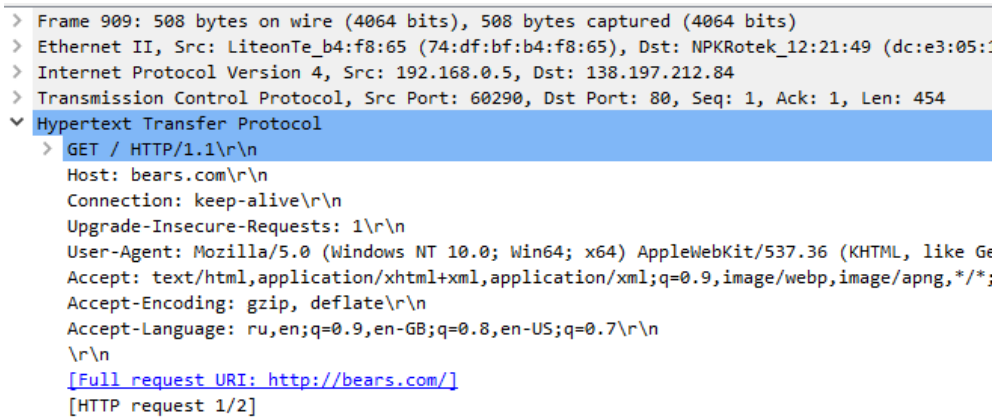
Заголовки объекта (Entity Headers):

- Last-Modified - дата и время, когда исходный сервер считает, что ресурс был изменен
- Content-Type – тип носителя тела объекта, отправленного получателю
- Transfer-Encoding – указывает, какой тип преобразования был применен к телу сообщения, чтобы безопасно передать его между отправителем и получателем

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).

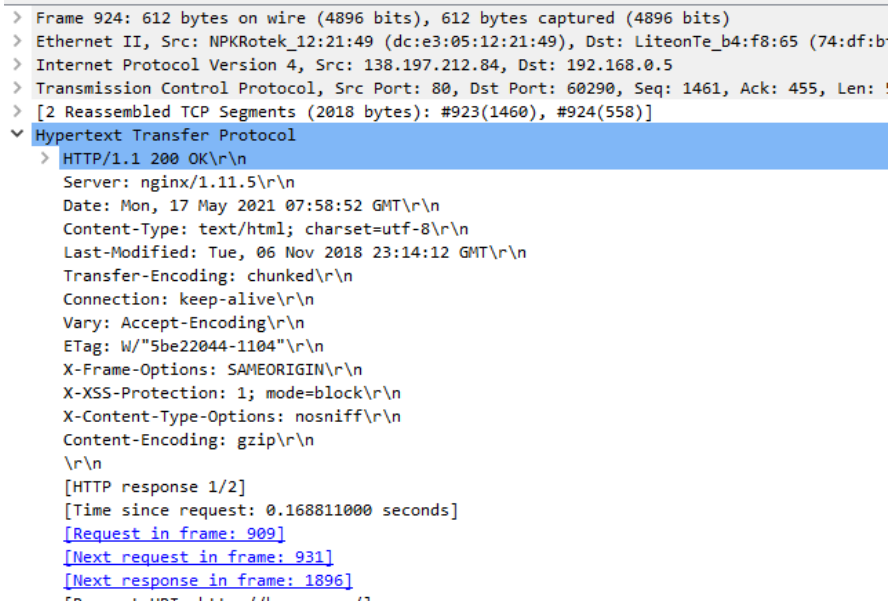
Первичный запрос:



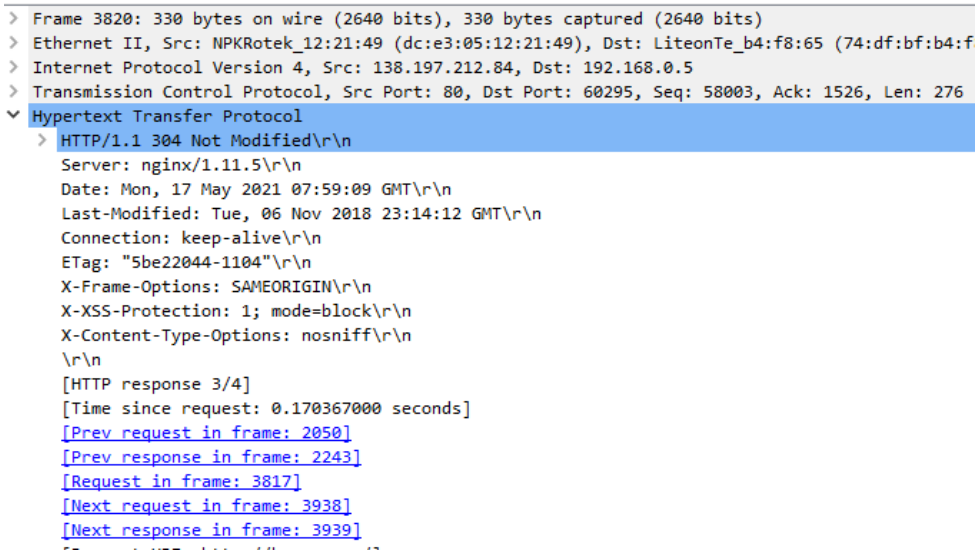
Вторичный запрос:

