

**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:

Ахраров Али, студент группы N3250



(подпись)

Проверил:

Есипов Д.А.

(отметка о выполнении)

(подпись)

Санкт-Петербург
2024 г.

СОДЕРЖАНИЕ

Введение.....	4
1 АНАЛИЗ ТРАФИКА УТИЛИТЫ PING	5
2 АНАЛИЗ ТРАФИКА УТИЛИТЫ TRACERT (TRACEROUTE).....	10
3 АНАЛИЗ HTTP-ТРАФИКА	13
4 АНАЛИЗ DNS-ТРАФИКА	15
5 АНАЛИЗ ARP-ТРАФИКА	18
6 АНАЛИЗ ТРАФИКА УТИЛИТЫ NSLOOKUP	20
7 АНАЛИЗ FTP-ТРАФИКА	22
8 АНАЛИЗ DHCP-ТРАФИКА	24
Заключение.....	27

ВВЕДЕНИЕ

Цель работы - исследование структуры протокольных блоков данных путем анализа реального сетевого трафика на компьютере студента с использованием свободно распространяемой утилиты Wireshark.

Для достижения этой цели необходимо выполнить следующие задачи:

- Установить программное обеспечение Wireshark;
- Проанализировать последовательности команд и назначение служебных данных;
- Освоить работу с фильтрами в Wireshark;
- Захватить и сохранить достаточное количество дампов пакетов;
- Ответить на поставленные вопросы.

1 АНАЛИЗ ТРАФИКА УТИЛИТЫ PING

Ping (Packet Internet Groper) — это утилита для измерения задержки и проверки доступности узлов в сети. Она работает, отправляя ICMP Echo Request сообщения на целевой узел и ожидая получения ICMP Echo Reply сообщений. Ping вычисляет задержку на основе времени отправки и получения ICMP сообщений.

Используемый сайт: <https://www.aar.com/>.

```
C:\Users\Али>ping -l 100 www.aar.com

Pinging aar.com [74.51.210.74] with 100 bytes of data:
Reply from 74.51.210.74: bytes=100 time=144ms TTL=48
Reply from 74.51.210.74: bytes=100 time=144ms TTL=48
Reply from 74.51.210.74: bytes=100 time=143ms TTL=48
Reply from 74.51.210.74: bytes=100 time=143ms TTL=48

Ping statistics for 74.51.210.74:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 143ms, Maximum = 144ms, Average = 143ms
```

Рисунок 1 – Пример использования утилиты ping

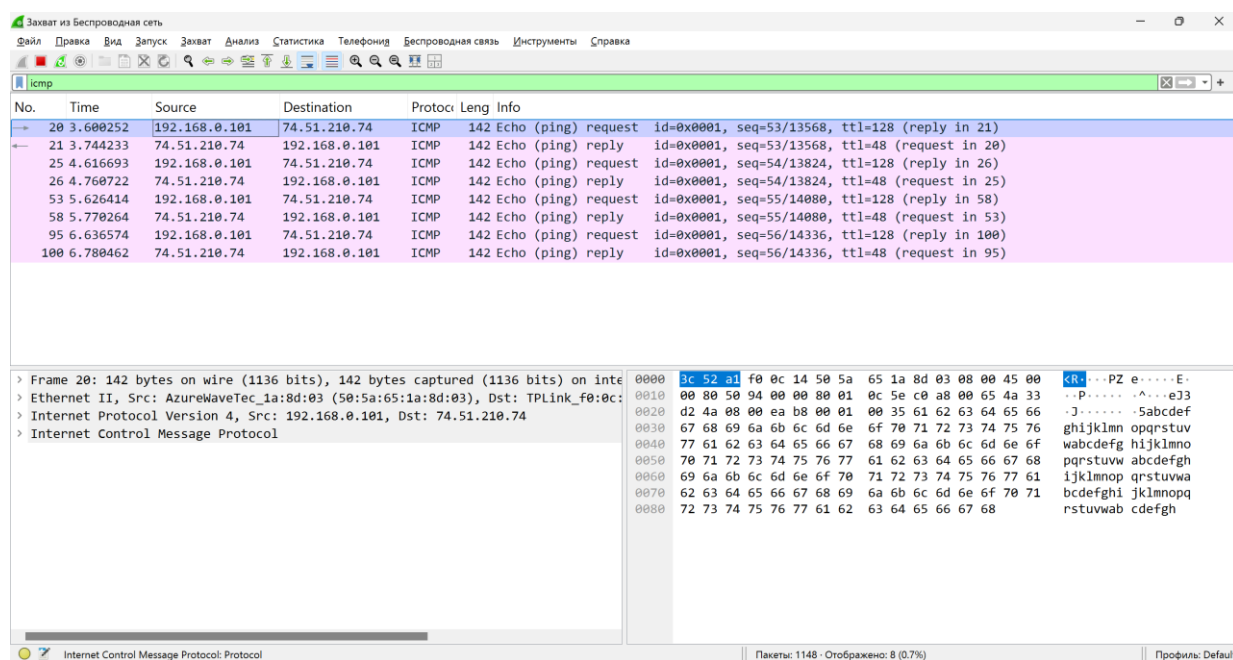


Рисунок 2 – Пример трафика утилиты ping

Ответы на вопросы:

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

```

v [7 IPv4 Fragments (10008 bytes): #12489(1480), #12490(1480), #12491(1480), #12492(1480), #12493(1480), #12494(1480), #12495(1128 bytes)]
[Frame: 12489, payload: 0-1479 (1480 bytes)]
[Frame: 12490, payload: 1480-2959 (1480 bytes)]
[Frame: 12491, payload: 2960-4439 (1480 bytes)]
[Frame: 12492, payload: 4440-5919 (1480 bytes)]
[Frame: 12493, payload: 5920-7399 (1480 bytes)]
[Frame: 12494, payload: 7400-8879 (1480 bytes)]
[Frame: 12495, payload: 8880-10007 (1128 bytes)]
[Fragment count: 7]
[Reassembled IPv4 length: 10008]
[Reassembled IPv4 data [...]: 08006d32000100396162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f90919293949596979899a0a1a2a3a4a5a6a7a8a9aaabacadaebaf0f1f2f3f4f5f6f7f8f9]
[Stream index: 4]

```

Фрагментация исходного пакета имеет место, что видно по разделу IPv4 Fragments. В новейшей версии Wireshark фрагментированные пакеты отображаются как единый пакет.

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

Установленный more fragments flag показывает, что пакет является промежуточным.

3. Чему равно количество фрагментов при передаче ring-пакетов? Количество фрагментов зависит от размера пакета и MTU сети (обычно 1500 байт для Ethernet). Количество фрагментов можно вычислить как:

$$\text{Количество фрагментов} = \frac{\text{Размер пакета}}{\text{MTU}} + 1$$

В нашем случае MTU равно 1480 байт.

4. Построить график, в котором на оси абсцисс находится размер_пакета, а по оси ординат – количество фрагментов, на которое был разделён каждый ring-пакет.

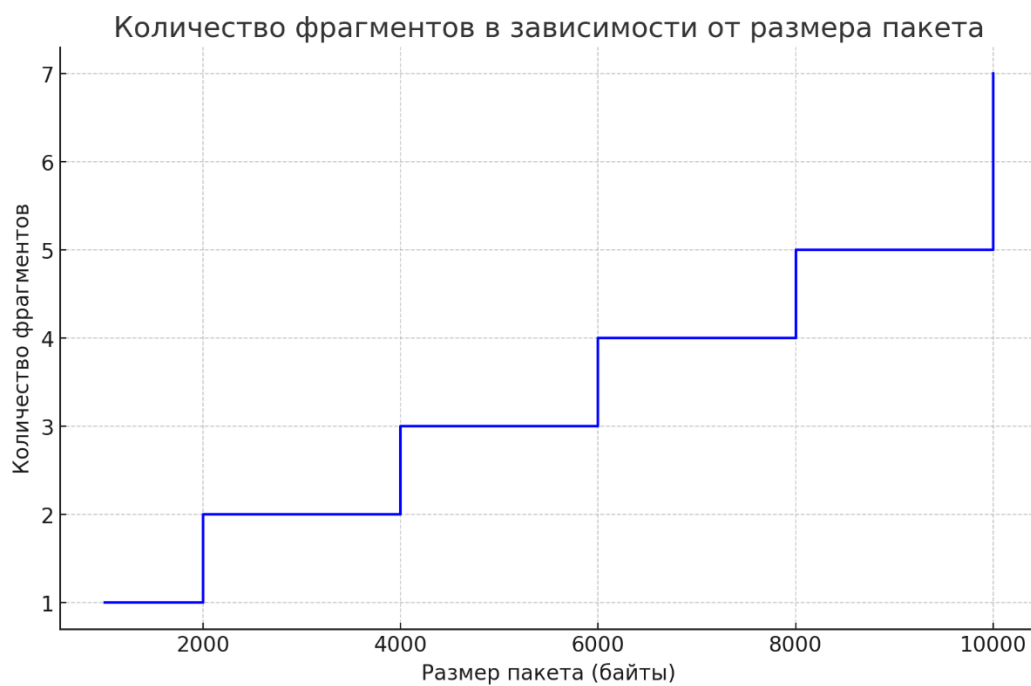


Рисунок 4 – График фрагментации

5. Как изменить поле TTL с помощью утилиты ping?

Листинг 1 – Изменение команды ping

ping -i <срок жизни в миллисекундах> <адрес>

```
C:\Users\Али>ping -l 100 -i 10 www.aar.com

Pinging aar.com [74.51.210.74] with 100 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 74.51.210.74:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Али>ping -l 100 -i 100 www.aar.com

Pinging aar.com [74.51.210.74] with 100 bytes of data:
Reply from 74.51.210.74: bytes=100 time=144ms TTL=48
Reply from 74.51.210.74: bytes=100 time=143ms TTL=48
Reply from 74.51.210.74: bytes=100 time=144ms TTL=48
Reply from 74.51.210.74: bytes=100 time=147ms TTL=48

Ping statistics for 74.51.210.74:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 143ms, Maximum = 147ms, Average = 144ms
```

Рисунок 5 – Изменение TTL

6. Что содержится в поле данных ping-пакета?

Поле данных ping обычно включает тестовую информацию — это может быть строка, состоящая из цифр, либо пустое поле, которое заполняется случайными байтами для измерения задержек в сети.

