

**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:

Чу Ван Доан, студент группы N3347



(подпись)

Проверил:

Есипов Дмитрий Андреевич

(отметка о выполнении)

(подпись)

Санкт-Петербург
2024 г.

СОДЕРЖАНИЕ

Введение.....	3
1 ХОД РАБОТЫ.....	4
1.1 Анализ трафика утилиты ping.....	4
1.2 Анализ трафика утилиты tracert.....	7
1.3 Анализ HTTP-трафика.....	10
1.4 Анализ DNS-трафика.....	12
1.5 Анализ ARP-трафика.....	14
1.6 Анализ трафика утилиты nslookup.....	16
1.7 Анализ FTP-трафика.....	18
1.8 Анализ DHCP-трафика.....	20
Заключение.....	23

Введение

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

Для достижения поставленной цели необходимо решить следующие задачи:

- установить Wireshark;
- ознакомиться с теорией;
- выполнить анализ трафика;
- составить отчёт.

1 ХОД РАБОТЫ

Чтобы выполнить эту лабораторную работу, я использую веб-сайт с доменом второго уровня "chu", что является моей фамилией: <https://chu292.github.io>

1.1 Анализ трафика утилиты ping

```
C:\Users\chudo>ping -l 1000 chu292.github.io

Pinging chu292.github.io [185.199.108.153] with 1000 bytes of data:
Reply from 185.199.108.153: bytes=1000 time=15ms TTL=255
Reply from 185.199.108.153: bytes=1000 time=22ms TTL=255
Reply from 185.199.108.153: bytes=1000 time=32ms TTL=255
Reply from 185.199.108.153: bytes=1000 time=20ms TTL=255

Ping statistics for 185.199.108.153:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 15ms, Maximum = 32ms, Average = 22ms
```

Рисунок 1 – Пример использования утилиты ping

icmp && ip.addr == 185.199.108.153							
No.	Time	Source	Destination	Protocol	Length	Info	
4	2024-11-30 1...	10.0.2.15	185.199.108.153	ICMP	1042	Echo (ping) request	id=0x0001, seq=114/29184, ttl=128 (reply in 5)
5	2024-11-30 1...	185.199.108.153	10.0.2.15	ICMP	1042	Echo (ping) reply	id=0x0001, seq=114/29184, ttl=255 (request in 4)
7	2024-11-30 1...	10.0.2.15	185.199.108.153	ICMP	1042	Echo (ping) request	id=0x0001, seq=115/29440, ttl=128 (reply in 10)
10	2024-11-30 1...	185.199.108.153	10.0.2.15	ICMP	1042	Echo (ping) reply	id=0x0001, seq=115/29440, ttl=255 (request in 7)
14	2024-11-30 1...	10.0.2.15	185.199.108.153	ICMP	1042	Echo (ping) request	id=0x0001, seq=116/29696, ttl=128 (reply in 15)
15	2024-11-30 1...	185.199.108.153	10.0.2.15	ICMP	1042	Echo (ping) reply	id=0x0001, seq=116/29696, ttl=255 (request in 14)
17	2024-11-30 1...	10.0.2.15	185.199.108.153	ICMP	1042	Echo (ping) request	id=0x0001, seq=117/29952, ttl=128 (reply in 18)
18	2024-11-30 1...	185.199.108.153	10.0.2.15	ICMP	1042	Echo (ping) reply	id=0x0001, seq=117/29952, ttl=255 (request in 17)

Рисунок 2 – Пример трафика утилиты ping

Ответы на вопросы:

1. Имеет ли место фрагментация исходного пакета, какое поле на это указывает?
 - Нет, фрагментация отсутствует.
 - Поля, которые указывают на это: "Flags" (значение 0x0) и "Fragment Offset" (значение 0).

```

✓ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 185.199.108.153
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1028
    Identification: 0x0b74 (2932)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: ICMP (1)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.0.2.15
    Destination Address: 185.199.108.153
    [Stream index: 2]

```

Рисунок 3 – Пакет со флагом 0x0

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

Установка флага MF (More fragments) в заголовке IPv4 указывает, что фрагмент пакета является промежуточным. Сброс флага MF (More fragments) в заголовке IPv4 указывает, что фрагмент пакета является последним (рисунок 4).

```

✓ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 185.199.108.153
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1028
    Identification: 0x0b74 (2932)
  ✓ 000. .... = Flags: 0x0
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0

```

Рисунок 4 - Последний пакет

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

Количество фрагментов при передаче ping-пакетов при разных размерах пакета показано в таблице 1.

Таблица 1. Зависимость количества фрагментов от размера пакета.

Размер Пакета	Количество Фрагментов	Размер Пакета	Количество Фрагментов	Размер Пакета	Количество Фрагментов
------------------	--------------------------	------------------	--------------------------	------------------	--------------------------

100	1	3600	3	7100	5
600	1	4100	3	7600	6
1100	1	4600	4	8100	6
1600	2	5100	4	8600	6
2100	2	5600	4	9100	7
2600	2	6100	5	9600	7
3100	3	6600	5	10000	7