В повторном запросе появляются такие поля, как Cache-Control, Cookie, If-None-Match и If-Modified-Since, используемые для ускорения прогрузки страницы.

Ответ на 1 запрос:



Ответ на 2 запрос:

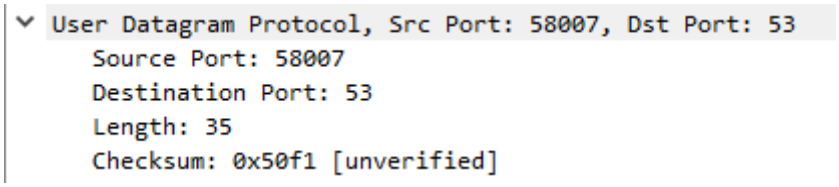


При повторном запросе исчезают поля, связанные с кодировкой тела запроса, т.к. они уже были определены при первом ответе.

4. Анализ DNS-трафика

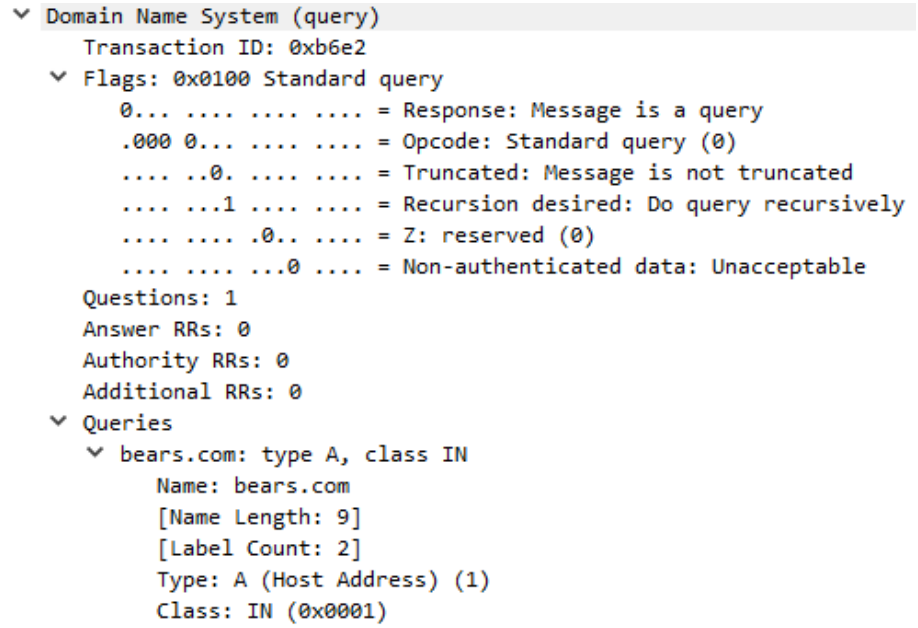
Структура PDU:

UDP – заголовок=8 байт



- Source Port (2 байт) – номер порта источника
- Destination Port (2 байт) – номер порта назначения
- Length (2 байт) – длина заголовка и данных в байтах
- Checksum (2 байт) – контрольная сумма заголовка и данных

DNS-запрос – заголовок=12 байт



- Transaction ID (2 байт)
- Flags (2 байт):
    - QR – тип операции запрос (0) или ответ (1)
    - Opcode (4 бит) – тип запроса, обычно используется только 0 – стандартный запрос
    - Truncated – пакет был обрезан (1) или не был (0)
    - Recursion desired – если равен 1, клиент просит сервер работать в рекурсивном режиме
    - Z – зарезервирован, равен 0
    - Non-authenticated data – указывает, требуется ли неподтвержденная информация
- Questions (2 байт) – количество запросов – обычно 1
- Answer RRs (2 байт) – количество ответов – обычно 0
- Authority RRs (2 байт) – количество ответов об авторитетных серверах – обычно 0
- Additional RRs (2 байт) – количество дополнительных ответов – обычно 0

Данные:

- Queries – запросы DNS
    - Name – доменное имя, к которому привязана или которому принадлежит данная ресурсная запись
    - Type (2 байт) – тип ресурсной записи, определяет формат и назначение записи
    - Class (2 байт) – класс записи, считается, что DNS может использоваться не только с TCP/IP, но и с другими типами сетей, код здесь определяет тип сети.