0010	04 7c 50 98 04 56 80 01	04 08 c0 a8 00 65 4a 33	· P..V.. ..eJ3
0020	d2 4a 72 73 74 75 76 77	61 62 63 64 65 66 67 68	·Jrstuvw abcdefgh
0030	69 6a 6b 6c 6d 6e 6f 70	71 72 73 74 75 76 77 61	ijklmnop qrstuvw
0040	62 63 64 65 66 67 68 69	6a 6b 6c 6d 6e 6f 70 71	bcdefghi jklmnopq
0050	72 73 74 75 76 77 61 62	63 64 65 66 67 68 69 6a	rstuvwab cdefghij
0060	6b 6c 6d 6e 6f 70 71 72	73 74 75 76 77 61 62 63	klmnopqr stuvwabc
0070	64 65 66 67 68 69 6a 6b	6c 6d 6e 6f 70 71 72 73	defghijk lmnopqrs
0080	74 75 76 77 61 62 63 64	65 66 67 68 69 6a 6b 6c	tuvwabcd efghijkl
0090	6d 6e 6f 70 71 72 73 74	75 76 77 61 62 63 64 65	mnopqrst uvwabcde
00a0	66 67 68 69 6a 6b 6c 6d	6e 6f 70 71 72 73 74 75	fghijklm nopqrstu
00b0	76 77 61 62 63 64 65 66	67 68 69 6a 6b 6c 6d 6e	vwabcdef ghijklmn
00c0	6f 70 71 72 73 74 75 76	77 61 62 63 64 65 66 67	opqrstuv wabcdefg
00d0	68 69 6a 6b 6c 6d 6e 6f	70 71 72 73 74 75 76 77	hijklmno pqrstuvw
00e0	61 62 63 64 65 66 67 68	69 6a 6b 6c 6d 6e 6f 70	abcdefgh ijklmnop
00f0	71 72 73 74 75 76 77 61	62 63 64 65 66 67 68 69	qrstuvw bcdefghi
0100	6a 6b 6c 6d 6e 6f 70 71	72 73 74 75 76 77 61 62	ijklmnopq rstuvwab
0110	63 64 65 66 67 68 69 6a	6b 6c 6d 6e 6f 70 71 72	cdefghij klmnopqr
0120	73 74 75 76 77 61 62 63	64 65 66 67 68 69 6a 6b	stuvwabc defghijk
0130	6c 6d 6e 6f 70 71 72 73	74 75 76 77 61 62 63 64	lmnopqrs tuvwabcd
0140	65 66 67 68 69 6a 6b 6c	6d 6e 6f 70 71 72 73 74	efghijkl mnopqrst
0150	75 76 77 61 62 63 64 65	66 67 68 69 6a 6b 6c 6d	uvwabcde fghijklm
0160	6e 6f 70 71 72 73 74 75	76 77 61 62 63 64 65 66	nopqrstu vwabcdef
0170	67 68 69 6a 6b 6c 6d 6e	6f 70 71 72 73 74 75 76	ghijklmn opqrstuv
0180	77 61 62 63 64 65 66 67	68 69 6a 6b 6c 6d 6e 6f	wabcdefg hijklmno
0190	70 71 72 73 74 75 76 77	61 62 63 64 65 66 67 68	pqrstuvw abcdefgh
01a0	69 6a 6b 6c 6d 6e 6f 70	71 72 73 74 75 76 77 61	ijklmnop qrstuvw
01b0	62 63 64 65 66 67 68 69	6a 6b 6c 6d 6e 6f 70 71	bcdefghi jklmnopq
01c0	72 73 74 75 76 77 61 62	63 64 65 66 67 68 69 6a	rstuvwab cdefghij
01d0	6b 6c 6d 6e 6f 70 71 72	73 74 75 76 77 61 62 63	klmnopqr stuvwabc
01e0	64 65 66 67 68 69 6a 6b	6c 6d 6e 6f 70 71 72 73	defghijk lmnopqrs
01f0	74 75 76 77 61 62 63 64	65 66 67 68 69 6a 6b 6c	tuvwabcd efghijkl
0200	6d 6e 6f 70 71 72 73 74	75 76 77 61 62 63 64 65	mnopqrst uvwabcde
0210	66 67 68 69 6a 6b 6c 6d	6e 6f 70 71 72 73 74 75	fghijklm nopqrstu
0220	76 77 61 62 63 64 65 66	67 68 69 6a 6b 6c 6d 6e	vwabcdef ghijklmn
0230	6f 70 71 72 73 74 75 76	77 61 62 63 64 65 66 67	opqrstuv wabcdefg
0240	68 69 6a 6b 6c 6d 6e 6f	70 71 72 73 74 75 76 77	hijklmno pqrstuvw
0250	61 62 63 64 65 66 67 68	69 6a 6b 6c 6d 6e 6f 70	abcdefgh ijklmnop
0260	71 72 73 74 75 76 77 61	62 63 64 65 66 67 68 69	qrstuvw bcdefghi
0270	6a 6b 6c 6d 6e 6f 70 71	72 73 74 75 76 77 61 62	ijklmnopq rstuvwab
0280	63 64 65 66 67 68 69 6a	6b 6c 6d 6e 6f 70 71 72	cdefghij klmnopqr

Рисунок 6 – Фрагмент данных ping-пакета

2 АНАЛИЗ ТРАФИКА УТИЛИТЫ TRACERT (TRACEROUTE)

Traceroute (tracert) — это утилита, предназначенная для определения маршрута прохождения данных в сети. Она отправляет пакеты на промежуточные узлы и отслеживает их путь обратно. Tracert постепенно увеличивает значение TTL в отправляемых пакетах, пока не достигнет целевого узла. Когда маршрутизатор получает пакет с TTL, равным 1, он возвращает его обратно, что позволяет tracert определить путь до этого маршрутизатора.

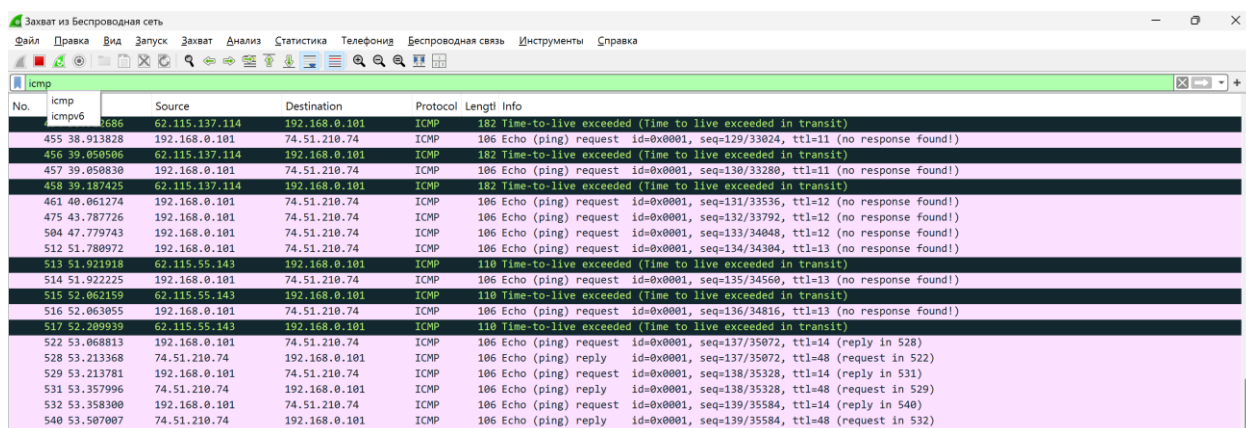
```
C:\Users\Али>tracert -d www.aar.com

Tracing route to aar.com [74.51.210.74]
over a maximum of 30 hops:

  1      1 ms      1 ms      2 ms    192.168.0.1
  2      5 ms     10 ms    106 ms   5.19.0.126
  3      4 ms      4 ms      4 ms    5.19.0.205
  4     13 ms      7 ms      8 ms    213.248.97.53
  5      4 ms      5 ms      5 ms    213.248.97.52
  6     18 ms     16 ms     16 ms    62.115.139.51
  7     21 ms     21 ms     21 ms    62.115.139.173
  8      *     122 ms    116 ms    80.91.254.91
  9      *      *      *      Request timed out.
 10     *      *      *      Request timed out.
 11    136 ms    136 ms    136 ms    62.115.137.114
 12     *      *      *      Request timed out.
 13    141 ms    139 ms    146 ms    62.115.55.143
 14    144 ms    144 ms    148 ms    74.51.210.74

Trace complete.
```

Рисунок 7 – Пример использования утилиты tracert



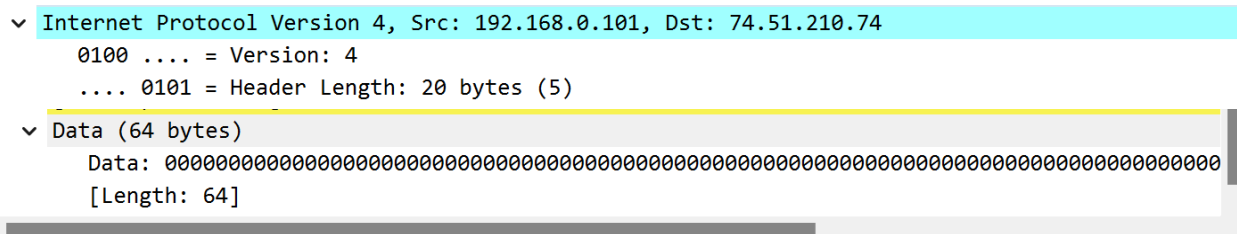
No.	icmp icmpv6	Source	Destination	Protocol	Length	Info
1686		62.115.137.114	192.168.0.101	ICMP	162	Time-to-live exceeded (Time to live exceeded in transit)
455	38.913828	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=129/33824, ttl=11 (no response found!)
456	39.050606	62.115.137.114	192.168.0.101	ICMP	162	Time-to-live exceeded (Time to live exceeded in transit)
457	39.050830	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=130/33280, ttl=11 (no response found!)
458	39.187425	62.115.137.114	192.168.0.101	ICMP	162	Time-to-live exceeded (Time to live exceeded in transit)
461	40.061274	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=131/33536, ttl=12 (no response found!)
475	43.787726	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=132/33792, ttl=12 (no response found!)
504	47.779743	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=133/34048, ttl=12 (no response found!)
512	51.780972	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=134/34304, ttl=13 (no response found!)
513	51.921918	62.115.55.143	192.168.0.101	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
514	51.922225	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=135/34560, ttl=13 (no response found!)
515	52.062159	62.115.55.143	192.168.0.101	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
516	52.063055	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=136/34816, ttl=13 (no response found!)
517	52.209939	62.115.55.143	192.168.0.101	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
522	53.068813	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=137/35072, ttl=14 (reply in 528)
528	53.213368	74.51.210.74	192.168.0.101	ICMP	106	Echo (ping) reply id=0x0001, seq=137/35072, ttl=48 (request in 522)
529	53.213781	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=138/35328, ttl=14 (reply in 531)
531	53.357996	74.51.210.74	192.168.0.101	ICMP	106	Echo (ping) reply id=0x0001, seq=138/35328, ttl=48 (request in 529)
532	53.358300	192.168.0.101	74.51.210.74	ICMP	106	Echo (ping) request id=0x0001, seq=139/35584, ttl=14 (reply in 540)
540	53.507007	74.51.210.74	192.168.0.101	ICMP	106	Echo (ping) reply id=0x0001, seq=139/35584, ttl=48 (request in 532)

Рисунок 8 - Пример трафика утилиты tracert

Ответы на вопросы:

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

20 байт в заголовке, 64 байта – в поле данных.