Рисунок 5 - Количество фрагментов

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

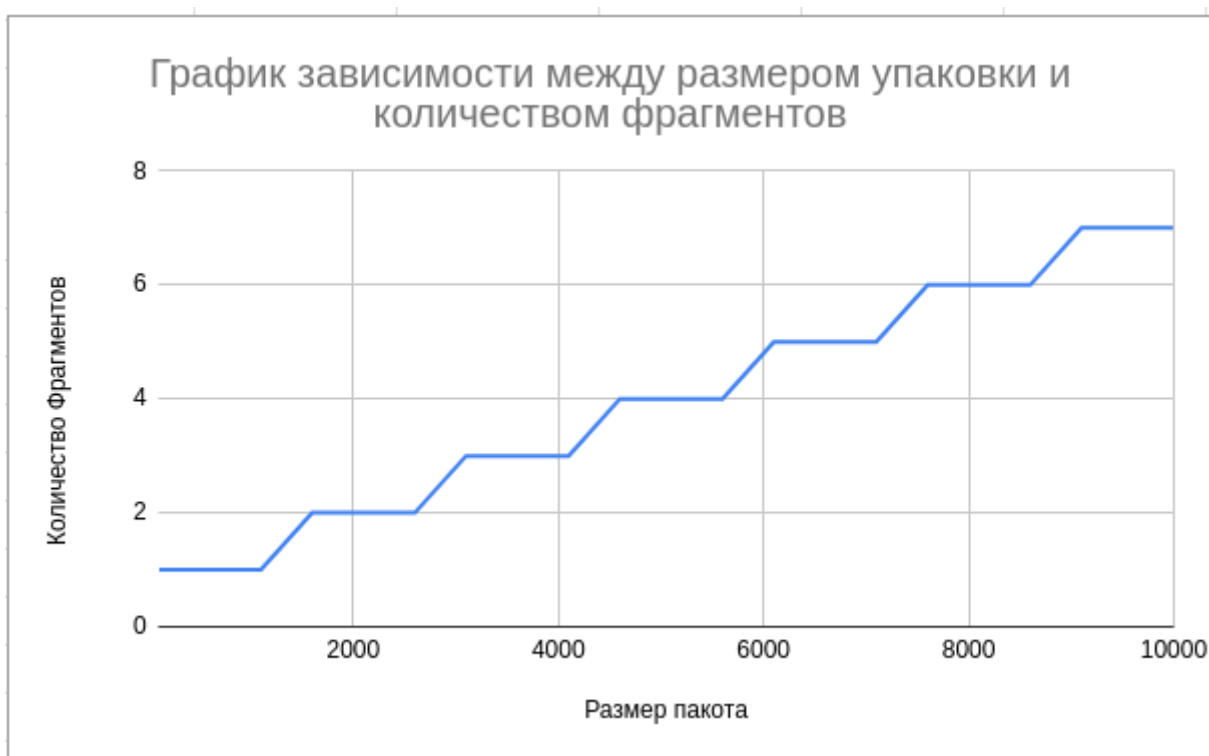


Рисунок 5 - Зависимость количества фрагментов от размера пакета

5. Как изменить поле TTL с помощью утилиты ping?
- ping -i <срок жизни в миллисекундах> <адрес>

```
C:\Users\chudo>ping -l 100 -i 64 chu292.github.io

Pinging chu292.github.io [185.199.108.153] with 100 bytes of data:
Reply from 185.199.108.153: bytes=100 time=31ms TTL=255
Reply from 185.199.108.153: bytes=100 time=33ms TTL=255
Reply from 185.199.108.153: bytes=100 time=42ms TTL=255
Reply from 185.199.108.153: bytes=100 time=28ms TTL=255

Ping statistics for 185.199.108.153:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 28ms, Maximum = 42ms, Average = 33ms
```

Рисунок 6 - Изменение TTL

6. Что содержится в поле данных ring-пакета?

Поле данных пакета ring содержит часть "payload", которая может быть настроена пользователем или заполнена по умолчанию операционной системой.

The image shows a Wireshark packet capture of an ICMP Echo (ping) request and reply. The packet list on the left shows packet 7 selected. The packet details pane shows the structure of the ICMP Echo (ping) request. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
7	Frame 767: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface \Device\NPF_{3FC6375C-0010-0080-0E-CF} 00:02:00:02:02:02 (52:55:0A:00:02:02), Dst: PC3Systemtec_B5:e6:4f (08:00:27:85:e6:4f)	185.199.108.153	10.0.2.15	Internet Control Message Protocol	100	Type: 0 (Echo (ping) reply)
Code: 0						
Checksum: 0xf24f [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 158 (0x009e)						
Sequence Number (LE): 40448 (0x9e00)						
[Request frame: 225]						
[Response time: 28.160 ms]						
Data (100 bytes)						
Data: 6162636465666768696a6b6c6d6e6f707172737475767768696a6b6c6d6e6f70717273747576776162 [Length: 100]						

The packet details pane shows the structure of the ICMP Echo (ping) request. The packet bytes pane shows the raw data in hexadecimal and ASCII.

```

0000  08 00 27 85 e6 4f 52 55 0a 00 02 02 00 45 00  ....ORU.....E
0010  00 80 0e cf 00 00 ff 01 7a 3e b9 c7 6c 99 0a 00  ....O.....>...1
0020  02 0f 00 00 f2 4f 00 01 00 9e 61 62 63 64 65 66  ....O.....hbcdef
0030  67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ....ghijklmnopqr
0040  77 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f  ....stuvwxyzabcd
0050  70 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68  ....efghijklmnop
0060  69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 61  ....qrsuvwxyzabc
0070  62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71  ....defghijklmnop
0080  72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 6a  ....rstuvwxyzabc
  
```

Рисунок 7 - Фрагмент данных ring-пакета

1.2 Анализ трафика утилиты tracert

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\chudo>tracert -d chu292.github.io

Tracing route to chu292.github.io [185.199.108.153]
over a maximum of 30 hops:

  1    15 ms    15 ms    15 ms  185.199.108.153

Trace complete.
```

Рисунок 8 - Пример использования утилиты tracert

No.	Time	Source	Destination	Protocol	Length	Info
3	2024-11-30 1...	10.0.2.15	185.199.108.153	ICMP	106	Echo (ping) request id=0x0001, seq=162/41472, ttl=1 (reply in 4)
4	2024-11-30 1...	185.199.108.153	10.0.2.15	ICMP	106	Echo (ping) reply id=0x0001, seq=162/41472, ttl=255 (request in 3)
5	2024-11-30 1...	10.0.2.15	185.199.108.153	ICMP	106	Echo (ping) request id=0x0001, seq=163/41728, ttl=1 (reply in 6)
6	2024-11-30 1...	185.199.108.153	10.0.2.15	ICMP	106	Echo (ping) reply id=0x0001, seq=163/41728, ttl=255 (request in 5)
7	2024-11-30 1...	10.0.2.15	185.199.108.153	ICMP	106	Echo (ping) request id=0x0001, seq=164/41984, ttl=1 (reply in 8)
8	2024-11-30 1...	185.199.108.153	10.0.2.15	ICMP	106	Echo (ping) reply id=0x0001, seq=164/41984, ttl=255 (request in 7)

Рисунок 9 - 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.

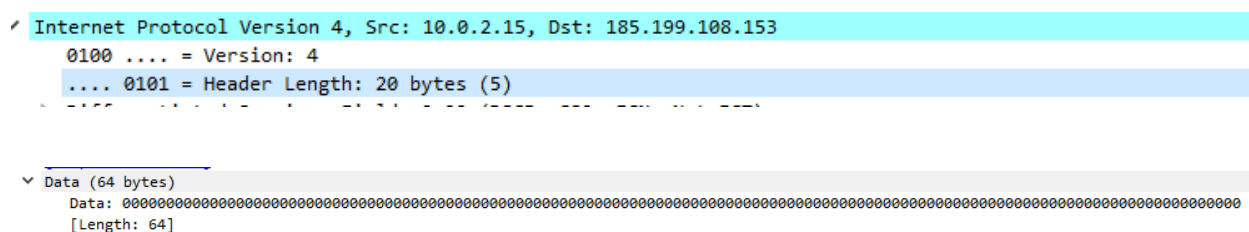


Рисунок 10 - 20 байт в заголовке, 64 байта – в поле данных

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

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 tracert 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, tracert может отследить путь, который пакет проделал через сеть, до самого конечного узла.

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

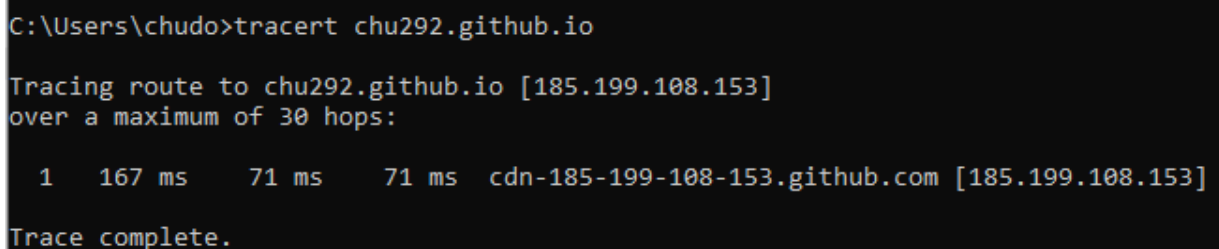
ICMP-пакеты, генерируемые утилитой tracert, отличаются от ICMP-пакетов, генерируемых утилитой ping, использованием поля TTL. Tracert модифицирует TTL, чтобы получать ICMP Time Exceeded (тип 11) от промежуточных маршрутизаторов, тогда как ping просто отправляет ICMP Echo Request (тип 8) и ожидает Echo Reply (тип 0) от конечного узла.

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

ICMP Reply нужен для подтверждения успешной доставки пакета. ICMP Error нужен для диагностики и обработки ошибок в сети. Оба типа пакетов критически важны для управления и отладки сетевых подключений.

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

Если убрать ключ -d, tracert начнет выполнять обратное разрешение DNS для IP-адресов промежуточных узлов, что замедлит выполнение команды и создаст дополнительный трафик в виде DNS-запросов.