DNS-ответ

```
Domain Name System (response)
    Transaction ID: 0xb6e2
  Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not au
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  > Queries
  Answers
    bears.com: type A, class IN, addr 138.197.212.84
        Name: bears.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 83153 (23 hours, 5 minutes, 53 seconds)
        Data length: 4
        Address: 138.197.212.84
```

Отличие от заголовка запроса:

- Flags (2 байт):
    - QR
    - Opcode (4 бит)
    - Authoritative
    - Truncated
    - Recursion desired

- Recursion available – если равен 1, сервер сообщает, что он может работать в рекурсивном режиме
- Z
- Answer authenticated – указывает, что полученный ответ авторитетный (1) или нет (0)
- Non-authenticated
- Reply code (4 бит) – статус выполнения операции, статус 0 говорит о том, что операция прошла успешно, любые другие коды говорят о какой-то ошибке.

Данные:

- Queries – содержит запросы
- Answers
  - Name – как в запросе
  - Type (2 байт) – как в запросе
  - Class (2 байт) – как в запросе
  - Time to Live (4 байт) – допустимое время хранения данной записи в кэше неответственного DNS-сервера
  - Data Length (2 байт) – длина поля данных
  - Address (4 байт) – ip-адрес сервера

Вопросы:

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

Запрос отправляется на DNS-сервер, где определяется ip-адрес

2. Какие бывают типы DNS-запросов?

- Recursive (рекурсивный) - это первый тип запроса, который возникает, когда клиентское устройство пытается получить доступ к веб-сайту, т.е. запрос для определения ip-адреса по доменному имени.
- Iterative (итеративный) - запрос, который возникает между DNS серверами, когда один из них не имеет соответствующих записей. Инициатор запроса будет контактировать с сервером, который имеет нужную запись.
- Inverse (обратный) – при известном IP-адресе запрашивается информация о доменном имени.

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

Когда изображения взяты со сторонних сервисов

5. Анализ ARP-трафика

Структура PDU

ARP – заголовок=28 байт

```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 9a:ca:33:7c:2c:64 (9a:ca:33:7c:2c:64)
    Sender IP address: 172.20.10.1
    Target MAC address: 9a:ca:33:c7:79:bb (9a:ca:33:c7:79:bb)
    Target IP address: 172.20.10.6
```

- Hardware type (2 байт) – тип аппаратного адреса
- Protocol type (2 байт) – тип адреса протокола, к которому будет приведено соответствие
- Hardware size (1 байт)– размер в байтах аппаратного адреса
- Protocol size (1 байт) – размер в байтах адреса протокола
- Opcode (2 байт) – тип операции
- Sender MAC address (6 байт) – аппаратный адрес отправителя
- Sender IP address (4 байт) – ip-адрес отправителя
- Target MAC address (6 байт) – аппаратный адрес назначения
- Target IP address (4 байт) – ip-адрес назначения

Вопросы:

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

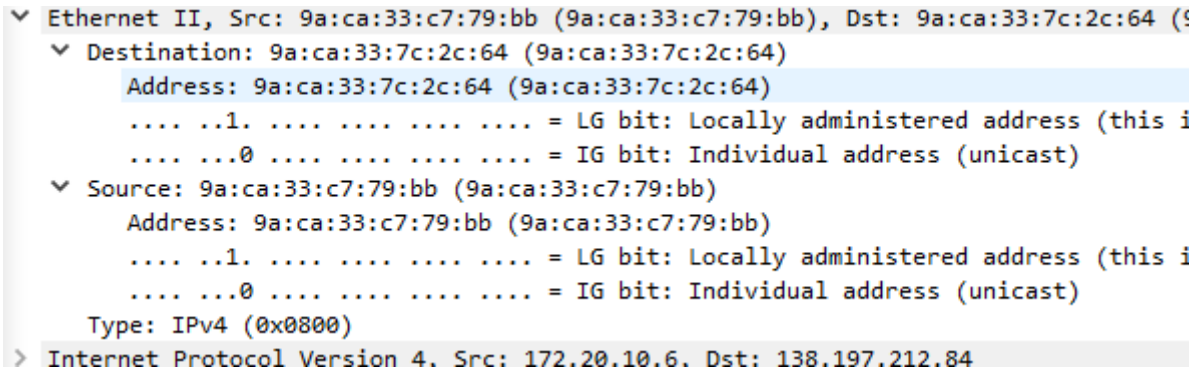
```
Ethernet II, Src: 9a:ca:33:c7:79:bb (9a:ca:33:c7:79:bb), Dst: Broadcast (ff:ff:ff:ff:f
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
      Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NO
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
  Source: 9a:ca:33:c7:79:bb (9a:ca:33:c7:79:bb)
      Address: 9a:ca:33:c7:79:bb (9a:ca:33:c7:79:bb)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NO
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
```