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

Чтобы ответить на этот вопрос, необходимо отследить изменение TTL при прохождении маршрута, содержащего более двух узлов. Tracert работает, увеличивая TTL (время жизни пакета) в IPv4, начиная с 1. Каждый раз, когда пакет достигает очередного узла, значение TTL возрастает на 1, пока не достигнет цели. Когда tracer отправляет пакет с TTL, равным 1, маршрутизатор по пути уменьшает TTL на 1 и пересылает его дальше. Если маршрутизатор получает пакет с TTL, равным нулю, он воспринимает это как ошибку и отправляет обратно сообщение ICMP (Internet Control Message Protocol) с кодом "Time-to-Live exceeded". Таким образом, tracer может отследить путь, который пакет проделал через сеть, до самого конечного узла.

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

Tracert использует протокол ICMP для определения маршрута к целевому узлу. Для этого он постепенно увеличивает значение TTL в отправляемых пакетах. Когда пакет с превышенным TTL достигает маршрутизатора, тот отправляет ответное сообщение с кодом "TTL expired", что позволяет tracer определить пройденный путь. Ping же используется для измерения времени отклика целевого узла. Он отправляет ICMP echo-request и ждет ICMP echo-reply, вычисляя время прохождения сигнала и задержку на основе времени

отправки и получения. Содержимое поля данных утилиты `tracert` обычно состоит из нулей, тогда как утилита `ping` использует алфавит.

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

Пакеты «ICMP error» отправляются, когда маршрутизатор обнаруживает, что время

жизни пакета (TTL) истекло

«ICMP reply» используется для проверки доступности удаленного узла. Если пакеты

«ICMP reply» получены, то узел считается доступным.

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. Увеличивает скорость вывода результатов

команды `tracert`.

```
C:\Users\Али>tracert www.aar.com

Tracing route to aar.com [74.51.210.74]
over a maximum of 30 hops:

  1     2 ms     4 ms     3 ms  192.168.0.1
  2    87 ms    22 ms     8 ms  5x19x0x126.static-business.spb.ertelecom.ru [5.19.0.126]
  3     4 ms     4 ms     4 ms  5x19x0x205.static-business.spb.ertelecom.ru [5.19.0.205]
  4    17 ms    17 ms    19 ms  ertelekom-ic-381104.ip.twelve99-cust.net [213.248.97.53]
  5     4 ms     4 ms     4 ms  sap-b5-link.ip.twelve99.net [213.248.97.52]
  6    17 ms    16 ms    16 ms  sto-bb2-link.ip.twelve99.net [62.115.139.51]
  7    21 ms    21 ms    21 ms  kbn-bb6-link.ip.twelve99.net [62.115.139.173]
  8   116 ms     *    116 ms  nyk-bb2-link.ip.twelve99.net [80.91.254.91]
  9      *      *      *    Request timed out.
 10     *      *      *    Request timed out.
 11   138 ms   141 ms   136 ms  den-bb2-link.ip.twelve99.net [62.115.137.114]
 12     *      *      *    Request timed out.
 13   140 ms   140 ms   139 ms  unite-ic-372992.ip.twelve99-cust.net [62.115.55.143]
 14   145 ms   150 ms   144 ms  74.51.210.74

Trace complete.
```

Рисунок 11 – Пример использования утилиты `tracert` без “-d”

3 АНАЛИЗ НТТР-ТРАФИКА

HTTP (Hypertext Transfer Protocol) — это протокол, используемый для передачи данных между веб-серверами и браузерами. Он работает на верхнем уровне модели OSI и используется для передачи HTML, CSS, JavaScript и других веб-ресурсов. HTTP поддерживает такие функции, как GET и POST запросы, cookies, сессии и т.д.

No.	Time	Source	Destination	Protocol	Length	Info
33185	1784.472591	192.168.0.101	185.218.1.126	HTTP	334	CONNECT jnn-pa.googleapis.com:443 HTTP/1.1
34239	1842.918128	185.218.1.126	192.168.0.101	TLSv1.3	334	Application Data, Application Data, Application Data, Application Data, Application Data, Application Data, Application Data
11915	1714.180616	185.218.1.126	192.168.0.101	TLSv1.3	336	Application Data, Application Data, Application Data
33159	1784.407159	192.168.0.101	185.218.1.126	TLSv1.3	337	Application Data
4984	477.640859	185.218.1.126	192.168.0.101	TLSv1.3	338	Application Data
11462	1686.129144	185.218.1.126	192.168.0.101	TLSv1.3	338	Application Data, Application Data, Application Data, Application Data
33902	1842.180363	192.168.0.101	185.218.1.126	HTTP	340	CONNECT files.osusernet.com:443 HTTP/1.1
6196	635.682836	192.168.0.101	185.218.1.126	TLSv1.3	341	Application Data
10160	1651.441054	23.196.236.82	192.168.0.101	HTTP	341	HTTP/1.1 204 No Content
32483	1779.183686	192.168.0.101	185.218.1.126	TLSv1.3	341	Application Data
+	98.6.862139	192.168.0.101	185.218.1.126	HTTP	342	CONNECT push.services.mozilla.com:443 HTTP/1.1
	115.7.042614	192.168.0.101	185.218.1.126	TCP	342	60317 → 6133 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=288
	188.17.620410	192.168.0.101	185.218.1.126	HTTP	342	CONNECT push.services.mozilla.com:443 HTTP/1.1
	3109.332.394287	185.218.1.126	192.168.0.101	TLSv1.3	342	Application Data
	3354.343.316675	185.218.1.126	192.168.0.101	TLSv1.3	342	Application Data
	6868.633.338798	192.168.0.101	185.218.1.126	HTTP	342	CONNECT push.services.mozilla.com:443 HTTP/1.1
	6986.633.338590	192.168.0.101	185.218.1.126	TCP	342	60470 → 531328 [ACK] Seq=1 Ack=1 Win=131328 Len=288
	6977.643.626690	192.168.0.101	185.218.1.126	HTTP	342	CONNECT push.services.mozilla.com:443 HTTP/1.1
	10725.1654.984405	192.168.0.101	185.218.1.126	HTTP	342	CONNECT push.services.mozilla.com:443 HTTP/1.1
	10751.1655.094427	192.168.0.101	185.218.1.126	TCP	342	60580 → 6133 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=288
	11013.1665.189006	192.168.0.101	185.218.1.126	HTTP	342	CONNECT push.services.mozilla.com:443 HTTP/1.1

Рисунок 12 – Фрагмент НТТР-трафика

No.	Time	Source	Destination	Protocol	Length	Info
1119	15.299108	192.168.0.101	192.168.149.23	HTTP	374	GET /SetigoECCDomainValidationSecureServerCA.crt HTTP/1.1
1122	15.318651	192.168.149.23	192.168.0.101	HTTP	1497	HTTP/1.1 200 OK (application/pkix-cert)
+	3401.22.075904	192.168.0.101	192.168.0.1	HTTP	257	GET /ukslm/gatedesc.xml HTTP/1.1
+	3407.22.080470	192.168.0.1	192.168.0.101	HTTP/X.	576	HTTP/1.1 200 OK
3493	26.221652	192.168.0.101	34.104.35.123	HTTP	314	HEAD /edgeid/dfiffgen-puffin/hfnpkplmhlgieaddgfemjhofmflmnb/cd2c85051a8a0c48212f0a37048849cfcf5d9d8706ed73af0669885.
3495	26.264914	34.104.35.123	192.168.0.101	HTTP	644	HTTP/1.1 200 OK