```
C:\Users\chudo>tracert chu292.github.io

Tracing route to chu292.github.io [185.199.108.153]
over a maximum of 30 hops:

  0  167 ms  71 ms  71 ms  cdn-185-199-108-153.github.com [185.199.108.153]
    >
Trace complete.
```

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

1.3 Анализ HTTP-трафика

Отследим и проанализируем HTTP-трафик, создаваемый браузером при посещении Интернет-сайта, заданного по варианту. Список захваченных пакетов показан на рисунке 12.

No.	Time	Source	Destination	Protocol	Length	Info
1722	2024-11-30 17:34:37,588266679	192.168.0.110	91.105.192.100	HTTP	363	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
1728	2024-11-30 17:34:37,609660121	91.105.192.100	192.168.0.110	HTTP	333	HTTP/1.1 200 OK
2659	2024-11-30 17:34:56,983245294	192.168.0.110	149.154.167.51	HTTP	524	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
2662	2024-11-30 17:34:56,983916558	192.168.0.110	149.154.167.41	HTTP	504	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
2670	2024-11-30 17:34:57,026156789	149.154.167.51	192.168.0.110	HTTP	405	HTTP/1.1 200 OK
2673	2024-11-30 17:34:57,029939784	149.154.167.41	192.168.0.110	HTTP	377	HTTP/1.1 200 OK
21602	2024-11-30 17:39:05,445705191	91.189.91.98	192.168.0.110	HTTP	251	HTTP/1.1 204 No Content

Рисунок 12 - Захваченные пакеты

HTTP (протокол передачи гипертекста) — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных. GET-сообщение от клиента показано на рисунке 12. Он применяется, когда браузер запрашивает объект, идентифицирующий полем URL.

