

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет безопасности информационных технологий

Дисциплина:
«Компьютерные сети»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
«Анализ трафика компьютерных сетей утилитой Wireshark»

Выполнили:

Бардышев Артём Антонович,
студент группы N3346



(подпись)

Проверил:

Ярошевский Дмитрий Сергеевич,
Ведущий инженер, ФБИТ

(отметка о выполнении)

(подпись)

Санкт-Петербург
2025 г.

СОДЕРЖАНИЕ

| | |
|--|----|
| Введение..... | 4 |
| 1 Подготовка | 5 |
| 2 Анализ полученных пакетов..... | 6 |
| 2.1 DNS-запрос..... | 6 |
| 2.2 TCP..... | 7 |
| 2.3 TLS-Handshake | 7 |
| 2.4 HTTP | 8 |
| 2.5 ARP | 8 |
| 2.6 ICMP | 9 |
| 2.7 nslookup..... | 9 |
| 3 Ответы на вопросы | 11 |
| 3.1 Имеет ли место фрагментация исходного пакета?..... | 11 |
| 3.2 Какая информация указывает, является ли фрагмент последним или промежуточным?..... | 11 |
| 3.3 Чему равно количество фрагментов при передаче ping-пакетов? | 11 |
| 3.4 Как изменить поле TTL? | 11 |
| 3.5 Что содержится в поле данных ping-пакета? | 12 |
| 3.6 Сколько байт в заголовке IP и в данных?..... | 12 |
| 3.7 Как и почему изменяется TTL в последовательных ICMP-пакетах tracer? | 12 |
| 3.8 Чем отличаются ICMP-пакеты tracer и ping?..... | 12 |
| 3.9 Чем отличаются ICMP reply и ICMP error? | 12 |
| 3.10 Что изменится, если убрать ключ -d? | 13 |
| 3.11 Как выглядит HTTP-запрос и ответ? | 13 |
| 3.12 Что означает строка состояния HTTP 200 OK? | 13 |
| 3.13 Что происходит при повторном обращении к странице? | 13 |
| 3.14 Почему адрес DNS-запроса не совпадает с адресом сайта?..... | 13 |
| 3.15 Какие бывают типы DNS-запросов?..... | 13 |
| 3.16 Почему браузер делает дополнительные DNS-запросы? | 14 |
| 3.17 Какие MAC-адреса есть в ARP-пакетах и что они означают? | 14 |
| 3.18 Почему в ARP-запросе указан IP-адрес источника? | 14 |
| 3.19 Чем различаются DNS-запросы типа A и NS?..... | 14 |
| 3.20 Что содержится в поле “Answers” DNS-ответа?..... | 14 |

| | | |
|---------------------------------------|--|----|
| 3.21 | Является ли ответ DNS-сервера авторитативным? | 15 |
| 3.22 | Почему адрес DNS-сервера отличается от адреса сайта? | 15 |
| 3.23 | Что содержится в поле “Authority” DNS-ответа? | 15 |
| 3.24 | Что произойдёт, если очистить ARP-кэш? | 15 |
| Заключение..... | | 16 |
| Список использованных источников..... | | 17 |

ВВЕДЕНИЕ

Цель работы – Изучить структуру протокольных блоков данных (PDU), проанализировав реальный сетевой трафик на своём компьютере при помощи Wireshark.

В процессе выполнения учебно-исследовательской работы (УИР) необходимо: – Освоить базовые навыки работы с анализатором трафика Wireshark: захват, фильтрация и анализ пакетов. – Исследовать работу служебных протоколов ICMP, DNS и ARP на примере системных утилит ping, tracert и nslookup.

1 ПОДГОТОВКА

Устанавливаем Wireshark, настраиваем фильтр захвата

host baatraining.com (т.к. для анализа было решено взять сайт содержащий мои инициалы – Бардышев Артём Антонович baatraining.com)