Запрос: 9a:ca:33:c7:79:bb – MAC-адрес ноутбука и широковещательный MAC-адрес (после очистки кэша неизвестен MAC-адрес)

```
Ethernet II, Src: 9a:ca:33:7c:2c:64 (9a:ca:33:7c:2c:64), Dst: 9a:ca:33:c7:79:bb (9a:
  Destination: 9a:ca:33:c7:79:bb (9a:ca:33:c7:79:bb)
      Address: 9a:ca:33:c7:79:bb (9a:ca:33:c7:79:bb)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is M
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: 9a:ca:33:7c:2c:64 (9a:ca:33:7c:2c:64)
      Address: 9a:ca:33:7c:2c:64 (9a:ca:33:7c:2c:64)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is M
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
```

Ответ: 9a:ca:33:c7:79:bb – MAC-адрес компьютера, 9a:ca:33:7c:2c:64 – MAC-адрес роутера

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



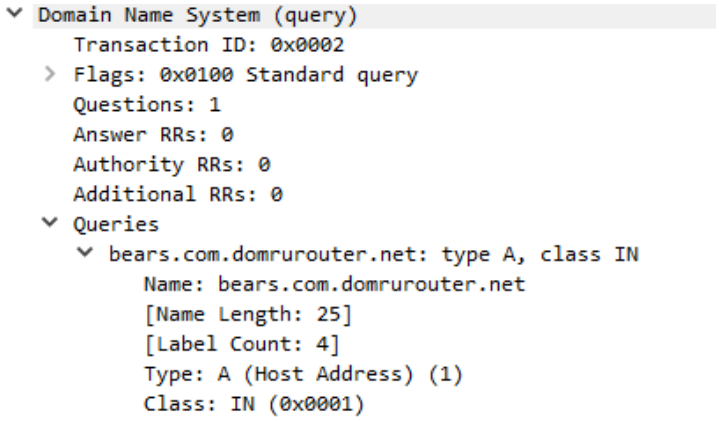
9a:ca:33:c7:79:bb – MAC-адрес материнской платы, 9a:ca:33:7c:2c:64 – MAC-адрес роутера

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

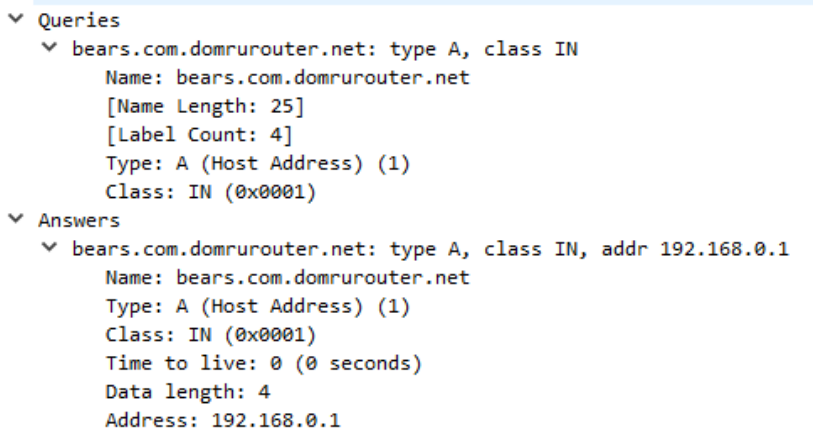
Чтобы отправить ответ с MAC-адресом обратно.