```
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
Host: connectivity-check.ubuntu.com\r\n
Accept: */*\r\n
Connection: close\r\n
\r\n
[Full request URI: http://connectivity-check.ubuntu.com/]
[HTTP request 1/1]
[Response in frame: 21602]
```

Рисунок 13 - GET-сообщение

Ответ сервера показан на рисунке 13. В ответе сервер отвечает ОК (код ответа 200) и присылает нужные данные.

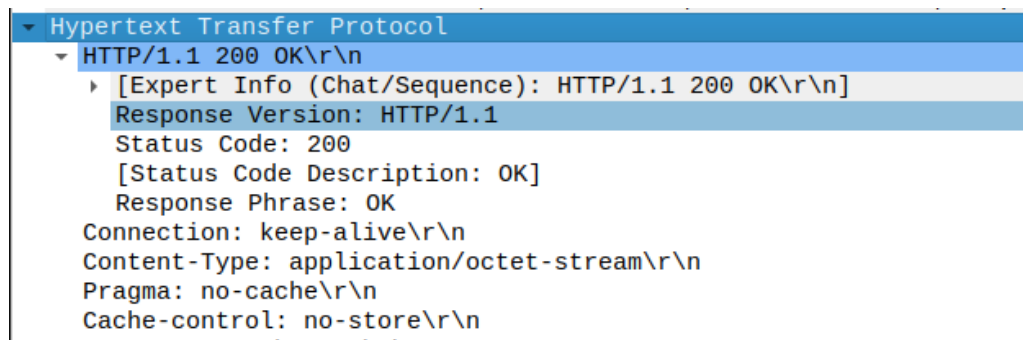


Рисунок 14 - Ответ сервера с кодом 200

HTTP-протокол имеет механизм, позволяющий прокси-серверу проверять актуальность объектов. Для этого применяется так называемый метод GET с условием. После того, что мы обновили страницу в браузере, условный GET был сгенерирован (рисунок 14).

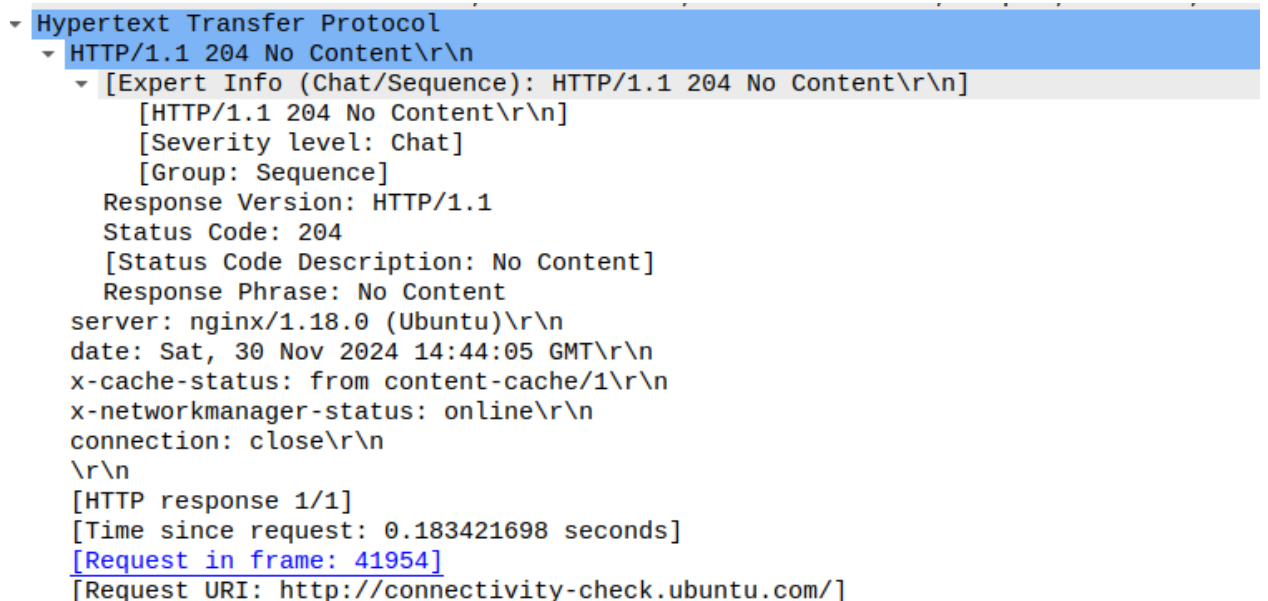


Рисунок 15 - Ответ сервера с кодом 204

Ответ с кодом 204 No Content в данном случае используется для проверки того, что сетевое подключение клиента к <http://connectivity-check.ubuntu.com/> работает нормально. Сервер не отправляет никакого содержимого, так как цель запроса заключается только в подтверждении состояния подключения, а не в получении данных (рисунок 15).

1.4 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 16 – Фрагмент DNS-трафика

No.	Time	Source	Destination	Protocol	Length	Info
92947	2024-11-30 17:52:27,74266620	192.168.0.110	192.168.0.1	DNS	87	Standard query 0x70a0 AAAA chu292.github.io OPT
92973	2024-11-30 17:52:27,89590394	192.168.0.1	192.168.0.110	DNS	151	Standard query response 0x06d5 A chu292.github.io A 185.199.108.153 A 185.199.111.153 A 185.
92989	2024-11-30 17:52:27,947919102	192.168.0.1	192.168.0.110	DNS	199	Standard query response 0x70a0 AAAA chu292.github.io AAAA 2006:50c0:8002::153 AAAA 2006:50c0:
93249	2024-11-30 17:52:30,126117035	192.168.0.110	8.8.8.8	DNS	83	Standard query 0x6c5c PTR 1.0.17.172.in-addr.arpa
93251	2024-11-30 17:52:30,157094375	8.8.8.8	192.168.0.110	DNS	83	Standard query response 0x6c5c No such name PTR 1.0.17.172.in-addr.arpa
93534	2024-11-30 17:52:32,818259610	8.8.8.8	192.168.0.110	DNS	140	Standard query response 0x744e A chu292.github.io A 185.199.108.153 A 185.199.111.153 A 185.
93535	2024-11-30 17:52:32,818259752	8.8.8.8	192.168.0.110	DNS	188	Standard query response 0x004c AAAA chu292.github.io AAAA 2006:50c0:8002::153 AAAA 2006:50c0:

Рисунок 16 - Фрагмент DNS-трафика

Frame 92947: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface wlo1, id 0	
Ethernet II, Src: IntelCor_09:87:f1 (68:3e:26:09:87:f1), Dst: TP-Link_58:3e:a0 (9c:a2:f4:58:3e:a0)	
Internet Protocol Version 4, Src: 192.168.0.110, 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: 73	
Identification: 0x0ef9 (3833)	
Flags: 0x00	
0... = Reserved bit: Not set	
.0.. = Don't fragment: Not set	
..0. = More fragments: Not set	
...0 0000 0000 0000 = Fragment Offset: 0	
Time to Live: 64	
Protocol: UDP (17)	
Header Checksum: 0xe9eb [validation disabled]	
[Header checksum status: Unverified]	
Source Address: 192.168.0.110	
Destination Address: 192.168.0.1	
User Datagram Protocol, Src Port: 37283, Dst Port: 53	
Domain Name System (query)	
0000	9c a2 f4 58 3e a0 68 3e 26 09 87 f1 08 00 45 00 ...X>:h> &.....E.
0010	00 49 0e f9 0e 40 11 e9 eb c0 a8 00 6e c0 a8 -I....@:.....n..
0020	00 01 91 a3 00 35 00 35 82 06 70 a0 01 00 00 015.5..p....
0030	00 00 00 00 00 01 06 63 68 75 32 39 32 06 67 69c hu292.gi
0040	74 68 75 62 02 69 6f 00 00 1c 00 01 00 00 29 05 thub.io.....)
0050	c0 00 00 00 00 00 00 00

Рисунок 17 – Структура DNS

Ответы на вопросы:

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

Адрес, на который отправлен DNS-запрос, не совпадает с адресом посещаемого сайта, потому что DNS-запрос отправляется к DNS-серверу, который отвечает за преобразование доменного имени в IP-адрес. DNS-сервер — это отдельный узел в сети, и его адрес обычно задается провайдером интернет-услуг (ISP) или настроен вручную. После получения IP-адреса сайт запрашивается уже напрямую.

2. Какие бывают типы DNS-запросов?

Типы DNS-запросов:

- A (Address Record) – Преобразует доменное имя в IPv4-адрес.
- AAAA (IPv6 Address Record) – Преобразует доменное имя в IPv6-адрес.
- CNAME (Canonical Name) – Используется для указания псевдонима доменного имени.
- MX (Mail Exchange) – Определяет почтовые серверы для домена.
- NS (Name Server) – Указывает DNS-серверы для домена.
- PTR (Pointer Record) – Обратное разрешение (из IP-адреса в доменное имя).
- TXT (Text Record) – Содержит текстовую информацию, часто используется для проверки доменов.
- SOA (Start of Authority) – Информация о зоне DNS.

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

Независимые DNS-запросы для получения изображений на сайте нужно выполнять в следующей ситуации:

Когда изображения на сайте загружаются с других доменов (например, через внешние ресурсы или CDN). В таких случаях для каждого домена, с которого загружаются изображения, выполняется отдельный DNS-запрос, чтобы преобразовать доменное имя в IP-адрес.

Пример:

Сайт может находиться на `example.com`, но изображения могут загружаться с `cdn.example.com` или другого внешнего сервера. В таком случае нужно сначала выполнить

DNS-запрос для домена `cdn.example.com`, чтобы получить его IP-адрес и загрузить изображения.

1.5 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
44181	2024-11-30 17:44:38,835658110	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	Who has 192.168.0.110? Tell 192.168.0.1
44182	2024-11-30 17:44:38,835673463	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	192.168.0.110 is at 68:3e:26:09:87:f1
47423	2024-11-30 17:45:22,488070722	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	Who has 192.168.0.110? Tell 192.168.0.1
47424	2024-11-30 17:45:22,488085464	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	192.168.0.110 is at 68:3e:26:09:87:f1
51244	2024-11-30 17:46:06,190570986	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	Who has 192.168.0.110? Tell 192.168.0.1
51245	2024-11-30 17:46:06,190600716	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	192.168.0.110 is at 68:3e:26:09:87:f1
56755	2024-11-30 17:46:49,837174293	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	Who has 192.168.0.110? Tell 192.168.0.1
56756	2024-11-30 17:46:49,837183029	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	192.168.0.110 is at 68:3e:26:09:87:f1
60737	2024-11-30 17:47:36,939837762	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	Who has 192.168.0.110? Tell 192.168.0.1
60738	2024-11-30 17:47:36,939848373	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	192.168.0.110 is at 68:3e:26:09:87:f1
62797	2024-11-30 17:47:53,582041608	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	Who has 192.168.0.1? Tell 192.168.0.110
62798	2024-11-30 17:47:53,584295196	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	192.168.0.1 is at 9c:a2:f4:58:3e:a0
65429	2024-11-30 17:48:22,625322792	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	Who has 192.168.0.110? Tell 192.168.0.1
65430	2024-11-30 17:48:22,625360132	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	192.168.0.110 is at 68:3e:26:09:87:f1
66704	2024-11-30 17:48:37,614001409	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	Who has 192.168.0.1? Tell 192.168.0.110
66705	2024-11-30 17:48:37,619335365	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	192.168.0.1 is at 9c:a2:f4:58:3e:a0
69723	2024-11-30 17:49:09,993459673	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	Who has 192.168.0.110? Tell 192.168.0.1
69724	2024-11-30 17:49:09,993471719	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	192.168.0.110 is at 68:3e:26:09:87:f1
75858	2024-11-30 17:49:54,926991256	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	Who has 192.168.0.1? Tell 192.168.0.110
75859	2024-11-30 17:49:54,930296980	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	192.168.0.1 is at 9c:a2:f4:58:3e:a0
82367	2024-11-30 17:50:44,589994772	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	Who has 192.168.0.1? Tell 192.168.0.110
82368	2024-11-30 17:50:44,593380289	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	192.168.0.1 is at 9c:a2:f4:58:3e:a0
88112	2024-11-30 17:51:33,743020310	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	Who has 192.168.0.1? Tell 192.168.0.110
88114	2024-11-30 17:51:33,811535294	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	192.168.0.1 is at 9c:a2:f4:58:3e:a0
90578	2024-11-30 17:51:57,942027075	TP-Link_58:3e:a0	IntelCor_09:87:f1	ARP	42	Who has 192.168.0.110? Tell 192.168.0.1
90579	2024-11-30 17:51:57,942043221	IntelCor_09:87:f1	TP-Link_58:3e:a0	ARP	42	192.168.0.110 is at 68:3e:26:09:87:f1

Frame 90579: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlo1, id 0

Ethernet II, Src: IntelCor_09:87:f1 (68:3e:26:09:87:f1), Dst: TP-Link_58:3e:a0 (9c:a2:f4:58:3e:a0)

Address Resolution Protocol (reply)

Рисунок 17 – Фрагмент ARP-трафика

Frame 124317: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlo1, id 0
Ethernet II, Src: IntelCor_09:87:f1 (68:3e:26:09:87:f1), Dst: TP-Link_58:3e:a0 (9c:a2:f4:58:3e:a0)
Destination: TP-Link_58:3e:a0 (9c:a2:f4:58:3e:a0)
Source: IntelCor_09:87:f1 (68:3e:26:09:87:f1)
Type: ARP (0x0806)
Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: IntelCor_09:87:f1 (68:3e:26:09:87:f1)
Sender IP address: 192.168.0.110
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.0.1

Рисунок 18 – Пример ARP-запроса