Выбираем активный интерфейс – беспроводная сеть

При запуске захвата открываем сайт в браузере, как только страница загрузится полностью захват можно останавливать.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|-----------------|-----------------|----------|--------|---|
| 1 | 0.000000 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 111 | Application Data |
| 2 | 0.000054 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 93 | Application Data |
| 3 | 0.017393 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=1 Ack=58 Win=305 Len=0 |
| 4 | 0.017393 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=1 Ack=97 Win=305 Len=0 |
| 5 | 0.017393 | 185.199.111.153 | 192.168.68.44 | TLSv1.2 | 93 | Application Data |
| 6 | 0.017393 | 185.199.111.153 | 192.168.68.44 | TLSv1.2 | 276 | Application Data |
| 7 | 0.017524 | 192.168.68.44 | 185.199.111.153 | TCP | 54 | 63394 → 443 [ACK] Seq=97 Ack=262 Win=254 Len=0 |
| 8 | 0.051215 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 124 | Application Data |
| 9 | 0.051292 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 126 | Application Data |
| 10 | 0.051322 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 124 | Application Data |
| 11 | 0.051577 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 128 | Application Data |
| 12 | 0.071377 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=262 Ack=167 Win=305 Len=0 |
| 13 | 0.071377 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=262 Ack=239 Win=305 Len=0 |
| 14 | 0.071377 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=262 Ack=309 Win=305 Len=0 |
| 14 | 0.071377 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=262 Ack=309 Win=305 Len=0 |
| 15 | 0.071377 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=262 Ack=383 Win=305 Len=0 |
| 16 | 0.071377 | 185.199.111.153 | 192.168.68.44 | TCP | 4422 | 443 → 63394 [PSH, ACK] Seq=262 Ack=383 Win=305 Len=436 |
| 17 | 0.071377 | 185.199.111.153 | 192.168.68.44 | TLSv1.2 | 1087 | Application Data |
| 18 | 0.071377 | 185.199.111.153 | 192.168.68.44 | TCP | 4422 | 443 → 63394 [PSH, ACK] Seq=5663 Ack=383 Win=305 Len=436 |
| 19 | 0.071563 | 192.168.68.44 | 185.199.111.153 | TCP | 54 | 63394 → 443 [ACK] Seq=383 Ack=10031 Win=255 Len=0 |
| 20 | 0.072873 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 89 | Application Data |
| 21 | 0.074688 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 126 | Application Data |
| 22 | 0.074830 | 185.199.111.153 | 192.168.68.44 | TLSv1.2 | 11702 | Application Data, Application Data |
| 23 | 0.074830 | 185.199.111.153 | 192.168.68.44 | TLSv1.2 | 418 | Application Data |
| 24 | 0.074879 | 192.168.68.44 | 185.199.111.153 | TCP | 54 | 63394 → 443 [ACK] Seq=490 Ack=22043 Win=255 Len=0 |
| 25 | 0.079842 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 123 | Application Data |
| 26 | 0.079919 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 127 | Application Data |
| 27 | 0.079947 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 123 | Application Data |
| 27 | 0.079947 | 192.168.68.44 | 185.199.111.153 | TLSv1.2 | 123 | Application Data |
| 28 | 0.099400 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=22043 Ack=418 Win=305 Len=0 |
| 29 | 0.099400 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=22043 Ack=490 Win=305 Len=0 |
| 30 | 0.099400 | 185.199.111.153 | 192.168.68.44 | TCP | 4422 | 443 → 63394 [PSH, ACK] Seq=22043 Ack=490 Win=305 Len=436 |
| 31 | 0.099400 | 185.199.111.153 | 192.168.68.44 | TLSv1.2 | 1123 | Application Data |
| 32 | 0.099561 | 192.168.68.44 | 185.199.111.153 | TCP | 54 | 63394 → 443 [ACK] Seq=701 Ack=27480 Win=255 Len=0 |
| 33 | 0.104705 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=27480 Ack=559 Win=305 Len=0 |
| 34 | 0.104705 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=27480 Ack=632 Win=305 Len=0 |
| 35 | 0.104705 | 185.199.111.153 | 192.168.68.44 | TCP | 60 | 443 → 63394 [ACK] Seq=27480 Ack=701 Win=305 Len=0 |
| 36 | 0.104705 | 185.199.111.153 | 192.168.68.44 | TCP | 8790 | 443 → 63394 [PSH, ACK] Seq=27480 Ack=701 Win=305 Len=8734 |
| 37 | 0.104812 | 192.168.68.44 | 185.199.111.153 | TCP | 54 | 63394 → 443 [ACK] Seq=701 Ack=36216 Win=255 Len=0 |
| 38 | 0.109865 | 185.199.111.153 | 192.168.68.44 | TCP | 7334 | 443 → 63394 [PSH, ACK] Seq=36216 Ack=701 Win=305 Len=7278 |
| 39 | 0.109865 | 185.199.111.153 | 192.168.68.44 | TLSv1.2 | 428 | Application Data |
| 40 | 0.109923 | 192.168.68.44 | 185.199.111.153 | TCP | 54 | 63394 → 443 [ACK] Seq=701 Ack=43870 Win=255 Len=0 |