> Frame 3401: 257 bytes on wire (2056 bits), 257 bytes captured (2056 bits) on interface \Device\NPF	0000	3c 52 a1 f0 0c 14 50 5a	65 1a 8d 03 08 00 45 00	<R...P2 e...E-
> Ethernet II, Src: RealtekU_1a:8d:03 (50:5a:65:1a:8d:03), Dst: TPLink_F0:0c:14 (3c:52:a1:f0:0c	0010	00 f3 0d d1 70 00 80 06	6a 7f c8 08 00 65 c0 a8	...j...e...
> Internet Protocol Version 4, Src: 192.168.0.101, Dst: 192.168.0.1	0020	00 01 e4 0e 07 6c 54 a9	e1 eb 77 4b c5 6e 50 18	...IT...WK nP...
> 0100 = Version: 4	0030	02 01 15 9d 00 00 47 a5	54 20 2f 75 6b 73 6c 6d	...GE T /ukslm
> 0110 = Header Length: 20 bytes (5)	0040	6d 2f 67 61 74 65 64 65	73 63 2e 78 6d 6c 20 48	m/gatede sc.xml H
> 0120 = Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	0050	54 54 50 2f 31 2e 31 0d	0a 43 61 63 68 65 2d 43	TP/1.1 - Cache-C
> Total Length: 243	0060	6f 6e 74 72 6f 6c 3a 20	6e 6f 2d 63 61 63 68 65	ontrol: no-cache
> Identification: 0x0dd1 (3537)	0070	0d 0a 43 6f 6e 6e 65 63	74 69 6f 6e 3a 20 43 6c	..Connec tion: Cl
> 0130 = Flags: 0x2, Don't fragment	0080	6f 73 65 0d 0a 50 72 61	67 6d 61 3a 20 6e 6f 2d	ose -Pra gma: no-
> 0... = Reserved bit: Not set	0090	63 61 63 68 65 0d 0a 41	63 63 65 78 74 3a 20 7d	cache -A ccept: t
> 1... = Don't fragment: Set	00a0	65 78 74 2f 78 6d 6c 2c	20 61 70 70 6c 69 63 61	ext/xml applica
> ..0... = More fragments: Not set	00b0	74 69 6f 6e 2f 78 6d 6c	0d 0a 68 67 73 74 3a 20	tion/xml ..Host:
> ...0 0000 0000 0000 = Fragment Offset: 0	00c0	31 39 32 2e 31 36 38 2e	30 2e 31 31 39 30 30	192.168.0.1:1900
> Time to Live: 128	00d0	0d 0a 55 73 65 72 d1 41	67 65 6e 74 3a 20 4d 69	-User-A gent: Ml
> Protocol: TCP (6)	00e0	63 72 6f 73 6f 66 5a 7d	57 69 6e 64 6f 77 73 7f	crosoft-W indows/
> Header Checksum: 0x6a7d [validation disabled]	00f0	31 30 2e 30 20 55 6e 50	6f 31 2e 30 0d 0a 0d	10.0 Upn P/1.0 ..
> [Header checksum status: Unverified]	0100	0a		
> Source Address: 192.168.0.101				
> Destination Address: 192.168.0.1				
> [Stream index: 9]				
> Transmission Control Protocol, Src Port: 58382, Dst Port: 1900, Seq: 1, Ack: 1, Len: 203				
> Hypertext Transfer Protocol				

Рисунок 13 – Пример GET-запроса

Первичный GET-запрос:

- при первом посещении сайта браузер отправляет обычный GET-запрос на сервер, и сервер возвращает полный HTTP-ответ, включая тело сообщения, которое

содержит все данные страницы (HTML, CSS, изображения и т.д.);

– заголовки запроса включают информацию о браузере, версии протокола HTTP, целевом ресурсе, языке и других настройках;

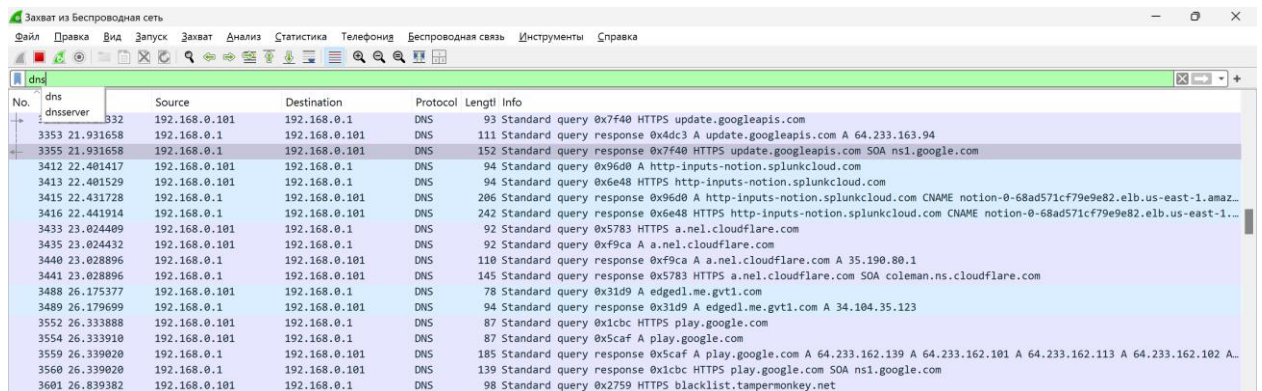
- в ответе сервера обычно содержится код состояния 200 ОК, а также тело сообщения, содержащее данные страницы.

Условный GET-запрос (Conditional GET):

- при повторной загрузке страницы, браузер отправляет условный GET-запрос, используя заголовки If-Modified-Since или If-None-Match для проверки, изменялся ли ресурс с момента последнего запроса;
- если ресурс не изменялся (содержимое страницы осталось прежним), сервер отвечает с кодом 304 Not Modified и не отправляет тело сообщения, что экономит трафик;
- если ресурс изменился, сервер вернет обновленную версию страницы с полным телом сообщения и кодом состояния 200 ОК.

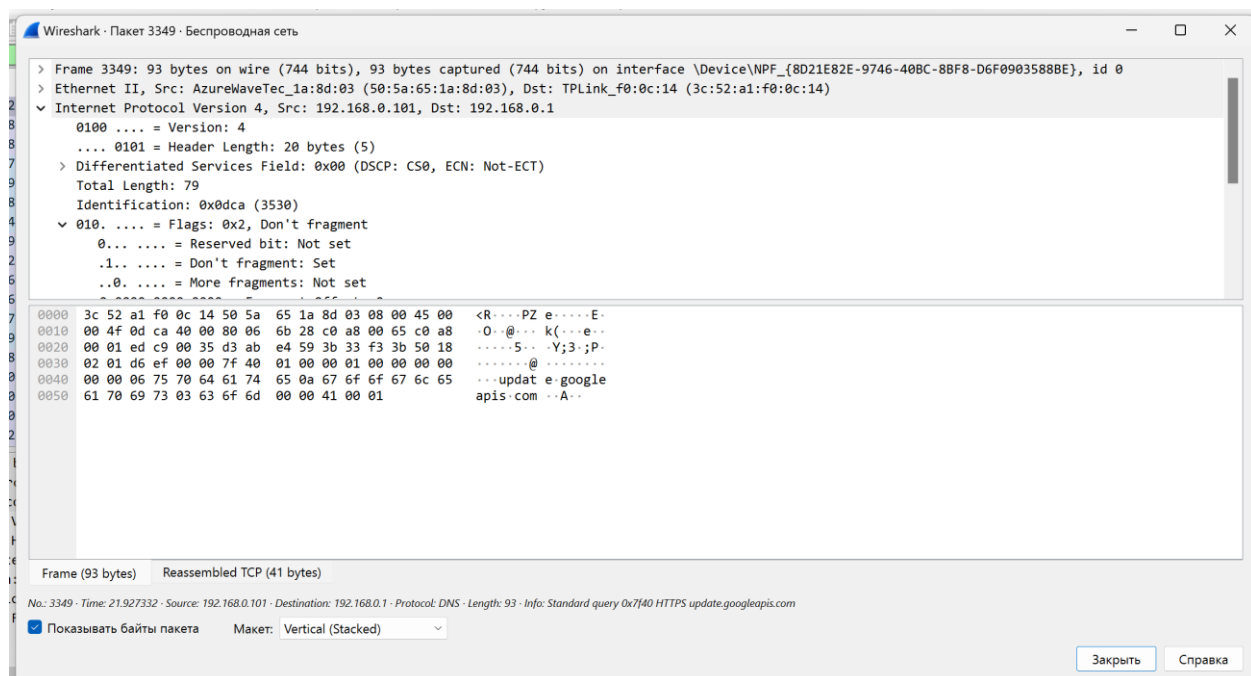
4 АНАЛИЗ DNS-ТРАФИКА

DNS (Domain Name System) — это система, которая переводит доменные имена в IP-адреса. Когда вы вводите доменное имя в адресной строке браузера, DNS-сервер ищет соответствующий IP-адрес, а затем направляет ваш запрос на нужный сервер. Протоколы DNS обеспечивают коммуникацию между DNS-клиентами (например, браузерами) и DNS-серверами. Они определяют, как запросы отправляются между клиентами и серверами, как ответы обрабатываются и как данные хранятся и обновляются. Рисунок 14 – Фрагмент DNS-трафика



No.	dns	dnsserver	Source	Destination	Protocol	Length	Info
332			192.168.0.101	192.168.0.1	DNS	93	Standard query 0x7f40 HTTPS update.googleapis.com
3353	21.931658		192.168.0.1	192.168.0.101	DNS	111	Standard query response 0x4dc3 A update.googleapis.com A 64.233.163.94
3355	21.931658		192.168.0.1	192.168.0.101	DNS	152	Standard query response 0x7f40 HTTPS update.googleapis.com SOA ns1.google.com
3412	22.401417		192.168.0.101	192.168.0.1	DNS	94	Standard query 0x96d0 A http-inputs-notion.splunkcloud.com
3413	22.401529		192.168.0.101	192.168.0.1	DNS	94	Standard query 0x6e48 HTTPS http-inputs-notion.splunkcloud.com
3415	22.431728		192.168.0.1	192.168.0.101	DNS	206	Standard query response 0x96d0 A http-inputs-notion.splunkcloud.com CNAME notion-0-68ad571cf79e9e82.elb.us-east-1.amaz...
3416	22.441914		192.168.0.1	192.168.0.101	DNS	242	Standard query response 0x6e48 HTTPS http-inputs-notion.splunkcloud.com CNAME notion-0-68ad571cf79e9e82.elb.us-east-1...
3433	23.024489		192.168.0.101	192.168.0.1	DNS	92	Standard query 0x5783 HTTPS a.nel.cloudflare.com
3435	23.024432		192.168.0.101	192.168.0.1	DNS	92	Standard query 0xf9ca A a.nel.cloudflare.com
3440	23.028896		192.168.0.1	192.168.0.101	DNS	110	Standard query response 0xf9ca A a.nel.cloudflare.com A 35.190.80.1
3441	23.028896		192.168.0.1	192.168.0.101	DNS	145	Standard query response 0x5783 HTTPS a.nel.cloudflare.com SOA coleman.ns.cloudflare.com
3488	26.175377		192.168.0.101	192.168.0.1	DNS	78	Standard query 0x31d9 A edgedl.me.gvt1.com
3489	26.179699		192.168.0.1	192.168.0.101	DNS	94	Standard query response 0x31d9 A edgedl.me.gvt1.com A 34.104.35.123
3552	26.333888		192.168.0.101	192.168.0.1	DNS	87	Standard query 0x1cbc HTTPS play.google.com
3554	26.333910		192.168.0.101	192.168.0.1	DNS	87	Standard query 0x5caf A play.google.com
3559	26.339020		192.168.0.1	192.168.0.101	DNS	185	Standard query response 0x5caf A play.google.com A 64.233.162.139 A 64.233.162.101 A 64.233.162.113 A 64.233.162.102 A...
3560	26.339020		192.168.0.1	192.168.0.101	DNS	139	Standard query response 0x1cbc HTTPS play.google.com SOA ns1.google.com
3601	26.839382		192.168.0.101	192.168.0.1	DNS	98	Standard query 0x2759 HTTPS blacklist.tampermonkey.net