```
Frame 125261: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlo1, id 0
Ethernet II, Src: TP-Link_58:3e:a0 (9c:a2:f4:58:3e:a0), Dst: IntelCor_09:87:f1 (68:3e:26:09:87:f1)
  Destination: IntelCor_09:87:f1 (68:3e:26:09:87:f1)
  Source: TP-Link_58:3e:a0 (9c:a2:f4:58:3e:a0)
  Type: ARP (0x0806)
Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: TP-Link_58:3e:a0 (9c:a2:f4:58:3e:a0)
  Sender IP address: 192.168.0.1
  Target MAC address: IntelCor_09:87:f1 (68:3e:26:09:87:f1)
  Target IP address: 192.168.0.110
```

Рисунок 19 – 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).

MAC-адрес отправителя (Source MAC Address):

- Что означает: MAC-адрес устройства, которое отправляет ARP-запрос или ARP-ответ.
- Что идентифицирует: Идентифицирует устройство, инициировавшее запрос или отправившее ответ.

MAC-адрес получателя (Destination MAC Address):

- Что означает: В ARP-запросе это широковещательный адрес FF:FF:FF:FF:FF:FF. В ARP-ответе это MAC-адрес устройства, которое ответило на запрос.
- Что идентифицирует: В запросе — все устройства в сети, в ответе — конкретное устройство, отвечающее на запрос.

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.

MAC-адрес отправителя (Source MAC Address):

- Что означает: MAC-адрес устройства, которое отправляет HTTP-запрос или ответ.

- Что идентифицирует: Идентифицирует устройство, которое отправляет пакет, например, клиент (компьютер или телефон), который отправляет запрос на сервер.

MAC-адрес получателя (Destination MAC Address):

- Что означает: MAC-адрес устройства, которому адресован HTTP-пакет.
- Что идентифицирует: Идентифицирует устройство, принимающее пакет, например, маршрутизатор или сервер, принимающий запрос.

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

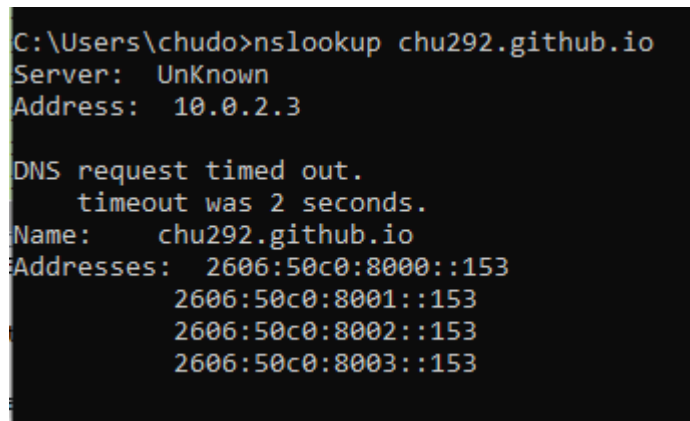
ARP-запрос содержит IP-адрес источника для того, чтобы устройство, получающее запрос, знало, от какого IP-адреса пришел запрос и могло отправить ARP-ответ обратно на правильный IP-адрес.

1.6 Анализ трафика утилиты nslookup

nslookup — утилита, предоставляющая пользователю интерфейс командной строки для обращения к системе DNS (проще говоря, DNS-клиент).

Отследим и проанализируем трафик протокола DNS, сгенерированный в результате выполнения следующих действий:

- настроить Wireshark-фильтр: “ip.addr == 10.0.2.3”.
- запустить команду “nslookup chu292.github.io” (рис.20);
- дождаться отправки трёх DNS-запросов и трёх DNS-ответов;
- повторить предыдущие два шага, используя команду: “nslookup -type=NS имя_сайта_по_варианту”.