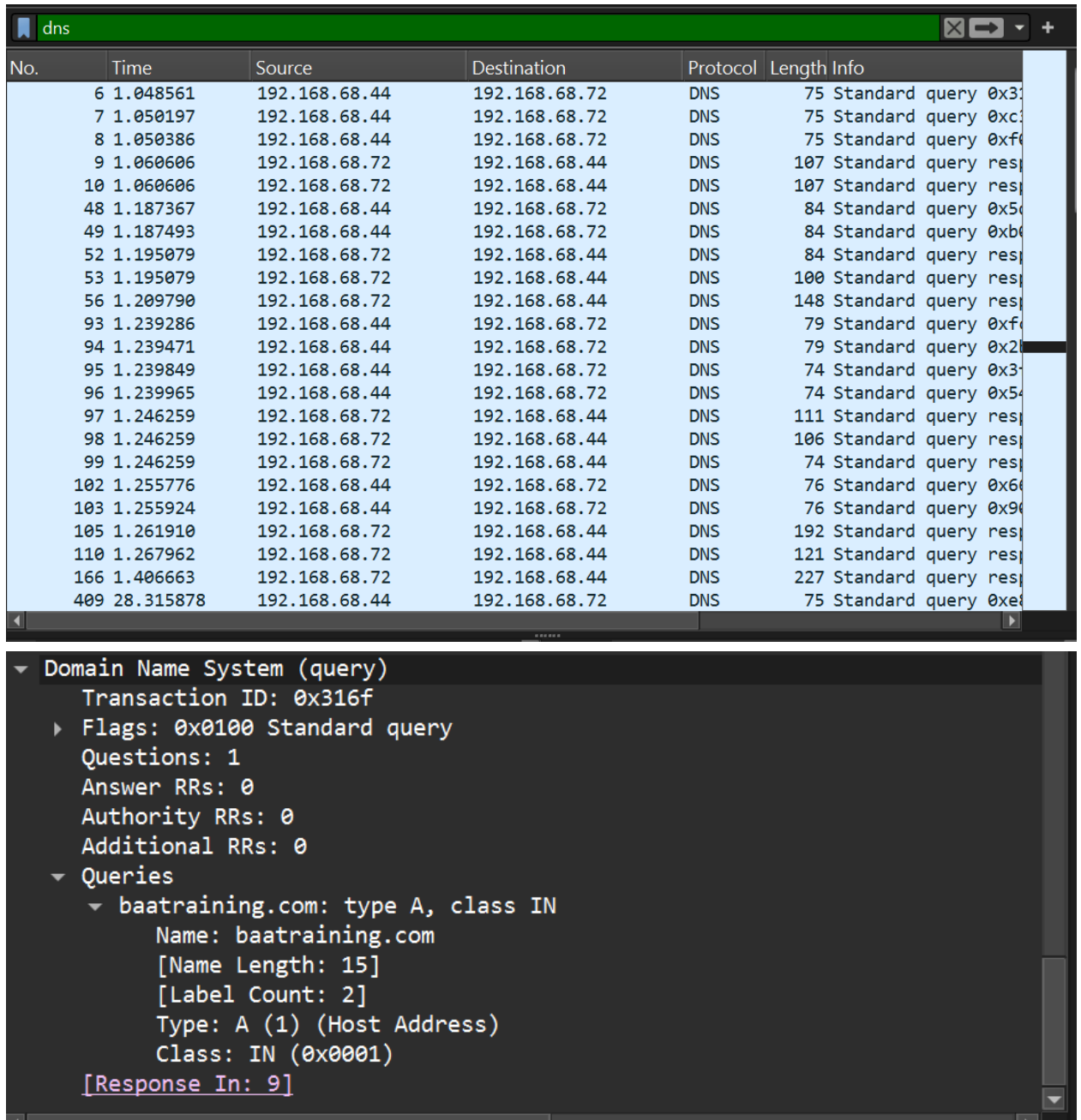
2 АНАЛИЗ ПОЛУЧЕННЫХ ПАКЕТОВ

Теперь нужно разобрать типы пакетов, которые появились в моем захвате.

2.1 DNS-запрос

В фильтре Wireshark вводим:

Dns



The image shows a Wireshark packet capture of DNS traffic. The top pane displays a list of 20 DNS packets. The bottom pane shows the details of a selected DNS query for baatraining.com.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|---------------|---------------|----------|--------|----------------------------------|
| 6 | 1.048561 | 192.168.68.44 | 192.168.68.72 | DNS | 75 | Standard query 0x316f |
| 7 | 1.050197 | 192.168.68.44 | 192.168.68.72 | DNS | 75 | Standard query 0xc316f |
| 8 | 1.050386 | 192.168.68.44 | 192.168.68.72 | DNS | 75 | Standard query 0xf0316f |
| 9 | 1.060606 | 192.168.68.72 | 192.168.68.44 | DNS | 107 | Standard query response 0x316f |
| 10 | 1.060606 | 192.168.68.72 | 192.168.68.44 | DNS | 107 | Standard query response 0xc316f |
| 48 | 1.187367 | 192.168.68.44 | 192.168.68.72 | DNS | 84 | Standard query 0x50316f |
| 49 | 1.187493 | 192.168.68.44 | 192.168.68.72 | DNS | 84 | Standard query 0xb0316f |
| 52 | 1.195079 | 192.168.68.72 | 192.168.68.44 | DNS | 84 | Standard query response 0x316f |
| 53 | 1.195079 | 192.168.68.72 | 192.168.68.44 | DNS | 100 | Standard query response 0xc316f |
| 56 | 1.209790 | 192.168.68.72 | 192.168.68.44 | DNS | 148 | Standard query response 0xf0316f |
| 93 | 1.239286 | 192.168.68.44 | 192.168.68.72 | DNS | 79 | Standard query 0xf0316f |
| 94 | 1.239471 | 192.168.68.44 | 192.168.68.72 | DNS | 79 | Standard query 0x20316f |
| 95 | 1.239849 | 192.168.68.44 | 192.168.68.72 | DNS | 74 | Standard query 0x316f |
| 96 | 1.239965 | 192.168.68.44 | 192.168.68.72 | DNS | 74 | Standard query 0x50316f |
| 97 | 1.246259 | 192.168.68.72 | 192.168.68.44 | DNS | 111 | Standard query response 0x316f |
| 98 | 1.246259 | 192.168.68.72 | 192.168.68.44 | DNS | 106 | Standard query response 0xc316f |
| 99 | 1.246259 | 192.168.68.72 | 192.168.68.44 | DNS | 74 | Standard query response 0xf0316f |
| 102 | 1.255776 | 192.168.68.44 | 192.168.68.72 | DNS | 76 | Standard query 0x60316f |
| 103 | 1.255924 | 192.168.68.44 | 192.168.68.72 | DNS | 76 | Standard query 0x90316f |
| 105 | 1.261910 | 192.168.68.72 | 192.168.68.44 | DNS | 192 | Standard query response 0x316f |
| 110 | 1.267962 | 192.168.68.72 | 192.168.68.44 | DNS | 121 | Standard query response 0xc316f |
| 166 | 1.406663 | 192.168.68.72 | 192.168.68.44 | DNS | 227 | Standard query response 0xf0316f |
| 409 | 28.315878 | 192.168.68.44 | 192.168.68.72 | DNS | 75 | Standard query 0xe0316f |