6. Анализ трафика утилиты nslookup

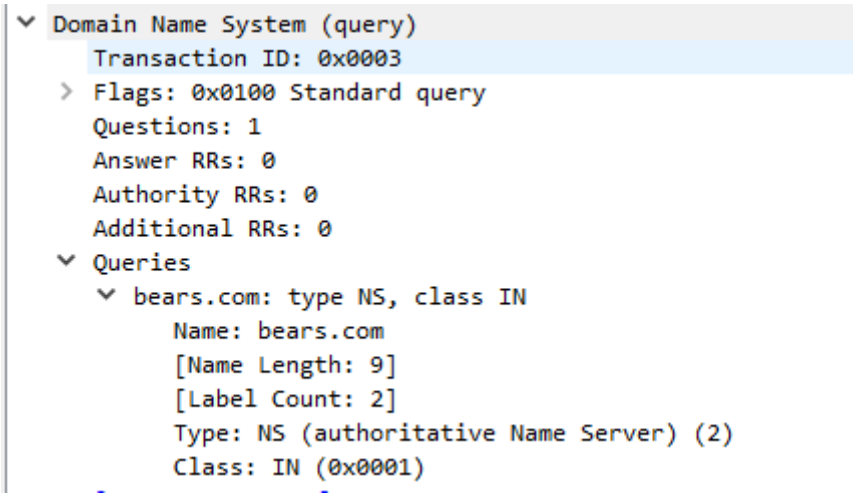
Query



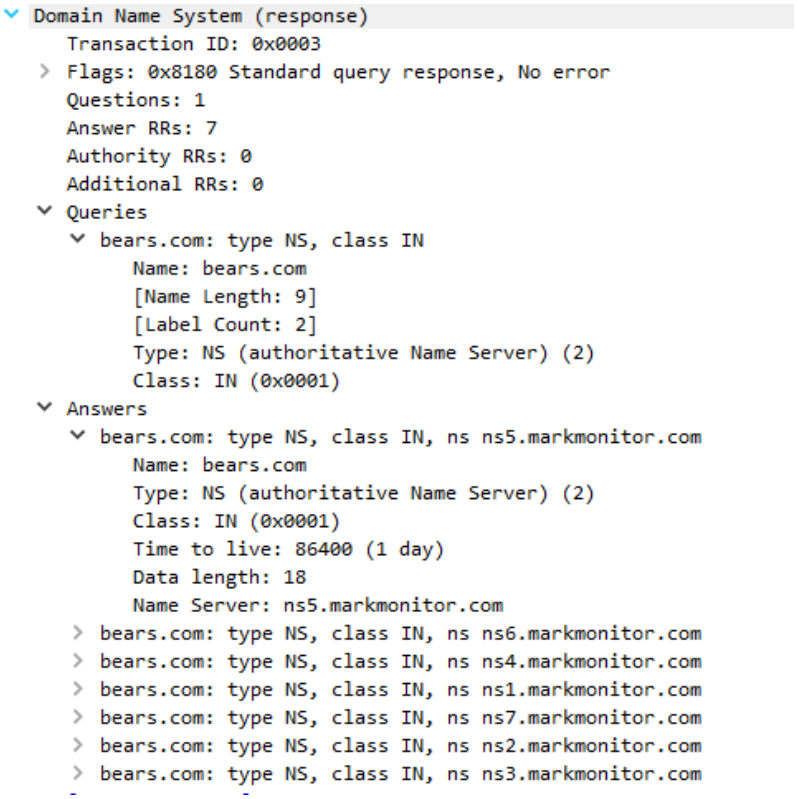
Response



Query -type=NS:



Response -type=NS:



1. Чем различается трасса трафика двух запросов выше?

С -type=NS возвращается список используемых DNS-серверов, без – ip-адрес по домену.

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

Информация о DNS-серверах

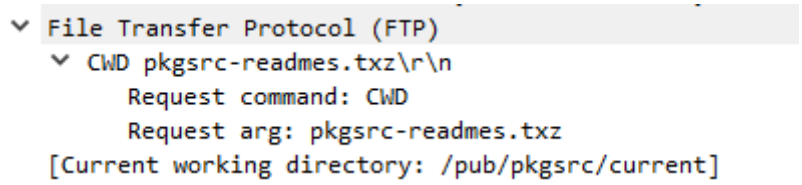
3. Каковы имена серверов, возвращающих авторитарный (authoritative) отклик?

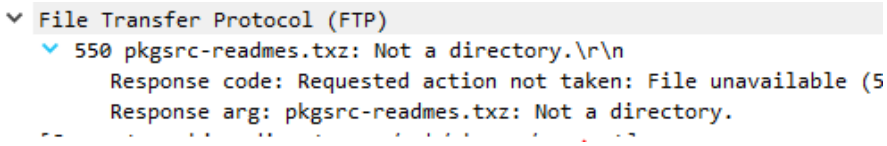
ns1.markmonitor.com – ns7.markmonitor.com

7. Анализ FTP-трафика

ftp://iso.netbsd.ru

FTP





FTP-request

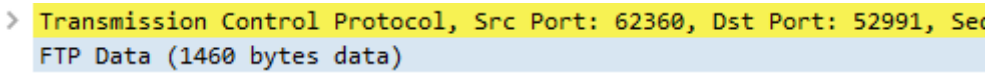
- Request command – команда от клиента к серверу
- Request arg – аргументы запроса

FTP-response

- Response code – код ответа
- Response arg – аргументы ответа

Вопросы:

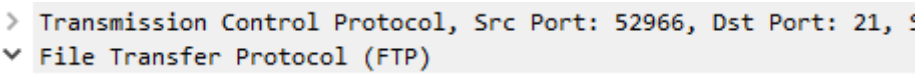
1. Сколько байт данных содержится в пакете FTP-DATA?