```

C:\Users\chudo>nslookup chu292.github.io
Server: UnKnown
Address: 10.0.2.3

DNS request timed out.
    timeout was 2 seconds.
Name:   chu292.github.io
Addresses: 2606:50c0:8000::153
           2606:50c0:8001::153
           2606:50c0:8002::153
           2606:50c0:8003::153

```

Рисунок 20 – Выполнение команды nslookup без опции -type=NS

No.	Time	Source	Destination	Protocol	Length	Info
17778	2024-11-30 16:37:04.587518	10.0.2.15	10.0.2.3	DNS	89	Standard query 0x65f2 A v10.events.data.microsoft.com
17779	2024-11-30 16:37:04.587844	10.0.2.3	10.0.2.3	DNS	89	Standard query 0xb144 AAAA v10.events.data.microsoft.com
17780	2024-11-30 16:37:04.650799	10.0.2.3	10.0.2.15	DNS	229	Standard query response 0x65f2 A v10.events.data.microsoft.com CNAME win-global-asimov-leafs-events-data.trafficmanager.net CNAME onedscolorpdcus04.centralus.cloudap
17781	2024-11-30 16:37:04.652405	10.0.2.3	10.0.2.15	DNS	275	Standard query response 0xb144 AAAA v10.events.data.microsoft.com CNAME win-global-asimov-leafs-events-data.trafficmanager.net CNAME onedscolorpdcus04.westeurope.clo
17797	2024-11-30 16:37:05.485226	10.0.2.15	10.0.2.3	DNS	81	Standard query 0x0001 PTR 3.2.0.10.in-addr.arpa
17807	2024-11-30 16:37:05.500957	10.0.2.3	10.0.2.15	DNS	81	Standard query response 0x0001 No such name PTR 3.2.0.10.in-addr.arpa
17808	2024-11-30 16:37:05.508069	10.0.2.15	10.0.2.3	DNS	76	Standard query 0x0002 A chu292.github.io
17826	2024-11-30 16:37:07.580764	10.0.2.15	10.0.2.3	DNS	76	Standard query 0x0003 AAAA chu292.github.io
17829	2024-11-30 16:37:08.238925	10.0.2.3	10.0.2.15	DNS	188	Standard query response 0x0003 AAAA chu292.github.io AAAA 2606:58c0:8000::153 AAAA 2606:58c0:8000::153 AAAA 2606:58c0:8000::153 AAAA 2606:58c0:8000::153
18038	2024-11-30 16:37:19.115904	10.0.2.15	10.0.2.3	DNS	78	Standard query 0x2fce AAAA edge.microsoft.com
18039	2024-11-30 16:37:19.116126	10.0.2.15	10.0.2.3	DNS	78	Standard query 0x047 A edge.microsoft.com
18040	2024-11-30 16:37:19.116250	10.0.2.15	10.0.2.3	DNS	78	Standard query 0x0c8f HTTPS edge.microsoft.com
18041	2024-11-30 16:37:19.181219	10.0.2.3	10.0.2.15	DNS	182	Standard query response 0x0c8f HTTPS edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net SOA ns1.a-msedge.net
18042	2024-11-30 16:37:19.188345	10.0.2.3	10.0.2.15	DNS	181	Standard query response 0x0e47 A edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net A 13.107.21.239 A 204.79.197.23
18043	2024-11-30 16:37:19.188345	10.0.2.3	10.0.2.15	DNS	205	Standard query response 0x2fce AAAA edge.microsoft.com CNAME edge-microsoft-com.dual-a-0036.a-msedge.net CNAME dual-a-0036.a-msedge.net AAAA 2620:1ec12::239 AAAA 2
18087	2024-11-30 16:37:19.953675	10.0.2.15	10.0.2.3	DNS	101	Standard query 0x0ad1 A msedge.b.tlu.dl.delivery.mp.microsoft.com CNAME star.b.tlu.dl.delivery.mp.microsoft.com CNAME cdp-f.tlu-net.
18091	2024-11-30 16:37:20.010439	10.0.2.3	10.0.2.15	DNS	390	Standard query response 0x0ad1 A msedge.b.tlu.dl.delivery.mp.microsoft.com CNAME star.b.tlu.dl.delivery.mp.microsoft.com CNAME cdp-f.tlu-net.
18185	2024-11-30 16:37:22.078522	10.0.2.15	10.0.2.3	DNS	72	Standard query 0xb006 AAAA www.bing.com
18186	2024-11-30 16:37:22.078710	10.0.2.15	10.0.2.3	DNS	72	Standard query 0x918c A www.bing.com
18187	2024-11-30 16:37:22.078889	10.0.2.15	10.0.2.3	DNS	72	Standard query 0x288e HTTPS www.bing.com
18190	2024-11-30 16:37:22.159989	10.0.2.3	10.0.2.15	DNS	333	Standard query response 0xb006 AAAA www.bing.com CNAME www-www.bing.com CNAME www.bing.com CNAME www.bing.com CNAME e86303.dscc.akamaiedge.net AAAA 2a82 v

Рисунок 21 - Результат анализа трафика утилиты nslookup без опции -type=NS

```

C:\Users\chudo>nslookup -type=NS chu292.github.io
Server:      UnKnown
Address:     10.0.2.3