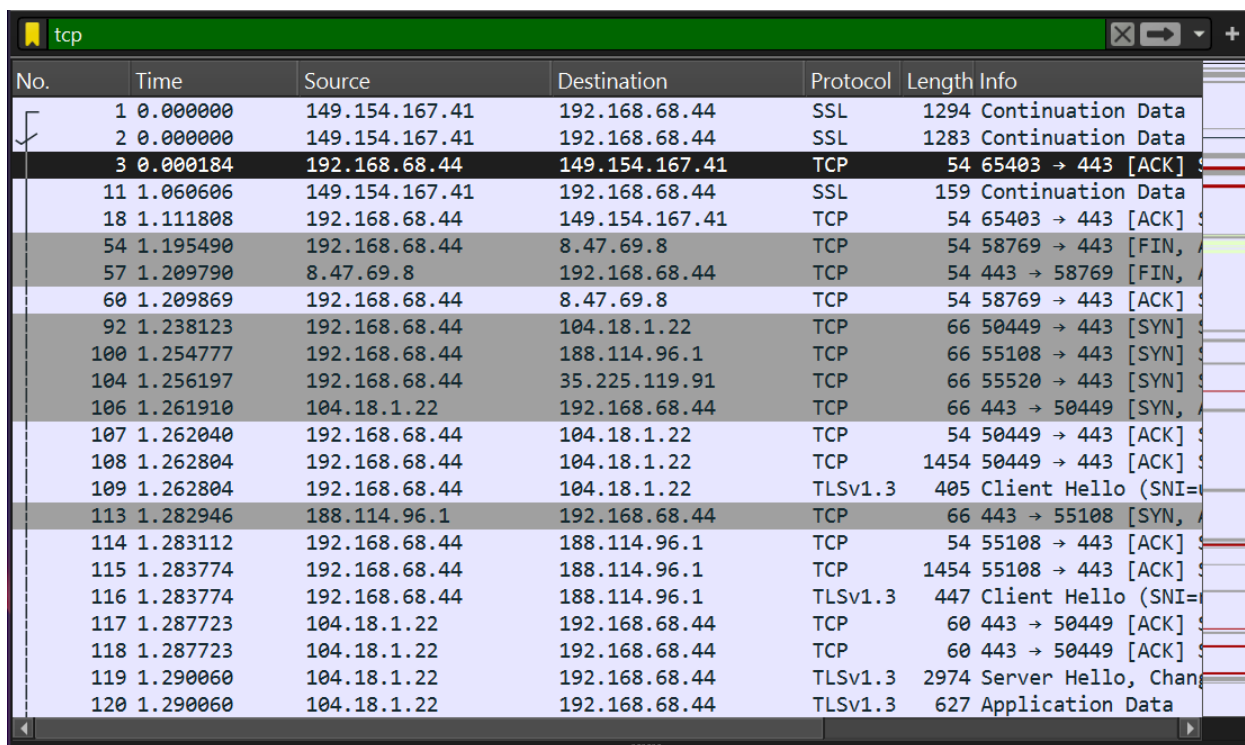
Domain Name System (query)
Transaction ID: 0x316f
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
baatraining.com: type A, class IN
Name: baatraining.com
[Name Length: 15]
[Label Count: 2]
Type: A (1) (Host Address)
Class: IN (0x0001)
[Response In: 9]

При обращении к сайту baatraining.com выполняется DNS запрос на получение IP – адреса, в запросе указан тип A (IPv4), в ответе возвращается ip адрес нужного сайта.

2.2 TCP

В фильтре Wireshark вводим:

tcp



| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|----------------|----------------|----------|--------|---------------------|
| 1 | 0.000000 | 149.154.167.41 | 192.168.68.44 | SSL | 1294 | Continuation Data |
| 2 | 0.000000 | 149.154.167.41 | 192.168.68.44 | SSL | 1283 | Continuation Data |
| 3 | 0.000184 | 192.168.68.44 | 149.154.167.41 | TCP | 54 | 65403 → 443 [ACK] |
| 11 | 1.060606 | 149.154.167.41 | 192.168.68.44 | SSL | 159 | Continuation Data |
| 18 | 1.111808 | 192.168.68.44 | 149.154.167.41 | TCP | 54 | 65403 → 443 [ACK] |
| 54 | 1.195490 | 192.168.68.44 | 8.47.69.8 | TCP | 54 | 58769 → 443 [FIN, A |
| 57 | 1.209790 | 8.47.69.8 | 192.168.68.44 | TCP | 54 | 443 → 58769 [FIN, A |
| 60 | 1.209869 | 192.168.68.44 | 8.47.69.8 | TCP | 54 | 58769 → 443 [ACK] |
| 92 | 1.238123 | 192.168.68.44 | 104.18.1.22 | TCP | 66 | 50449 → 443 [SYN] |
| 100 | 1.254777 | 192.168.68.44 | 188.114.96.1 | TCP | 66 | 55108 → 443 [SYN] |
| 104 | 1.256197 | 192.168.68.44 | 35.225.119.91 | TCP | 66 | 55520 → 443 [SYN] |
| 106 | 1.261910 | 104.18.1.22 | 192.168.68.44 | TCP | 66 | 443 → 50449 [SYN, A |
| 107 | 1.262040 | 192.168.68.44 | 104.18.1.22 | TCP | 54 | 50449 → 443 [ACK] |
| 108 | 1.262804 | 192.168.68.44 | 104.18.1.22 | TCP | 1454 | 50449 → 443 [ACK] |
| 109 | 1.262804 | 192.168.68.44 | 104.18.1.22 | TLSv1.3 | 405 | Client Hello (SNI= |
| 113 | 1.282946 | 188.114.96.1 | 192.168.68.44 | TCP | 66 | 443 → 55108 [SYN, A |
| 114 | 1.283112 | 192.168.68.44 | 188.114.96.1 | TCP | 54 | 55108 → 443 [ACK] |
| 115 | 1.283774 | 192.168.68.44 | 188.114.96.1 | TCP | 1454 | 55108 → 443 [ACK] |
| 116 | 1.283774 | 192.168.68.44 | 188.114.96.1 | TLSv1.3 | 447 | Client Hello (SNI= |
| 117 | 1.287723 | 104.18.1.22 | 192.168.68.44 | TCP | 60 | 443 → 50449 [ACK] |
| 118 | 1.287723 | 104.18.1.22 | 192.168.68.44 | TCP | 60 | 443 → 50449 [ACK] |
| 119 | 1.290060 | 104.18.1.22 | 192.168.68.44 | TLSv1.3 | 2974 | Server Hello, Chang |
| 120 | 1.290060 | 104.18.1.22 | 192.168.68.44 | TLSv1.3 | 627 | Application Data |