Рисунок 14 – Фрагмент DNS-трафика



Wireshark · Пакет 3349 · Беспроводная сеть

> Frame 3349: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface \Device\NPF_{8D21E82E-9746-40BC-8BF8-D6F0903588BE}, id 0

> Ethernet II, Src: AzureWaveTec_1a:8d:03 (50:5a:65:1a:8d:03), Dst: TPLink_f0:0c:14 (3c:52:a1:f0:0c:14)

> Internet Protocol Version 4, Src: 192.168.0.101, Dst: 192.168.0.1

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 79

Identification: 0x0dca (3530)

> 010. = Flags: 0x2, Don't fragment

0... = Reserved bit: Not set

.1.. = Don't fragment: Set

..0. = More fragments: Not set

0000 3c 52 a1 f0 0c 14 50 5a 65 1a 8d 03 08 00 45 00 <R...PZ e.....E-

0010 00 4f 0d ca 40 00 80 06 6b 28 c0 a8 00 65 c0 a8 .O...@...k(...e-

0020 00 01 ed c9 00 35 d3 ab e4 59 3b 33 f3 3b 50 185...Y;3;P;

0030 02 01 d6 ef 00 00 7f 40 01 00 00 01 00 00 00 00@.....

0040 00 00 06 75 70 64 61 74 65 0a 67 6f 67 6c 65 ...updat e-google

0050 61 70 69 73 03 63 6f 6d 00 00 41 00 01 apis-com --A-

Frame (93 bytes) Reassembled TCP (41 bytes)

No.: 3349 · Time: 21.927332 · Source: 192.168.0.101 · Destination: 192.168.0.1 · Protocol: DNS · Length: 93 · Info: Standard query 0x7f40 HTTPS update.googleapis.com

☒ Показывать байты пакета Макет: Vertical (Stacked)

Закреть Справка

Рисунок 15 – Структура DNS

Ответы на вопросы:

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

DNS-запрос отправляется не на адрес сайта, который вы пытаетесь посетить а на DNS-сервер, который отвечает за преобразование доменного имени (например, example.com) в IP-адрес. Ваш компьютер не знает IP-адреса сайта до получения ответа от DNS-сервера, поэтому изначально отправляет запрос на IP-адрес DNS-сервера (например, на сервер вашего интернет-провайдера или общедоступные DNS-сервера, такие как Google DNS с адресом 8.8.8.8).

- 2. Какие бывают типы DNS-запросов?*

Основные типы DNS-запросов включают:

- А-запрос (Address Record): возвращает IPv4-адрес для указанного домена;
- AAAA-запрос: возвращает IPv6-адрес для указанного домена;
- CNAME-запрос (Canonical Name Record): возвращает каноническое (основное) доменное имя для указанного псевдонима;
- MX-запрос (Mail Exchange Record): возвращает адрес почтового сервера для указанного домена;
- NS-запрос (Name Server Record): возвращает список DNS-серверов, отвечающих за домен;
- PTR-запрос (Pointer Record): используется для обратного DNS, возвращая доменное имя для указанного IP-адреса;
- SOA-запрос (Start of Authority): возвращает информацию о DNS-зоне, включая основные данные о DNS-сервере.

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

Независимые DNS-запросы требуются в том случае, если изображения на сайте загружаются с других доменов, отличных от основного домена страницы. Например, если основной сайт запрашивается по адресу example.com, но изображения загружаются с

домена `images.example.com` или другого ресурса, тогда для каждого отдельного домена выполняется свой DNS-запрос. Это также происходит, если ресурсы, такие как изображения, загружаются с внешних источников (например, CDN или рекламных серверов).

5 АНАЛИЗ ARP-ТРАФИКА

ARP (Address Resolution Protocol) – протокол, который используется для определения MAC-адреса устройства по его IP-адресу в локальной сети. Он работает, отправляя широковещательные запросы на все устройства в сети, которые содержат IP-адрес устройства, ищущего MAC-адрес. Устройства, имеющие указанный IP-адрес, отвечают со своим MAC-адресом, и таким образом ARP определяет соответствие между IP- и MAC-адресами.

No.	Time	Source	Destination	Protocol	Length	Info
1243	16.463143	TPLink_f0:0c:14	AzureWaveTec_1a:8d:03	ARP	42	Who has 192.168.0.101? Tell 192.168.0.1
1244	16.463157	AzureWaveTec_1a:8d:03	TPLink_f0:0c:14	ARP	42	192.168.0.101 is at 50:5a:65:1a:8d:03
11534	74.416415	TPLink_f0:0c:14	AzureWaveTec_1a:8d:03	ARP	42	Who has 192.168.0.101? Tell 192.168.0.1
11535	74.416431	AzureWaveTec_1a:8d:03	TPLink_f0:0c:14	ARP	42	192.168.0.101 is at 50:5a:65:1a:8d:03
19762	97.840387	TPLink_f0:0c:14	AzureWaveTec_1a:8d:03	ARP	42	Who has 192.168.0.101? Tell 192.168.0.1
19763	97.840409	AzureWaveTec_1a:8d:03	TPLink_f0:0c:14	ARP	42	192.168.0.101 is at 50:5a:65:1a:8d:03
20045	121.712239	TPLink_f0:0c:14	AzureWaveTec_1a:8d:03	ARP	42	Who has 192.168.0.101? Tell 192.168.0.1
20046	121.712255	AzureWaveTec_1a:8d:03	TPLink_f0:0c:14	ARP	42	192.168.0.101 is at 50:5a:65:1a:8d:03
22003	145.616800	TPLink_f0:0c:14	AzureWaveTec_1a:8d:03	ARP	42	Who has 192.168.0.101? Tell 192.168.0.1
22004	145.616816	AzureWaveTec_1a:8d:03	TPLink_f0:0c:14	ARP	42	192.168.0.101 is at 50:5a:65:1a:8d:03
22283	169.136742	TPLink_f0:0c:14	AzureWaveTec_1a:8d:03	ARP	42	Who has 192.168.0.101? Tell 192.168.0.1
22284	169.136757	AzureWaveTec_1a:8d:03	TPLink_f0:0c:14	ARP	42	192.168.0.101 is at 50:5a:65:1a:8d:03
23429	193.809071	TPLink_f0:0c:14	AzureWaveTec_1a:8d:03	ARP	42	Who has 192.168.0.101? Tell 192.168.0.1
23430	193.809089	AzureWaveTec_1a:8d:03	TPLink_f0:0c:14	ARP	42	192.168.0.101 is at 50:5a:65:1a:8d:03
23729	217.461012	TPLink_f0:0c:14	AzureWaveTec_1a:8d:03	ARP	42	Who has 192.168.0.101? Tell 192.168.0.1
23730	217.461029	AzureWaveTec_1a:8d:03	TPLink_f0:0c:14	ARP	42	192.168.0.101 is at 50:5a:65:1a:8d:03
23962	240.961570	TPLink_f0:0c:14	AzureWaveTec_1a:8d:03	ARP	42	Who has 192.168.0.101? Tell 192.168.0.1
23963	240.961590	AzureWaveTec_1a:8d:03	TPLink_f0:0c:14	ARP	42	192.168.0.101 is at 50:5a:65:1a:8d:03

Рисунок 16 – Фрагмент ARP-трафика

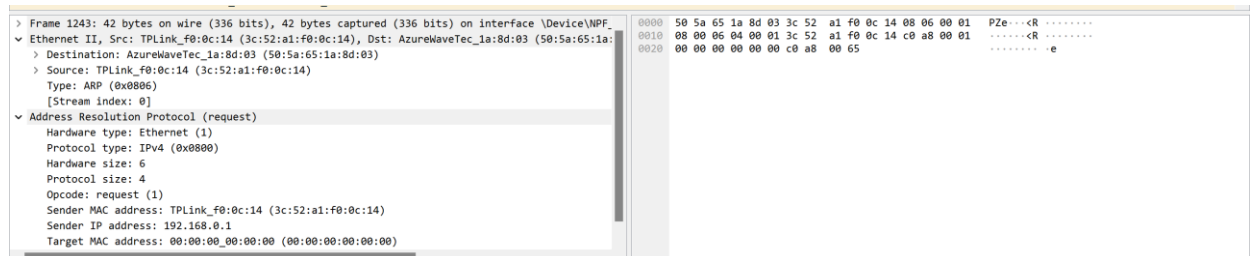


Рисунок 17 – Пример ARP-запроса

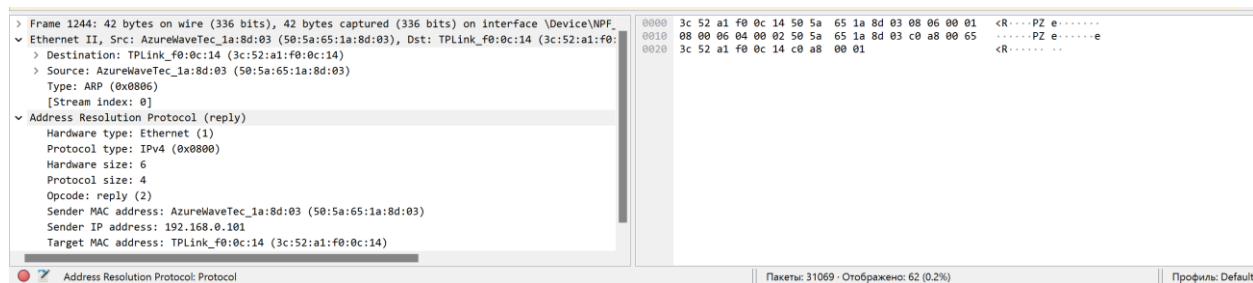


Рисунок 18 – Пример ARP-ответа

Ответы на вопросы:

1. *Какие MAC-адреса присутствуют в захваченных пакетах ARP-протокола? Что означают эти адреса? Какие устройства они идентифицируют? В ARP-протоколе обычно встречаются два типа MAC-адресов: MAC-адрес отправителя (Source MAC Address) и MAC-адрес получателя (Destination MAC Address).*

Source MAC Address — это MAC-адрес устройства, инициировавшего ARP-запрос (обычно это ваш компьютер или маршрутизатор). Destination MAC Address — это MAC-адрес устройства, к которому запрашивается связь (устройства, которое имеет указанный IP-адрес). Эти адреса идентифицируют физические устройства (сетевые интерфейсы) в локальной сети, такие как ваш компьютер, маршрутизатор или другой узел.