github.io
primary name server = ns-1622.awsdns-10.co.uk
responsible mail addr = awsdns-hostmaster.amazon.com
serial = 1
refresh = 7200 (2 hours)
retry = 900 (15 mins)
expire = 1209600 (14 days)
default TTL = 86400 (1 day)

```

Рисунок 22 - Выполнение команды nslookup с опцией -type=NS

	Source	Destination	Protocol	Length	Info
1-30	16:40:50.974172	10.0.2.15	DNS	78	Standard query 0x02b8 A edge.microsoft.com
1-30	16:40:50.974313	10.0.2.15	DNS	78	Standard query 0xc7e2 HTTPS edge.microsoft.com
1-30	16:40:51.041350	10.0.2.3	DNS	182	Standard query response 0xc7e2 HTTPS edge.microsoft.com
1-30	16:40:51.041350	10.0.2.3	DNS	181	Standard query response 0x02b8 A edge.microsoft.com CNAME
1-30	16:40:51.041350	10.0.2.3	DNS	205	Standard query response 0x1aaf AAAA edge.microsoft.com C
1-30	16:42:22.236722	10.0.2.15	DNS	81	Standard query 0x0001 PTR 3.2.0.10.in-addr.arpa
1-30	16:42:22.315051	10.0.2.3	DNS	81	Standard query response 0x0001 No such name PTR 3.2.0.10
1-30	16:42:22.316695	10.0.2.15	DNS	76	Standard query 0x0002 NS chu292.github.io
1-30	16:42:22.610140	10.0.2.3	DNS	163	Standard query response 0x0002 NS chu292.github.io SOA r
1-30	16:42:36.817531	10.0.2.15	DNS	78	Standard query 0xd968 AAAA edge.microsoft.com
1-30	16:42:36.817755	10.0.2.15	DNS	78	Standard query 0xa83d A edge.microsoft.com

Рисунок 24 - Результат анализа трафика утилиты nslookup с опцией -type=NS

Ответы на вопросы

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

В пунктах 2 и 4 трасса трафика различается по следующему:

В пункте 2, при использовании обычного запроса DNS, происходит разрешение доменного имени в IP-адрес.

В пункте 4, при использовании `-type=NS`, запрос выполняется для поиска серверов имен (NS-записи), и результатом является IP-адреса DNS-серверов, которые управляют доменом.

Таким образом, в пункте 2 трафик направлен к основному серверу, который предоставляет IP-адрес, а в пункте 4 — к DNS-серверам для получения информации о зоне домена.

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

В поле «Answers» DNS-ответа содержатся записи, соответствующие запросу. Это могут быть:

- A (IP-адрес для доменного имени),
- AAAA (IPv6-адрес),
- MX (почтовые серверы),
- NS (серверы имен) и другие типы записей, в зависимости от типа запроса.

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

Имена серверов, возвращающих авторитативный отклик, указаны в NS-записях DNS-ответа. Эти серверы являются авторитативными DNS-серверами для домена и отвечают на запросы с точной информацией о домене.

1.7 Анализ FTP-трафика

Отследим и проанализируем трафик протокола FTP, сгенерированный в результате выполнения следующих действий:

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

Подключение к FTP-серверу

- ftp ftp.dlptest.com
- Name: dlpuser
- Password: rNrKYTX9g7z3RgJRmxWuGHbeu