TCP – соединение устанавливается по стандартной трёхфазной схеме: клиент отправляет SYN, сервер отвечает SYN, если отправляется ACK, то и в ответ приходит ACK, таким образом устанавливается надежное соединение для дальнейшей передачи данных.

2.3 TLS-Handshake

В фильтре Wireshark вводим:

tls

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|----------------|---------------|----------|--------|---------------------|
| 1 | 0.000000 | 149.154.167.41 | 192.168.68.44 | SSL | 1294 | Continuation Data |
| 2 | 0.000000 | 149.154.167.41 | 192.168.68.44 | SSL | 1283 | Continuation Data |
| 11 | 1.060606 | 149.154.167.41 | 192.168.68.44 | SSL | 159 | Continuation Data |
| 13 | 1.075904 | 192.168.68.44 | 8.47.69.8 | QUIC | 1292 | Initial, DCID=a5506 |
| 16 | 1.111079 | 8.47.69.8 | 192.168.68.44 | QUIC | 1242 | Initial, SCID=01560 |
| 17 | 1.111079 | 8.47.69.8 | 192.168.68.44 | QUIC | 1242 | Handshake, SCID=015 |
| 51 | 1.187867 | 192.168.68.44 | 104.18.1.22 | QUIC | 1292 | Initial, DCID=eef53 |
| 58 | 1.209790 | 104.18.1.22 | 192.168.68.44 | QUIC | 1242 | Initial, SCID=01306 |
| 59 | 1.209790 | 104.18.1.22 | 192.168.68.44 | QUIC | 1242 | Initial, SCID=01306 |
| 109 | 1.262804 | 192.168.68.44 | 104.18.1.22 | TLSv1.3 | 405 | Client Hello (SNI= |
| 112 | 1.268865 | 192.168.68.44 | 31.13.72.36 | QUIC | 1292 | Initial, DCID=0e0c9 |
| 116 | 1.283774 | 192.168.68.44 | 188.114.96.1 | TLSv1.3 | 447 | Client Hello (SNI= |
| 119 | 1.290060 | 104.18.1.22 | 192.168.68.44 | TLSv1.3 | 2974 | Server Hello, Chang |
| 120 | 1.290060 | 104.18.1.22 | 192.168.68.44 | TLSv1.3 | 627 | Application Data |
| 122 | 1.292193 | 192.168.68.44 | 104.18.1.22 | TLSv1.3 | 118 | Change Cipher Spec |
| 123 | 1.292305 | 192.168.68.44 | 104.18.1.22 | TLSv1.3 | 146 | Application Data |
| 124 | 1.292413 | 192.168.68.44 | 104.18.1.22 | TLSv1.3 | 493 | Application Data |
| 126 | 1.318508 | 104.18.1.22 | 192.168.68.44 | TLSv1.3 | 566 | Application Data, A |
| 128 | 1.318508 | 104.18.1.22 | 192.168.68.44 | TLSv1.3 | 85 | Application Data |
| 129 | 1.318508 | 188.114.96.1 | 192.168.68.44 | TLSv1.3 | 2974 | Server Hello, Chang |
| 130 | 1.318508 | 188.114.96.1 | 192.168.68.44 | TLSv1.3 | 588 | Application Data |
| 133 | 1.319424 | 192.168.68.44 | 104.18.1.22 | TLSv1.3 | 85 | Application Data |
| 134 | 1.321833 | 192.168.68.44 | 188.114.96.1 | TLSv1.3 | 118 | Change Cipher Spec |

После установления TCP-соединения начинается TLS-рукопожатие, в ходе которого клиент и сервер договариваются о параметрах шифрования. Сервер передает сертификат подтверждающий подлинность сайта.

2.4 HTTP

Клиент отправляет запрос GET /, сервер отвечает HTTP/1.1 200 OK, это означает что сейчас будет открыта HTML страница и соединение установлено успешно, так же в этот момент передаются вспомогательные файлы css, js, изображения.