2. *Какие MAC-адреса присутствуют в захваченных HTTP-пакетах и что означают эти адреса? Какие устройства они идентифицируют? В захваченных HTTP-пакетах также содержатся MAC-адреса отправителя и получателя.*

Source MAC Address в HTTP-пакете — это адрес сетевого интерфейса вашего устройства, которое отправляет запрос к серверу. Destination MAC Address — это адрес маршрутизатора (или другого сетевого устройства), через которое передается запрос на сервер. Эти адреса идентифицируют устройства в локальной сети или ближайшем сегменте сети, через который проходит HTTP-запрос.

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

ARP-запрос содержит IP-адрес источника для того, чтобы узел, который отправляет ARP-ответ, знал, кому именно отправлять ответное сообщение. Это позволяет получателю ARP-запроса узнать как MAC-адрес устройства, отправляющего запрос, так и его IP-адрес, чтобы затем связать их для дальнейшего общения в локальной сети.

6 АНАЛИЗ ТРАФИКА УТИЛИТЫ NSLOOKUP

NSLOOKUP — это утилита, которая позволяет через командную строку узнать содержимое DNS.

Она может помочь:

- узнать IP-адрес;
- узнать A, NS, SOA, MX-записи для домена.

```
C:\Users\Али>nslookup www.aar.com
Server: UnKnown
Address: 192.168.0.1

Non-authoritative answer:
Name: aar.com
Address: 74.51.210.74
Aliases: www.aar.com
```

Рисунок 19 – Использование nslookup

```
C:\Users\Али>nslookup -type=NS www.aar.com
Server: UnKnown
Address: 192.168.0.1

Non-authoritative answer:
www.aar.com canonical name = aar.com
aar.com nameserver = ns39.domaincontrol.com
aar.com nameserver = ns40.domaincontrol.com
```

Рисунок 20 – Использование nslookup с дополнительным параметром

no.	time	source	destination	protocol	length	info
518	14.559659	192.168.0.101	192.168.0.1	DNS	94	Standard query 0xb603 A http-inputs-notion.splunkcloud.com
519	14.559771	192.168.0.101	192.168.0.1	DNS	94	Standard query 0xa94a HTTPS http-inputs-notion.splunkcloud.com
521	14.596159	192.168.0.1	192.168.0.101	DNS	242	Standard query response 0xa94a HTTPS http-inputs-notion.splunkcloud.com CNAME notion-0-68ad571cf79e9e82.elb.us-east-1...
522	14.599650	192.168.0.1	192.168.0.101	DNS	206	Standard query response 0xb603 A http-inputs-notion.splunkcloud.com CNAME notion-0-68ad571cf79e9e82.elb.us-east-1.amaz...
809	44.592934	192.168.0.101	192.168.0.1	DNS	94	Standard query 0xd750 A p2p-sto2.discovery.steamserver.net
810	44.597959	192.168.0.1	192.168.0.101	DNS	142	Standard query response 0xd750 A p2p-sto2.discovery.steamserver.net A 155.133.252.54 A 155.133.252.39 A 155.133.252.40
997	56.373271	192.168.0.101	192.168.0.1	DNS	70	Standard query 0x7f83 A c.pki.goog
998	56.377685	192.168.0.1	192.168.0.101	DNS	121	Standard query response 0x7f83 A c.pki.goog CNAME pki-goog.l.google.com A 74.125.205.94
1047	70.335199	192.168.0.101	192.168.0.1	DNS	84	Standard query 0x0001 PTR 1.0.168.192.in-addr.arpa
1048	70.339623	192.168.0.1	192.168.0.101	DNS	143	Standard query response 0x0001 No such name PTR 1.0.168.192.in-addr.arpa SOA localhost
1049	70.347600	192.168.0.101	192.168.0.1	DNS	71	Standard query 0x0002 A www.aar.com
1050	70.458140	192.168.0.1	192.168.0.101	DNS	101	Standard query response 0x0002 A www.aar.com CNAME aar.com A 74.51.210.74
1051	70.459677	192.168.0.101	192.168.0.1	DNS	71	Standard query 0x0003 AAAA www.aar.com

Рисунок 21 – Трафик от nslookup

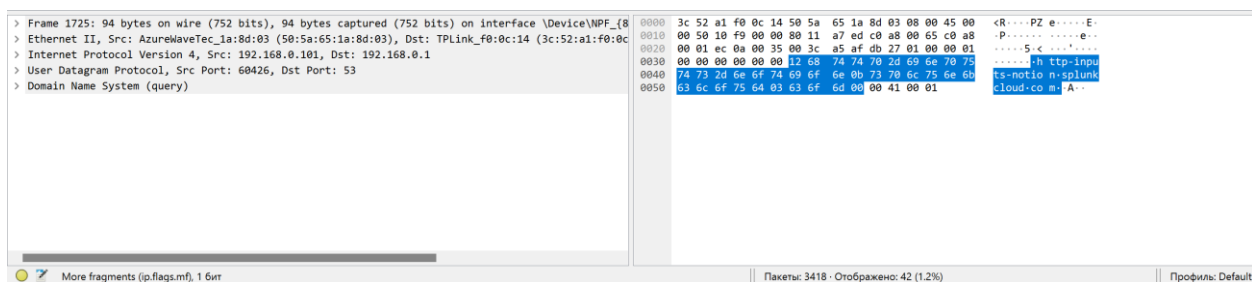


Рисунок 22 – Пример запроса DNS при действии nslookup

Ответы на вопросы

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

В п.2 утилита nslookup отправляет стандартный DNS-запрос, чтобы получить А-запись (адрес хоста) для указанного домена. В п.4 команда nslookup -type=NS запрашивает NS-записи, которые содержат имена DNS-серверов, ответственных за домен. Эти записи не содержат IP-адресов самого сайта, а указывают на серверы, отвечающие за его зону.

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

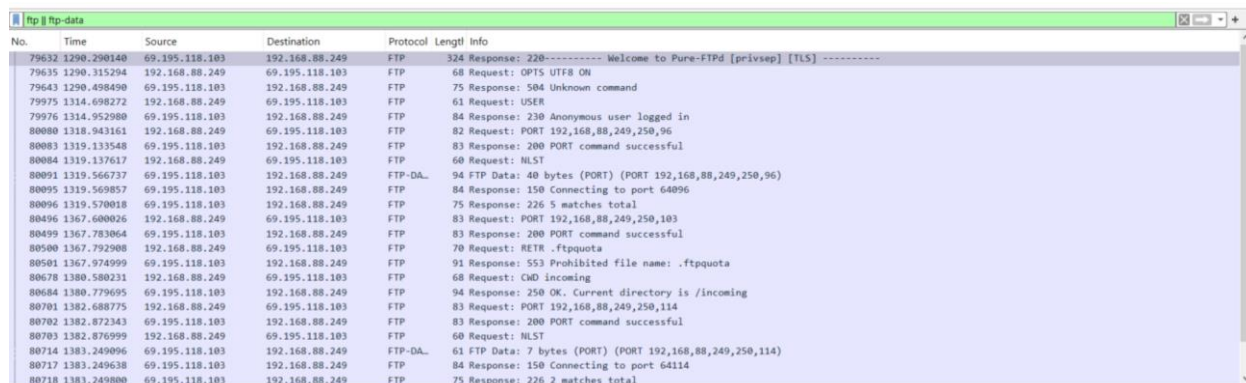
Поле Answers в DNS-ответе содержит информацию о запрашиваемом ресурсе. В случае стандартного запроса (п.2), это IP-адрес сайта (А-запись). В случае запроса на NS-записи (п.4), в поле Answers содержатся имена серверов, которые ответственны за зону указанного домена.

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

Имена серверов, дающих авторитативные ответы, находятся в поле Authority. Это серверы, которые ответственны за домен, к которому принадлежит запрашиваемый ресурс. Их можно найти в ответах на NS-запросы и в случае с А-запросами, если они обрабатываются авторитативным сервером.

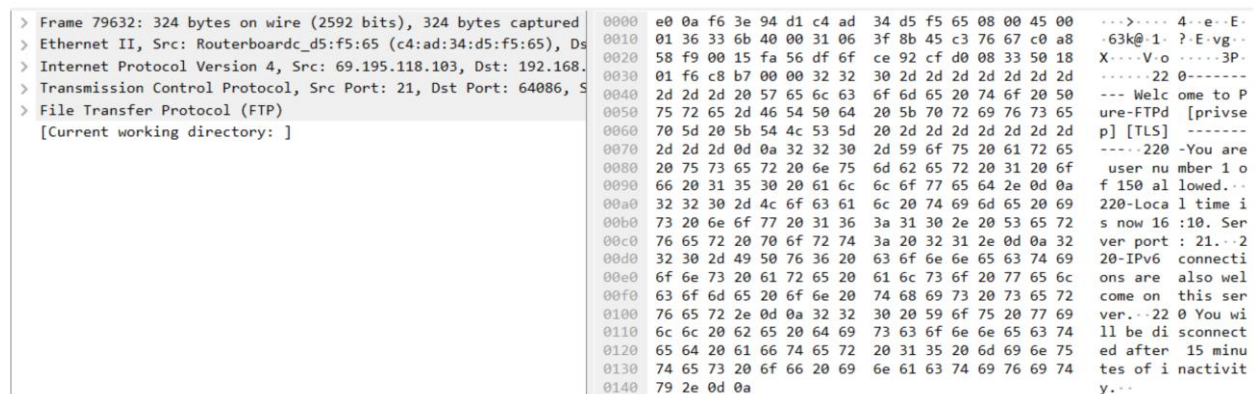
7 АНАЛИЗ FTP-ТРАФИКА

FTP (File Transfer Protocol) — это протокол, который предназначен для передачи файлов через Интернет или локальную компьютерную сеть. Основное назначение FTP — пересылать, копировать или передавать файл с удаленного компьютера на локальный и наоборот. Кроме того, при помощи FTP можно работать со своими файлами прямо на удаленном компьютере. Например, можно передать доступ к файлам или к части файлов своему разработчику, а он сможет переименовывать их, удалять или создавать каталоги.