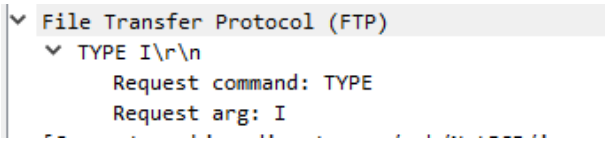
1460 байт

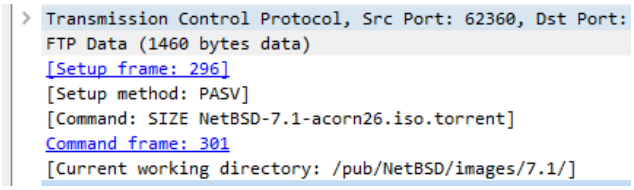
2. Как выбирается порт транспортного уровня, который используется для передачи FTP-пакетов?

Использует 2 заданных порта: 21 для управления и 20 для передачи данных



3. Чем отличаются пакеты FTP от FTP-DATA?



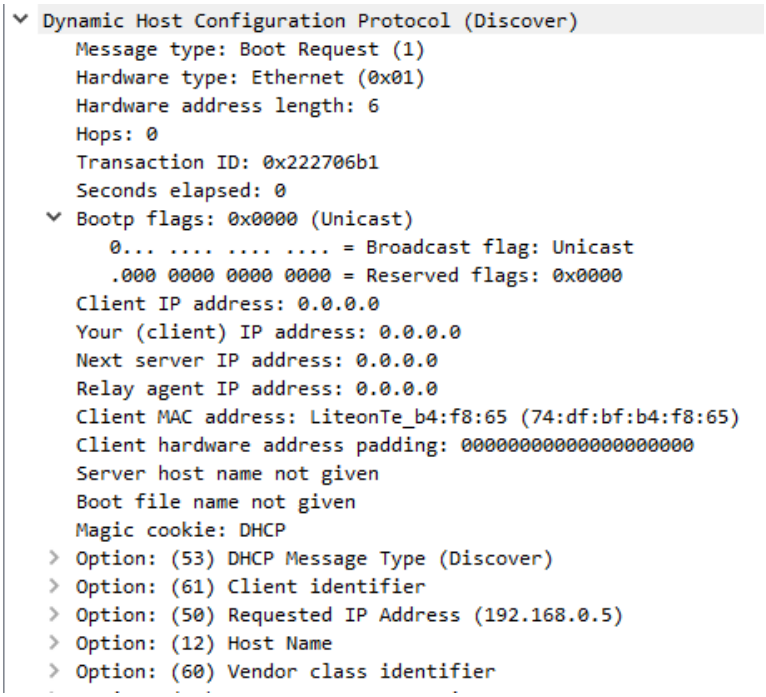


FTP содержит заголовки взаимодействия сервера-клиента, FTP-Data содержит установленный метод и текущую рабочую директорию. Первый используется для управления, а второй для передачи данных.

8. Анализ DHCP-трафика

Структура PDU

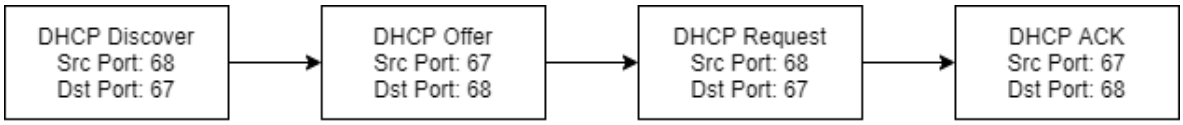
DHCP



- Message type (1 байт) – тип DHCP-сообщения
- Hardware type (1 байт) – тип адреса на канальном уровне
- Hardware address length (1 байт) – длина аппаратного адреса в байтах
- Hops (1 байт) – количество промежуточных маршрутизаторов, которые находятся на пути между клиентов и сервером
- Transaction ID (4 байт) – когда клиент начинает процесс получения IP-адреса, он генерирует значение для этого поля, чтобы сервер не перепутал конкретный процесс этого клиента с другим процессом
- Seconds elapsed (2 байт) – время в секундах с момента начала процесса получения IP-адреса
- Bootp flags (2 байт):
    - Broadcast flag
    - Reserved flags (15 бит) - зарезервированные
- Client IP address (4 байт) – IP-адрес клиента, заполняется только если у клиента уже есть IP-адрес и он может ответить на ARP-запрос
- Your (client) IP address (4 байт) – IP-адрес, который DHCP-сервер вписывает, тем самым предлагая клиенту
- Next server IP address (4 байт) – IP-адрес сервера
- Relay agent IP address (4 байт) – если используется схема с DHCP Relay Agent, в этом поле передается его IP-адрес
- Client MAC address (6 байт) – если на канальном уровне используется протокол Ethernet, то в это поле записывается MAC-адрес клиента
- Client hardware address padding (10 байт)
- Server host name (64 байт) – если у сервера есть доменное имя/имя хоста, то он может сообщить его в этом поле
- Boot file name (128 байт) – указатель для бездисковых рабочих станций о том, как называется файл на сервере, которые следует использовать для загрузки
- Options – поле опций, в котором передается полезная информация для динамической конфигурации хоста.