2.5 ARP

ARP используется для определения физического MAC адреса устройства в локальной сети.

Клиент отправляет запрос: Кто имеет IP-адрес шлюза?

Приходит ответ с MAC-адресом маршрутизатора.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|------------------------|------------------------|----------|--------|--------------------|
| 4 | 0.814310 | TpLinkTechno_a0:92:... | Broadcast | ARP | 60 | Who has 192.168.68 |
| 281 | 2.555052 | iRobot_92:7f:e9 | Broadcast | ARP | 60 | Who has 192.168.68 |
| 294 | 5.992887 | CloudNetwork_bf:31:... | Synology_36:b8:29 | ARP | 42 | Who has 192.168.68 |
| 295 | 5.999537 | Synology_36:b8:29 | CloudNetwork_bf:31:... | ARP | 60 | 192.168.68.72 is a |
| 323 | 17.506215 | TpLinkTechno_a0:92:... | Broadcast | ARP | 42 | Who has 192.168.68 |
| 340 | 19.656559 | TpLinkTechno_a0:92:... | Broadcast | ARP | 42 | Who has 192.168.68 |
| 352 | 20.885401 | TpLinkTechno_a0:92:... | Broadcast | ARP | 42 | Who has 192.168.68 |
| 364 | 22.627597 | TpLinkTechno_a0:92:... | CloudNetwork_bf:31:... | ARP | 42 | Who has 192.168.68 |
| 365 | 22.627615 | CloudNetwork_bf:31:... | TpLinkTechno_a0:92:... | ARP | 42 | 192.168.68.44 is a |
| 394 | 26.824935 | iRobot_92:7f:e9 | Broadcast | ARP | 60 | Who has 192.168.68 |
| 431 | 29.589905 | TpLinkTechno_a0:92:... | Broadcast | ARP | 60 | Who has 192.168.68 |
| 459 | 32.764300 | TpLinkTechno_a0:92:... | Broadcast | ARP | 60 | Who has 192.168.68 |
| 460 | 32.986085 | CloudNetwork_bf:31:... | Synology_36:b8:29 | ARP | 42 | Who has 192.168.68 |
| 461 | 32.994015 | Synology_36:b8:29 | CloudNetwork_bf:31:... | ARP | 60 | 192.168.68.72 is a |
| 465 | 35.425957 | TpLinkTechno_a0:92:... | Broadcast | ARP | 60 | Who has 192.168.68 |
| 466 | 35.426000 | CloudNetwork_bf:31:... | TpLinkTechno_a0:92:... | ARP | 42 | 192.168.68.44 is a |
| 498 | 38.088952 | TpLinkTechno_a0:92:... | Broadcast | ARP | 60 | Who has 192.168.68 |
| 565 | 41.878835 | Synology_36:b8:29 | CloudNetwork_bf:31:... | ARP | 60 | Who has 192.168.68 |
| 566 | 41.878885 | CloudNetwork_bf:31:... | Synology_36:b8:29 | ARP | 42 | 192.168.68.44 is a |
| 604 | 43.310999 | TpLinkTechno_a0:92:... | Broadcast | ARP | 60 | Who has 192.168.68 |
| 653 | 46.281459 | TpLinkTechno_a0:92:... | Broadcast | ARP | 60 | Who has 192.168.68 |
| 696 | 50.070450 | TpLinkTechno_a0:92:... | Broadcast | ARP | 60 | Who has 192.168.68 |
| 769 | 54.371277 | TpLinkTechno_a0:92:... | Broadcast | ARP | 60 | Who has 192.168.68 |

2.6 ICMP

| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|-----------|---------------|---------------|----------|--------|---------------------|
| → 415 | 28.477107 | 192.168.68.44 | 8.47.69.8 | ICMP | 142 | Echo (ping) request |
| ← 416 | 28.491510 | 8.47.69.8 | 192.168.68.44 | ICMP | 142 | Echo (ping) reply |
| 429 | 29.493149 | 192.168.68.44 | 8.47.69.8 | ICMP | 142 | Echo (ping) request |
| 430 | 29.508671 | 8.47.69.8 | 192.168.68.44 | ICMP | 142 | Echo (ping) reply |
| 447 | 30.503971 | 192.168.68.44 | 8.47.69.8 | ICMP | 142 | Echo (ping) request |
| 448 | 30.520050 | 8.47.69.8 | 192.168.68.44 | ICMP | 142 | Echo (ping) reply |
| 456 | 31.521991 | 192.168.68.44 | 8.47.69.8 | ICMP | 142 | Echo (ping) request |
| 457 | 31.538185 | 8.47.69.8 | 192.168.68.44 | ICMP | 142 | Echo (ping) reply |
| 463 | 35.126994 | 192.168.68.44 | 8.47.69.8 | ICMP | 1042 | Echo (ping) request |
| 464 | 35.142864 | 8.47.69.8 | 192.168.68.44 | ICMP | 1042 | Echo (ping) reply |
| 467 | 36.139299 | 192.168.68.44 | 8.47.69.8 | ICMP | 1042 | Echo (ping) request |
| 468 | 36.155278 | 8.47.69.8 | 192.168.68.44 | ICMP | 1042 | Echo (ping) reply |
| 490 | 37.155769 | 192.168.68.44 | 8.47.69.8 | ICMP | 1042 | Echo (ping) request |
| 491 | 37.176010 | 8.47.69.8 | 192.168.68.44 | ICMP | 1042 | Echo (ping) reply |
| 499 | 38.162725 | 192.168.68.44 | 8.47.69.8 | ICMP | 1042 | Echo (ping) request |
| 500 | 38.176787 | 8.47.69.8 | 192.168.68.44 | ICMP | 1042 | Echo (ping) reply |
| 606 | 43.334440 | 192.168.68.44 | 8.47.69.8 | ICMP | 562 | Echo (ping) request |
| 608 | 43.348835 | 8.47.69.8 | 192.168.68.44 | ICMP | 562 | Echo (ping) reply |
| 629 | 44.340008 | 192.168.68.44 | 8.47.69.8 | ICMP | 562 | Echo (ping) request |
| 631 | 44.355707 | 8.47.69.8 | 192.168.68.44 | ICMP | 562 | Echo (ping) reply |
| 642 | 45.353653 | 192.168.68.44 | 8.47.69.8 | ICMP | 562 | Echo (ping) request |
| 644 | 45.369746 | 8.47.69.8 | 192.168.68.44 | ICMP | 562 | Echo (ping) reply |
| 656 | 46.358365 | 192.168.68.44 | 8.47.69.8 | ICMP | 562 | Echo (ping) request |