```
chu@chu-Latitude-5510:~$ ftp ftp.dlptest.com
Connected to ftp.dlptest.com.
220 Welcome to the DLP Test FTP Server
Name (ftp.dlptest.com:chu): dlpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||1046|).
150 Here comes the directory listing.
-rw-r--r-- 1 1001 1001 131138 Nov 30 16:10 10.101.1.2_20241130-11102843_IVA.jpg
-rw-r--r-- 1 1001 1001 130176 Nov 30 16:10 10.101.1.2_20241130-11103124_IVA.jpg
-rw-r--r-- 1 1001 1001 129197 Nov 30 16:10 10.101.1.2_20241130-11103624_IVA.jpg
-rw-r--r-- 1 1001 1001 132042 Nov 30 16:10 10.101.1.2_20241130-11104004_IVA.jpg
-rw-r--r-- 1 1001 1001 120845 Nov 30 16:10 10.101.1.2_20241130-11104303_IVA.jpg
-rw-r--r-- 1 1001 1001 131875 Nov 30 16:10 10.101.1.2_20241130-11104624_IVA.jpg
-rw-r--r-- 1 1001 1001 130808 Nov 30 16:10 10.101.1.2_20241130-11105084_IVA.jpg
-rw-r--r-- 1 1001 1001 131555 Nov 30 16:10 10.101.1.2_20241130-11105404_IVA.jpg
-rw-r--r-- 1 1001 1001 132662 Nov 30 16:10 10.101.1.2_20241130-11105644_IVA.jpg
-rw-r--r-- 1 1001 1001 1202436 Nov 30 16:10 1_9316790593169040232_17-9ULspeedtest.upt
-rw-r--r-- 1 1001 1001 100000 Nov 30 16:10 _11475028
drwxr-xr-x 2 1001 1001 58 Nov 30 16:10 ftp
226 Directory send OK.
ftp>
```

Рисунок 25 - Подключение к FTP-серверу

No.	Time	Source	Destination	Protocol	Length	Info
1895	2024-11-30 19:10:13,410284257	44.241.66.173	192.168.0.110	FTP	106	Response: 220 Welcome to the DLP Test FTP Server
2090	2024-11-30 19:10:17,384909316	192.168.0.110	44.241.66.173	FTP	80	Request: USER dlpuser
2118	2024-11-30 19:10:17,958995434	44.241.66.173	192.168.0.110	FTP	100	Response: 331 Please specify the password.
2228	2024-11-30 19:10:22,841286530	192.168.0.110	44.241.66.173	FTP	98	Request: PASS rNrKYTX9g7z3RgJrmxwUgHbeu
2266	2024-11-30 19:10:23,203364299	44.241.66.173	192.168.0.110	FTP	89	Response: 230 Login successful.
2269	2024-11-30 19:10:23,283651892	192.168.0.110	44.241.66.173	FTP	72	Request: SYST
2274	2024-11-30 19:10:23,463547075	44.241.66.173	192.168.0.110	FTP	85	Response: 215 UNIX Type: L8
2275	2024-11-30 19:10:23,463859061	192.168.0.110	44.241.66.173	FTP	72	Request: FEAT
2280	2024-11-30 19:10:23,640469173	44.241.66.173	192.168.0.110	FTP	81	Response: 211-Features:
2281	2024-11-30 19:10:23,640554119	44.241.66.173	192.168.0.110	FTP	77	Response: AUTH TLS
2283	2024-11-30 19:10:23,641621053	44.241.66.173	192.168.0.110	FTP	80	Response: EPRT
2284	2024-11-30 19:10:23,643037361	44.241.66.173	192.168.0.110	FTP	138	Response: MDTM
2427	2024-11-30 19:10:35,164376199	192.168.0.110	44.241.66.173	FTP	72	Request: EPSV
2428	2024-11-30 19:10:35,521410957	44.241.66.173	192.168.0.110	FTP	114	Response: 229 Entering Extended Passive Mode (1046).
2433	2024-11-30 19:10:35,825238156	192.168.0.110	44.241.66.173	FTP	72	Request: LIST
2441	2024-11-30 19:10:36,051394013	44.241.66.173	192.168.0.110	FTP	105	Response: 150 Here comes the directory listing.
2448	2024-11-30 19:10:36,342455107	44.241.66.173	192.168.0.110	FTP	90	Response: 226 Directory send OK.
2442	2024-11-30 19:10:36,051396437	44.241.66.173	192.168.0.110	FTP-DATA	1139	FTP Data: 1073 bytes (EPASV) (LIST)

Рисунок 26 - Анализ трассы FTP

Ответы на вопросы:

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

В пакете FTP-DATA содержится 1073 байт.

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

Порт транспортного уровня, который используется для передачи FTP-пакетов является 21.

3. Чем отличаются пакеты FTP от FTP-DATA?

Пакеты FTP используются для передачи управляющей информации (например, команды, ответы сервера), в то время как пакеты FTP-DATA предназначены для передачи файловых данных. Главные различия:

- FTP: управляет сессией, передает команды и ответы.

- FTP-DATA: передает сам файл или его части в рамках сессии FTP.

FTP использует управляющий канал (обычно порт 21), а FTP-DATA — канал данных (порт 20 для активного режима или случайный порт в пассивном режиме).

1.8 Анализ DHCP-трафика

DHCP (протокол динамической настройки узла) — прикладной протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер».

No.	Time	Source	Destination	Protocol	Length	Info
2436	2024-11-30 16:33:46.554534	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 0x321d4850
2437	2024-11-30 16:33:46.554931	10.0.2.2	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0x321d4850
25643	2024-11-30 16:52:44.139138	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 0xe767ea08
25644	2024-11-30 16:52:44.139286	10.0.2.2	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0xe767ea08

Рисунок 27 - Анализ DHCP трассы

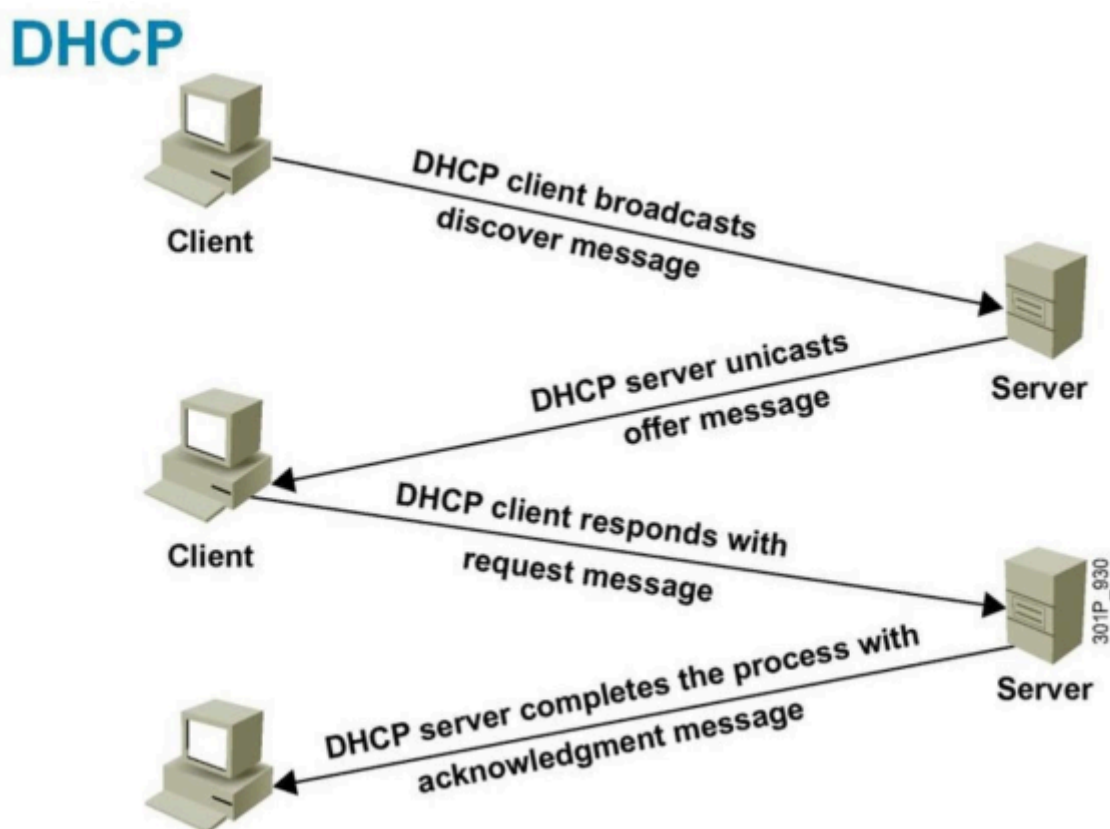


Рисунок 28 - Работа протокола DHCP

Ответы на вопросы

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

Пакеты DHCP Discover и DHCP Request имеют следующие различия:

DHCP Discover:

- Используется клиентом для поиска DHCP-сервера.
- Отправляется с широковещательным адресом (255.255.255.255).
- Содержит запрос на получение IP-адреса и другую информацию.

DHCP Request:

- Отправляется клиентом после того, как он получил предложение от DHCP-сервера.
- Содержит запрос на подтверждение выбранного IP-адреса.
- Может быть использован для продления аренды IP-адреса, если клиент уже имеет активную аренду.

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

MAC-адреса:

- В DHCP Discover и DHCP Request исходный MAC-адрес принадлежит клиенту, а MAC-адрес назначения — это широковещательный адрес (255.255.255.255), поскольку клиент еще не знает адреса сервера.
- В ответах сервера, например в DHCP Offer и DHCP Ack, исходный MAC-адрес будет указывать на сервер DHCP, а MAC-адрес назначения — это MAC-адрес клиента.
- MAC-адреса меняются, потому что на этапе обнаружения сервер и клиент взаимодействуют по каналу Ethernet, и адреса используются для точной передачи пакетов между ними.

IP-адреса:

- В DHCP Discover IP-адрес источника клиента обычно равен 0.0.0.0, потому что клиент еще не имеет IP-адреса и пытается получить его через DHCP.
- В DHCP Offer и DHCP Ack сервер присваивает клиенту IP-адрес, который он предлагает или подтверждает.
- Адреса источника и назначения изменяются, потому что сервер на момент отправки DHCP Offer или DHCP Ack знает, какой IP-адрес можно присвоить клиенту.

3. Каков IP-адрес DHCP-сервера?

До назначения IP-адреса клиенту:

В пакете DHCP Discover IP-адрес источника будет 0.0.0.0, так как клиент еще не имеет IP-адреса. IP-адрес назначения будет 255.255.255.255, это широковещательный адрес, на который отправляется запрос.

Ответ сервера:

Когда сервер DHCP отправляет DHCP Offer или DHCP Ack, его IP-адрес будет отображен в поле `siaddr` (серверный IP-адрес) в пакете. Это IP-адрес самого DHCP-сервера, который клиент может использовать для получения конфигурации.

`siaddr`: Это поле в пакете DHCP может содержать IP-адрес сервера, который клиенту необходимо использовать для получения конфигурации или для дальнейшей работы. Этот адрес может быть статический (например, заданный в настройках сети) или выделенный сервером в зависимости от конфигурации сети.

4. Что произойдёт, если очистить использованный фильтр “bootp”?

- Wireshark перестанет фильтровать пакеты DHCP: Будут отображаться все пакеты, а не только те, которые связаны с DHCP (BootP).
- Пакеты других протоколов будут видны: Например, пакеты ARP, TCP, UDP и другие, которые были до этого скрыты фильтром "bootp".

Заключение

Выполнены задачи:

- выполнены наблюдения за передаваемым трафиком с компьютера пользователя в Интернет и в обратном направлении;
- анализированы последовательности команд и назначение служебных данных, используемых для организации обмена данными в следующих протоколах: ARP, DNS, FTP, HTTP, DHCP.

После выполнения работы узнала, как читать захваченный пакет, как использовать фильтр для того, чтобы получить только пакеты, которые нам нужны, а также понятнее природу протоколов.