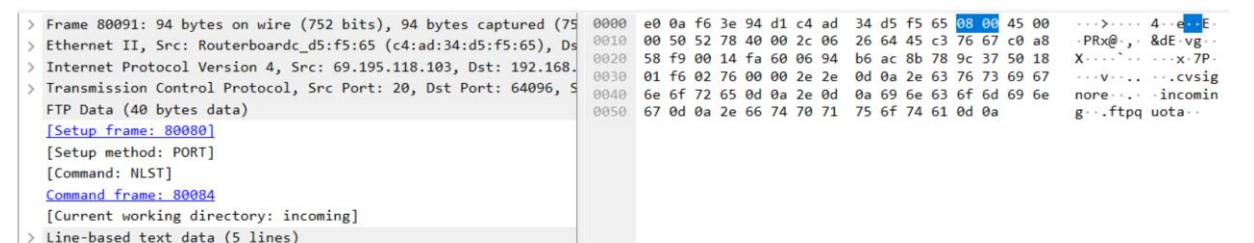
No.	Time	Source	Destination	Protocol	Length	Info
79632	1290.290140	69.195.118.103	192.168.88.249	FTP	324	Response: 220----- Welcome to Pure-FTPd [privsep] [TLS] -----
79635	1290.315294	192.168.88.249	69.195.118.103	FTP	68	Request: OPTS UTF8 ON
79643	1290.498490	69.195.118.103	192.168.88.249	FTP	75	Response: 504 Unknown command
79975	1314.608272	192.168.88.249	69.195.118.103	FTP	61	Request: USER
79976	1314.952980	69.195.118.103	192.168.88.249	FTP	84	Response: 230 Anonymous user logged in
80000	1318.943161	192.168.88.249	69.195.118.103	FTP	82	Request: PORT 192,168,88,249,250,96
80003	1319.133548	69.195.118.103	192.168.88.249	FTP	83	Response: 200 PORT command successful
80004	1319.137617	192.168.88.249	69.195.118.103	FTP	60	Request: NLST
80091	1319.566737	69.195.118.103	192.168.88.249	FTP-DA	94	FTP Data: 40 bytes (PORT) (PORT 192,168,88,249,250,96)
80095	1319.569857	69.195.118.103	192.168.88.249	FTP	84	Response: 150 Connecting to port 64096
80096	1319.570018	69.195.118.103	192.168.88.249	FTP	75	Response: 226 5 matches total
80496	1367.600026	192.168.88.249	69.195.118.103	FTP	83	Request: PORT 192,168,88,249,250,103
80499	1367.783064	69.195.118.103	192.168.88.249	FTP	83	Response: 200 PORT command successful
80500	1367.792908	192.168.88.249	69.195.118.103	FTP	70	Request: RETR .ftpquota
80501	1367.974999	69.195.118.103	192.168.88.249	FTP	91	Response: 553 Prohibited file name: .ftpquota
80678	1380.580231	192.168.88.249	69.195.118.103	FTP	68	Request: CWD incoming
80684	1380.779695	69.195.118.103	192.168.88.249	FTP	94	Response: 250 OK. Current directory is /incoming
80701	1382.688775	192.168.88.249	69.195.118.103	FTP	83	Request: PORT 192,168,88,249,250,114
80702	1382.872343	69.195.118.103	192.168.88.249	FTP	83	Response: 200 PORT command successful
80703	1382.876999	192.168.88.249	69.195.118.103	FTP	60	Request: NLST
80714	1383.249096	69.195.118.103	192.168.88.249	FTP-DA	61	FTP Data: 7 bytes (PORT) (PORT 192,168,88,249,250,114)
80717	1383.249638	69.195.118.103	192.168.88.249	FTP	84	Response: 150 Connecting to port 64114
80718	1383.249800	69.195.118.103	192.168.88.249	FTP	75	Response: 226 2 matches total

Рисунок 24 – Взаимодействие с публичным ftp-сервером



> Frame 79632: 324 bytes on wire (2592 bits), 324 bytes captured	0000	e0 0a f6 3e 94 d1 c4 ad 34 d5 f5 65 08 00 45 00	...>... 4...E...
> Ethernet II, Src: Routerboardc_d5:f5:65 (c4:ad:34:d5:f5:65), Dst: 08:00:00:00:00:00	0010	01 36 33 6b 40 00 31 06 3f 8b 45 c3 76 67 c0 a8	...63k@1...?Evg...
> Internet Protocol Version 4, Src: 69.195.118.103, Dst: 192.168.88.249	0020	58 f9 00 15 fa 56 df 6f ce 92 cf d0 08 33 50 18	X...V...3P...
> Transmission Control Protocol, Src Port: 21, Dst Port: 64086, Seq: 302044800	0030	01 f6 c8 b7 00 00 32 32 30 2d 2d 2d 2d 2d 2d 2d	...22 0-----
> File Transfer Protocol (FTP)	0040	2d 2d 2d 2d 2d 57 65 6c 63 6f 6d 65 20 74 6f 20 50	--- Welc ome to P
[Current working directory: /]	0050	75 72 65 2d 46 54 50 64 20 5b 70 72 69 76 73 65	ure-FTPd [privse
	0060	70 5d 20 5b 54 4c 53 5d 20 2d 2d 2d 2d 2d 2d 2d	p] [TLS] -----
	0070	2d 2d 2d 0d 0a 32 32 30 2d 59 6f 75 20 61 72 65	-----220 -You are
	0080	20 75 73 65 72 20 6e 75 6d 62 65 72 20 31 20 6f	user num ber 1 o
	0090	66 20 31 35 30 20 61 6c 6c 6f 77 65 64 2e 0d 0a	f 150 al lowed...
	00a0	32 32 30 2d 4c 6f 63 61 6c 20 74 69 6d 65 20 69	220-Loca l time i
	00b0	73 20 6e 6f 77 20 31 36 3a 31 30 2e 20 53 65 72	s now 16 :10. Ser
	00c0	76 65 72 20 70 6f 72 74 3a 20 32 31 2e 0d 0a 32	ver port : 21...2
	00d0	32 30 2d 49 50 76 36 20 63 6f 6e 6e 65 63 74 69	20-IPv6 connecti
	00e0	6f 6e 73 20 61 72 65 20 61 6c 73 6f 20 77 65 6c	ons are also wel
	00f0	63 6f 6d 65 20 6f 6e 20 74 68 69 73 20 73 65 72	come on this ser
	0100	76 65 72 2e 0d 0a 32 32 30 20 59 6f 75 20 77 69	ver...22 0 You wi
	0110	6c 6c 20 62 65 20 64 69 73 63 6f 6e 6e 65 63 74	ll be di sconnect
	0120	65 64 20 61 66 74 65 72 20 31 35 20 6d 69 6e 75	ed after 15 minu
	0130	74 65 73 20 6f 6e 20 69 6e 61 63 74 69 76 69 74	tes of i nactivit
	0140	79 2e 0d 0a	y...

Рисунок 25 – Ответ о подключении к серверу (ftp)



> Frame 80091: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0	0000	e0 0a f6 3e 94 d1 c4 ad 34 d5 f5 65 08 00 45 00	...>... 4...E...
> Ethernet II, Src: Routerboardc_d5:f5:65 (c4:ad:34:d5:f5:65), Dst: 08:00:00:00:00:00	0010	00 50 52 78 40 00 2c 06 26 64 45 c3 76 67 c0 a8	...PRx@...&dEvg...
> Internet Protocol Version 4, Src: 69.195.118.103, Dst: 192.168.88.249	0020	58 f9 00 14 fa 60 06 9a b6 ac 8b 78 9c 37 50 18	X...V...x-7P...
> Transmission Control Protocol, Src Port: 20, Dst Port: 64096, Seq: 302044800	0030	01 f6 02 76 00 00 2e 2e 0d 0a 2e 63 76 73 69 67	...V...cvsig
FTP Data (40 bytes data)	0040	6e 6f 72 65 0d 0a 2e 0d 0a 69 6e 63 6f 6d 69 6e	none...incomin
[Setup frame: 80080]	0050	67 0d 0a 2e 66 74 70 71 75 6f 74 61 0d 0a	g...ftqp uota...
[Setup method: PORT]			
[Command: NLST]			
Command frame: 80084			
[Current working directory: incoming]			
> Line-based text data (5 lines)			

Рисунок 26 – Получение данных с ftp-сервера (ftp-data)

Ответы на вопросы:

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

FTP-DATA — это пакеты, содержащие полезные данные, которые передаются в процессе передачи файлов через FTP. Размер данных в пакете может варьироваться, но обычно составляет 1460 байт, что связано с ограничением на максимальный размер сегмента (MSS) TCP для большинства сетей. Чтобы узнать точный размер данных в пакете, в Wireshark можно посмотреть поле "Length" в деталях пакета FTP-DATA. Оно указывает количество байт данных, переданных в данном пакете.

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

FTP использует два порта:

- порт 21 для команд управления (FTP control), используется для передачи управляющих команд между клиентом и сервером;
- порт 20 (или случайно выбранный порт в активном или пассивном режиме) для передачи данных (FTP-DATA). В активном режиме клиент сообщает серверу о порте, который нужно использовать для передачи данных, а в пассивном режиме сервер сообщает клиенту порт, который будет использоваться для передачи данных. В активном режиме клиент сообщает серверу, какой порт он должен использовать для передачи данных, а в пассивном режиме сервер выбирает случайный порт для передачи данных.

3. Чем отличаются пакеты FTP от FTP-DATA?

FTP-пакеты — это управляющие пакеты, которые используются для отправки команд между клиентом и сервером. Например, команды авторизации, навигации по директориям или запроса на скачивание файла. Эти пакеты передаются по порту

FTP-DATA — это пакеты, которые содержат полезные данные (например, части файлов), передаваемые между клиентом и сервером. Эти пакеты передаются по порту 20 или случайному порту в пассивном режиме. Основное различие заключается в том, что FTP-пакеты используются для управления сессией и отправки команд, а FTP-DATA содержат фактические данные, передаваемые между клиентом и сервером.