При увеличении размера ICMP-пакета наблюдается фрагментация IP-датаграмм.

Максимальный размер без фрагментации – 1500 байт

2.7 nslookup

в командной строке выполняем команды:

```
nslookup baatraining.com
```

nslookup -type=NS baatraining.com

- При выполнении первой команды nslookup baatraining.com был отправлен запрос **типа А** (Address), предназначенный для получения IPv4-адреса веб-сайта.

В ответе от DNS-сервера были получены записи:

Name: baatraining.com

Address: 104.26.7.125

Address: 104.26.6.125

Эти IP-адреса принадлежат инфраструктуре Cloudflare, через которую обслуживается сайт.

- При выполнении второй команды nslookup -type=NS baatraining.com был отправлен запрос **типа NS** (Name Server), возвращающий список авторитативных DNS-серверов домена.

В ответ были получены записи:

baatraining.com nameserver = ray.ns.cloudflare.com

baatraining.com nameserver = dawn.ns.cloudflare.com

В заголовке DNS-пакетов в Wireshark видно:

Source IP: IP-адрес моего компьютера (например, 192.168.0.102)

Destination IP: адрес DNS-сервера (например, 192.168.0.1 — локальный маршрутизатор)

Protocol: UDP

Destination Port: 53

3 ОТВЕТЫ НА ВОПРОСЫ

3.1 Имеет ли место фрагментация исходного пакета?

Да, фрагментация возникает, когда размер пакета превышает **MTU (Maximum Transmission Unit)**, обычно равный 1500 байт.

Пакеты `ping -l 100` и `ping -l 1000` проходят без фрагментации, а начиная с `ping -l 2000` пакет разбивается на несколько частей.

На фрагментацию указывают поля:

- **Flags: More fragments = 1** — пакет не последний;
- **Fragment offset > 0** — смещение следующего фрагмента.

3.2 Какая информация указывает, является ли фрагмент последним или промежуточным?

Поле **More Fragments** в заголовке IP.

- Если **Set**, значит фрагмент **промежуточный**.
- Если **Not set**, но **Offset > 0** — это **последний** фрагмент.

3.3 Чему равно количество фрагментов при передаче ping-пакетов?

Пример расчёта при **MTU = 1500 байт**:

(1480 — полезная нагрузка на фрагмент)

Размер пакета – кол-во фрагментов

100 – 1

1000 – 1

2000 – 2

4000 – 3

8000 – 6

3.4 Как изменить поле TTL?

Команда:

```
ping -i <значение> baatraining.com
```

Например:

```
ping -i 10 baatraining.com
```

установит `TTL = 10`.

3.5 Что содержится в поле данных ping-пакета?

В Windows по умолчанию поле данных содержит повторяющуюся последовательность символов латинского алфавита (ASCII):

```
abcdefghijklmnopqrstuvwabcdefghi...
```

которая заполняет пространство до указанного размера пакета.

3.6 Сколько байт в заголовке IP и в данных?

- **Заголовок IP:** 20 байт (без опций).
- **Поле данных:** 64 байта (в ICMP Echo Request).

3.7 Как и почему изменяется TTL в последовательных ICMP-пакетах tracer?

tracert отправляет серию пакетов с `TTL = 1, 2, 3...`

Каждый маршрутизатор уменьшает TTL на единицу.

Когда `TTL = 0`, маршрутизатор отбрасывает пакет и отправляет обратно ICMP-сообщение

Time to live exceeded.

Это позволяет определить IP каждого промежуточного узла.

3.8 Чем отличаются ICMP-пакеты tracer и ping?

- **ping** использует фиксированный TTL (обычно 128) и ждёт Echo Reply.
 - **tracert** искусственно снижает TTL (1, 2, 3 ...), чтобы получить от каждого маршрутизатора Time Exceeded.
- Обе утилиты используют ICMP, но цели разные: ping проверяет доступность, tracert строит маршрут.

3.9 Чем отличаются ICMP reply и ICMP error?

- **ICMP Reply** (Echo Reply, тип 0) — нормальный ответ от конечного хоста.
 - **ICMP Error** (Time Exceeded, тип 11) — сообщает, что TTL пакета истёк на промежуточном узле.
- Tracert использует **оба** типа ответов: первые показывают маршрут, последний — достижение цели.