Вопросы:

Нарисуйте временную диаграмму, иллюстрирующую последовательность обмена первыми четырьмя DHCP-пакетами Discover/Offer/Request/ACK. Укажите для каждого пакеты номера портов источника и назначения.



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

Оба пакета служат для получения клиентом IP-адреса от сервера DHCP.

Если клиент еще не имеет собственного IP-адреса, то сначала он выполняет широковещательный запрос по всей физической сети с целью обнаружить доступные DHCP-серверы, отправляя сообщение типа DHCP Discover, в качестве адреса источника указывается 0.0.0.0, а в качестве адреса назначения – широковещательный адрес 255.255.255.255.

Если клиент ранее уже получал IP-адрес и срок его аренды еще не прошел, то клиент может пропустить стадию DHCP Discover, начав с запроса DHCP Request с идентификатором сервера, который выдал адрес в прошлый раз.

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

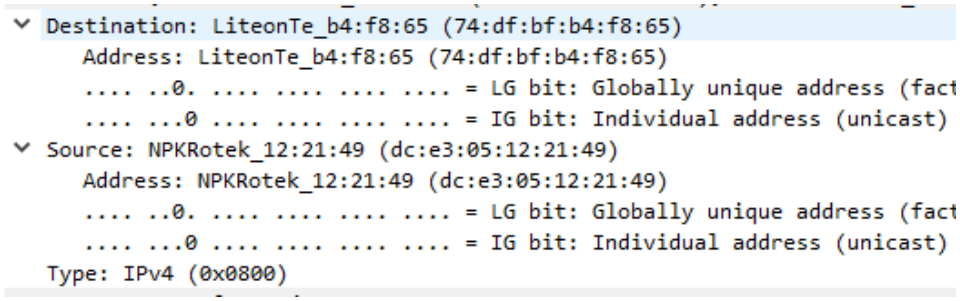
```
0.0.0.0             255.255.255.255     DHCP      344 DHCP Discover
192.168.0.1         192.168.0.5         DHCP      590 DHCP Offer
0.0.0.0             255.255.255.255     DHCP      370 DHCP Request
192.168.0.1         192.168.0.5         DHCP      590 DHCP ACK
```

```
Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x222706b1
    Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.0.5
```

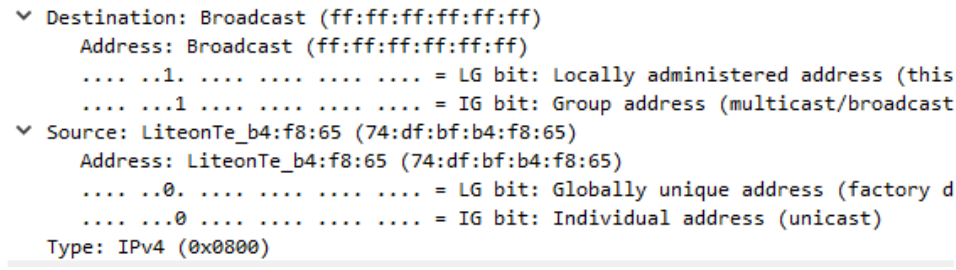
При Discover и Request адрес источника равен 0.0.0.0, т.к. компьютер еще не имеет свой ip-адрес. В Discover и Request адрес 255.255.255.255, т.к. это широковещательное сообщение, Request широковещательное, т.к. клиент должен сообщить всем серверам о том, какой адрес он хочет получить и с каким сервером он хочет продолжить взаимодействие. В Offer и ACK 192.168.0.1 – это адрес DHCP-сервера, а 192.168.0.5 – предлагаемый адрес, он появляется, т.к. до DHCP-сервера на этот момент уже дошел запрос и он поставил ip-адрес.

```
> Destination: Broadcast (ff:ff:ff:ff:ff:ff)
v Source: LiteonTe_b4:f8:65 (74:df:bf:b4:f8:65)
    Address: LiteonTe_b4:f8:65 (74:df:bf:b4:f8:65)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
```

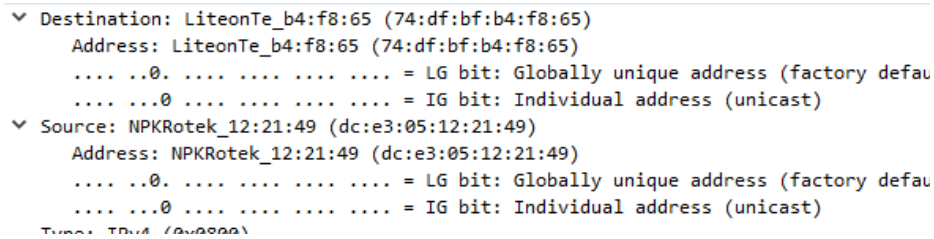
Discover – MAC-адрес назначения широковещательный, MAC-адрес источника – MAC-адрес клиента.