8 АНАЛИЗ DHCP-ТРАФИКА

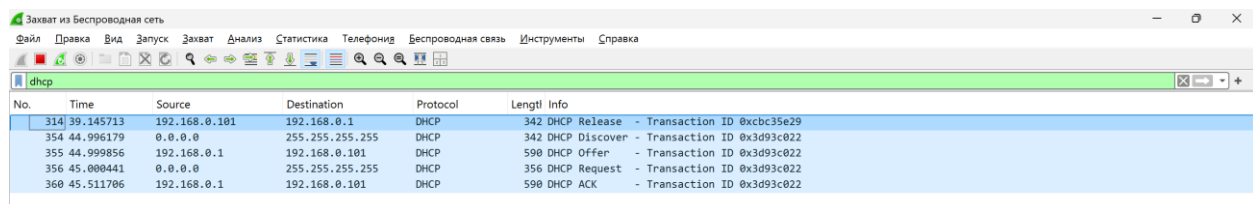
DHCP (Dynamic Host Configuration Protocol) — это сетевой протокол, используемый для автоматического назначения IP-адресов и других сетевых параметров (например, шлюза, маски подсети и DNS-серверов) устройствам в сети.

Этот процесс позволяет упростить управление IP-адресами и уменьшить вероятность конфликта IP-адресов, когда два устройства случайно получают один и тот же IP-адрес.

```
Wireless LAN adapter Беспроводная сеть:

Connection-specific DNS Suffix . : 
Description . . . . . : MediaTek Wi-Fi 6E MT7922 160MHz Wireless LAN Card
Physical Address. . . . . : 50-5A-65-1A-8D-03
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . : fe80::4f86:43e6:88bb:b97c%16(Preferred)
IPv4 Address. . . . . : 192.168.0.101(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 9 ноября 2024 г. 15:23:54
Lease Expires . . . . . : 10 ноября 2024 г. 15:23:50
Default Gateway . . . . . : 192.168.0.1
DHCP Server . . . . . : 192.168.0.1
DHCPv6 IAID . . . . . : 340810341
DHCPv6 Client DUID. . . . . : 00-01-00-01-2D-19-D5-62-58-11-22-88-2D-40
DNS Servers . . . . . : 192.168.0.1
                        0.0.0.0
NetBIOS over Tcpi. . . . . : Enabled
```

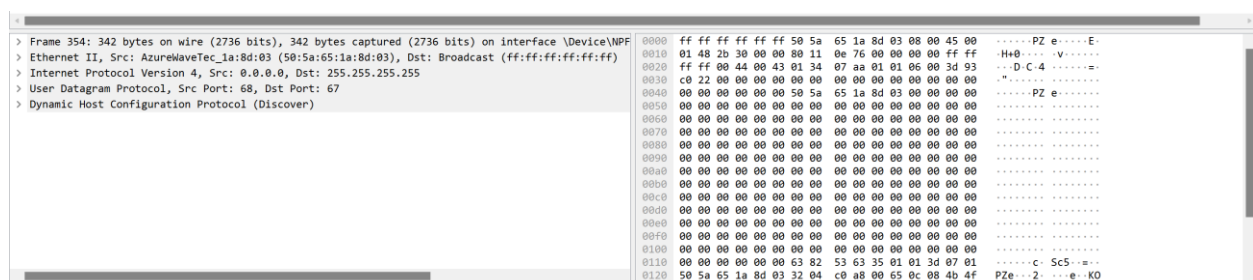
Рисунок 27 – IP-адрес компьютеру выдан DHCP-сервером



The screenshot shows a Wireshark capture of network traffic on the 'Беспроводная сеть' interface. The 'dhcp' filter is applied, and the packet list shows several DHCP messages. The packet details pane shows the structure of a DHCP Release message.

No.	Time	Source	Destination	Protocol	Length	Info
314	39.145713	192.168.0.101	192.168.0.1	DHCP	342	DHCP Release - Transaction ID 0xc3d93c022
354	44.996179	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xc3d93c022
355	44.999856	192.168.0.1	192.168.0.101	DHCP	590	DHCP Offer - Transaction ID 0xc3d93c022
356	45.000441	0.0.0.0	255.255.255.255	DHCP	356	DHCP Request - Transaction ID 0xc3d93c022
360	45.511706	192.168.0.1	192.168.0.101	DHCP	590	DHCP ACK - Transaction ID 0xc3d93c022

Рисунок 28 – Работа DHCP (было выполнено 2 команды: ipconfig /release и ipconfig /renew)



The screenshot shows the details of a DHCP Discover message (Frame 354). The left pane shows the protocol stack: Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Dynamic Host Configuration Protocol (Discover). The right pane shows the raw packet data in hexadecimal and ASCII.

```
> Frame 354: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface \Device\NPF...
> Ethernet II, Src: AzureWaveTec_1a:8d:03 (50:5a:65:1a:8d:03), Dst: Broadcast (ff: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)
```

Рисунок 29 – Строение Discover

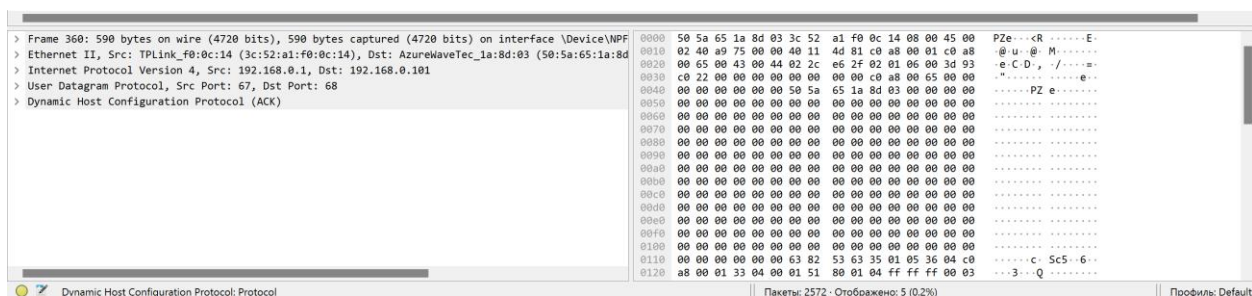


Рисунок 30 – Строение ACK

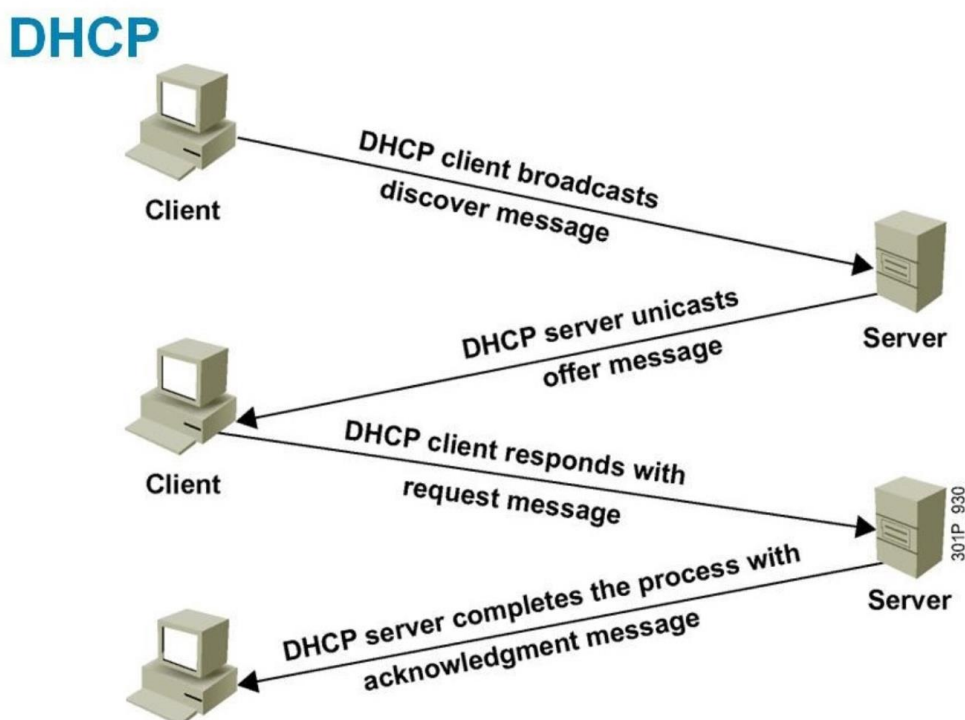


Рисунок 31 – Работа протокола DHCP

Ответы на вопросы

1. Чем различаются пакеты «DHCP Discover» и «DHCP Request»?
 - DHCP Discover — это первый пакет, отправленный клиентом в широковещательном режиме (broadcast), чтобы найти DHCP-сервер и начать процесс получения IP-адреса.
 - DHCP Request — это пакет, отправляемый клиентом после получения предложения от DHCP-сервера (DHCP Offer), чтобы официально запросить предложенный IP-адрес.
2. Как и почему менялись MAC- и IP-адреса источника и назначения в переданных DHCP-пакетах?

В начале обмена (пакет Discover) клиент ещё не знает свой IP-адрес, поэтому он использует широковещательный IP-адрес (0.0.0.0) и MAC-адрес своего сетевого интерфейса. В пакете DHCP Offer сервер сообщает клиенту, какой IP-адрес может быть назначен. MAC-адрес источника — это всегда MAC-адрес клиента, а MAC-адрес назначения зависит от того, кому отправляется пакет: серверу или всем устройствам в сети (в случае broadcast).

3. *Каков IP-адрес DHCP-сервера?*

Адрес DHCP-сервера можно найти в поле "Server Identifier" в пакете DHCP Offer или DHCP ACK. Он указывает, какой сервер отвечает на запросы.

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

Очистка фильтра в Wireshark приведет к тому, что вы снова будете видеть весь сетевой трафик, а не только DHCP-пакеты. Это может усложнить анализ, так как в захваченном трафике будет больше данных.

9 ЗАКЛЮЧЕНИЕ

В результате проделанной работы была исследована структура протокольных блоков данных, а также проведен анализ реального сетевого трафика на компьютере студента с использованием бесплатной утилиты Wireshark.

Изучены возможности программы Wireshark, выполнен сбор дампов пакетов и произведено детальное исследование необходимых сетевых пакетов.