3.10 Что изменится, если убрать ключ -d?

Tracert начнёт отправлять **обратные DNS-запросы** (PTR-записи), чтобы узнать доменные имена маршрутизаторов по их IP. Это создаст дополнительный **DNS-трафик**.

3.11 Как выглядит HTTP-запрос и ответ?

Видна последовательность:

```
Client Hello → Server Hello → Certificate → Encrypted Application Data
```

Обычных строк GET и HTTP/1.1 200 OK нет, так как данные шифруются.

3.12 Что означает строка состояния HTTP 200 OK?

Эта строка указывает, что запрос клиента успешно обработан, и сервер отправил содержимое страницы.

(Если бы сайт использовал HTTP, а не HTTPS, в Wireshark было бы видно HTTP/1.1 200 OK.)

3.13 Что происходит при повторном обращении к странице?

Браузер использует **кэш**.

Если данные не изменились, отправляется **Conditional GET** с заголовком:

```
If-Modified-Since: <дата>
```

Сервер отвечает:

```
304 Not Modified
```

и страница берётся из кэша.

3.14 Почему адрес DNS-запроса не совпадает с адресом сайта?

DNS-запрос идёт на **адрес DNS-сервера** (например, 192.168.0.1 или 8.8.8.8), а не на сайт baatraining.com.

Клиент спрашивает у DNS-сервера: «Какой IP у baatraining.com?», а уже затем обращается к этому IP.

3.15 Какие бывают типы DNS-запросов?

- A — IPv4-адрес;

- **AAAA** — IPv6-адрес;
- **CNAME** — псевдоним;

- **NS** — имя авторитативного DNS-сервера.

Для сайта `baatraining.com` зафиксированы запросы типа **A** и **AAAA**.

3.16 Почему браузер делает дополнительные DNS-запросы?

Потому что страница сайта подгружает ресурсы (скрипты, картинки) с других доменов (например, `cdn.ba.com`, `static.cloudflare.com`).

Каждый новый домен требует отдельного DNS-запроса.

3.17 Какие MAC-адреса есть в ARP-пакетах и что они означают?

- **Source MAC**: MAC твоего компьютера.
- **Destination MAC**: широковещательный (`ff:ff:ff:ff:ff:ff`) — в ARP Request.
- **Target MAC**: в ARP Reply — MAC твоего шлюза (роутера).
Таким образом, ARP определяет физический адрес устройства в локальной сети.

3.18 Почему в ARP-запросе указан IP-адрес источника?

Чтобы принимающее устройство знало, куда отправить ответ.

Например, твой компьютер указывает свой IP (`192.168.x.x`), а роутер использует его, чтобы вернуть ARP-ответ напрямую

3.19 Чем различаются DNS-запросы типа A и NS?

Запрос типа **A** возвращает IP-адреса хоста (ресурсные записи с типом `Address`).

Запрос типа **NS** возвращает имена серверов, обслуживающих доменную зону.

3.20 Что содержится в поле “Answers” DNS-ответа?

В поле *Answers* содержатся ресурсные записи (`Resource Records`), запрошенные клиентом.

Для запроса типа **A** — это IP-адреса,

для типа **NS** — имена серверов зоны (`ray.ns.cloudflare.com`, `dawn.ns.cloudflare.com`).

3.21 Является ли ответ DNS-сервера авторитативным?

Нет, флаг Authoritative Answer (AA) в заголовке DNS-пакета **сброшен**, что указывает на неавторитативный ответ.

Ответ предоставлен локальным кэширующим DNS-сервером (роутером).

3.22 Почему адрес DNS-сервера отличается от адреса сайта?

Потому что запрос направляется не на сам сайт, а на DNS-сервер, который выполняет разрешение имени в IP-адрес.

Сайт — это конечный ресурс, а DNS-сервер — посредник, выполняющий поиск IP по имени.

3.23 Что содержится в поле “Authority” DNS-ответа?

В поле *Authority* указаны авторитативные DNS-серверы, ответственные за зону `baatraining.com` —

в нашем случае это **`ray.ns.cloudflare.com`** и **`dawn.ns.cloudflare.com`**.

3.24 Что произойдёт, если очистить ARP-кэш?

При следующем обращении к интернету компьютер не найдёт сохранённый MAC-адрес шлюза и **выполнит новый ARP-запрос**.

В Wireshark сразу появится пакет вида:

```
Who has 192.168.0.1? Tell 192.168.0.100
```

ЗАКЛЮЧЕНИЕ

В ходе работы была проведена запись и анализ сетевого трафика при обращении к веб-сайту `baatraining.com`

Были исследованы DNS-, TCP-, TLS-, HTTP- и ARP-пакеты, установлены их структура и взаимодействие.

Анализ показал, что передача данных осуществляется по модели TCP/IP, а шифрование выполняется с использованием протокола TLS 1.3.

Программа Wireshark позволяет детально изучить каждый уровень сетевой модели и выявить особенности обмена данными в Интернет.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алиев Т.И., Соснин В.В., Шинкарук Д.Н. Компьютерные сети и телекоммуникации: задания и тесты. – СПб: Университет ИТМО – 2018. – 112 с.
2. Куроуз Дж. Ф., Росс К. В. Компьютерные сети: Нисходящий подход / пер. с англ. – 6-е изд. – М.: Эксмо, 2016. – 912 с.