Offer – MAC-адрес источника – MAC-адрес DHCP-сервера



Request – MAC-адрес назначения широковещательный



ACK

3. Каков IP-адрес DHCP-сервера?

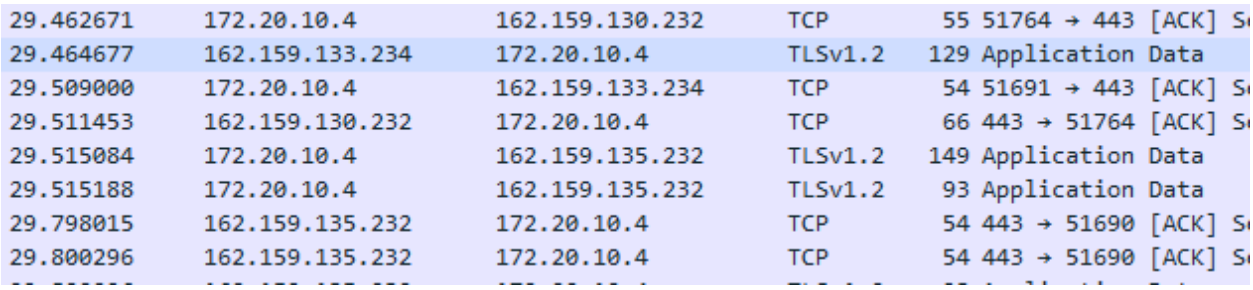
Из скринов выше: 192.168.0.1

4. Что произойдёт, если очистить использованный фильтр «bootp»?

Отобразятся все захваченные пакеты

<u>9. Анализ Discord-трафика</u>

Текстовое сообщение:



TLS дает возможность клиент-серверным приложениям осуществлять связь так, что нельзя прослушать пакеты и осуществить несанкционированный доступ (т.е. сообщение прочитать не получится).

Сеанс аудио-общения:

345	4.260458	172.20.10.4	188.122.64.156	RTCP	102 Sender Rep
346	4.577315	188.122.64.156	172.20.10.4	RTCP	94 Receiver R
347	4.619033	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
348	4.622768	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
349	4.646832	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
350	4.667096	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
351	4.690858	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
352	4.706722	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
353	4.726786	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
354	4.751121	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
355	4.766994	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
356	4.808846	188.122.64.156	172.20.10.4	UDP	85 50008 → 52
357	5.570745	188.122.64.156	172.20.10.4	RTCP	94 Receiver R
358	5.571165	172.20.10.4	162.159.138.234	TLSv1.2	138 Applicatio

Аудио-общение осуществляется на UDP-пакетах. RTCP используется для передачи информации о задержках и потерях медиа-пакетов, уровне звукового сигнала.

Сеанс видео-общения:

6188	18.078242	188.122.83.7	172.20.10.4	UDP	1151 50002 → 569
6189	18.078375	188.122.83.7	172.20.10.4	UDP	1151 50002 → 569
6190	18.079001	188.122.83.7	172.20.10.4	UDP	1151 50002 → 569
6191	18.079001	188.122.83.7	172.20.10.4	UDP	94 50002 → 569
6192	18.079080	188.122.83.7	172.20.10.4	UDP	1147 50002 → 569
6193	18.084678	172.20.10.4	188.122.83.7	RTCP	102 Sender Repo
6194	18.086389	188.122.83.7	172.20.10.4	UDP	1147 50002 → 569
6195	18.086525	188.122.83.7	172.20.10.4	UDP	1147 50002 → 569

Аналогично аудио-общению.

Вопросы:

1. Чем различаются пакеты разных видов трафика (текст, аудио, видео)?

Текст передается через TLS-пакеты, аудио и видео через RTCP, UDP и TLS пакеты.

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

Для идентификации аудио и видео можно использовать фильтр rtcp || udp.

Вывод:

В данной лабораторной работе я изучила основные сетевые протоколы, их PDU, на практике увидела разницу между ними. До выполнения все протоколы сливались в один, не было понятно, зачем так много разных протоколов. Теперь стало ясно, что каждый протокол создан для определенной задачи. Однако все еще остается удивительным то, как быстро анализируются заголовки пакетов.