

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

Факультет безопасности информационных технологий

Дисциплина:

«Вычислительные сети и контроль безопасности в компьютерных сетях»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

«Сканирование и анализ сетевого трафика»

Выполнили:

Чу Ван Доан, студент группы N3347



(подпись)

Чан Бао Линь, студент группы N3346



(подпись)

Проверил:

Савков Сергей Витальевич, инженер факультета БИТ

(отметка о выполнении)

(подпись)

Санкт-Петербург

2025 г.

СОДЕРЖАНИЕ

Содержание.....	2
Введение.....	3
1. Задание.....	4
2. Ход работы.....	4
2.1. Установка инструментов.....	4
2.1.1. tcpdump.....	4
2.1.2. Wireshark.....	4
2.2. Сканирование трафика с помощью tcpdump.....	5
2.3. Сканирование трафика с помощью Wireshark.....	8
2.4. Проверьте IP-адреса источника и назначения для поиска.....	11
2.5. Анализ полученных результатов с помощью tcpdump.....	12
2.6. Анализ полученных результатов с помощью Wireshark.....	12
2.7. Сравнение Wireshark и tcpdump.....	13
Заключение.....	14

ВВЕДЕНИЕ

Цель работы – Изучить инструменты сканирования сетевого трафика, такие как tcpdump и Wireshark.

Для достижения поставленной цели необходимо решить следующие задачи:

- ознакомиться с назначением и возможностями следующих инструментов: tcpdump, Wireshark;
- установить на физическую или виртуальную машину указанные инструменты;
- выполнить сканирование трафика в соответствии с вариантом и сохранить дампы трафика; изучить использование фильтров Wireshark;
- результаты выполнения работы оформить в виде отчета.
- З/ПК с ОС Kali Linux или другой версией Debian-based или другого дистрибутива Linux, допускается использование виртуальной машины.

1. Задание

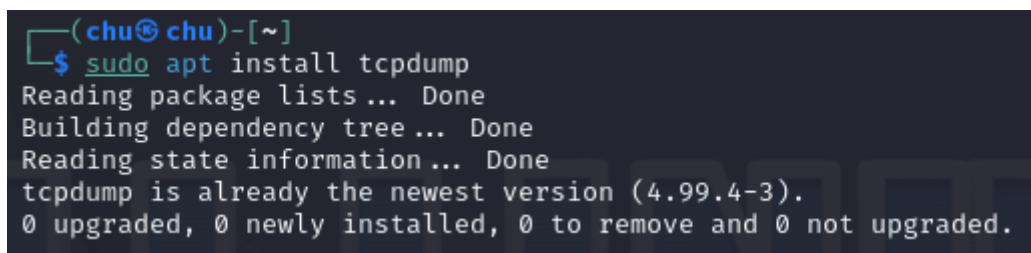
- Вариант 4: звонок в Telegram
- выполнить сканирование трафика с помощью tcpdump;
- дополнительное: выполнить сканирование трафика при помощи Wireshark и сравнить полученные дампы трафика.

2. Ход работы

2.1. Установка инструментов

2.1.1. tcpdump

- tcpdump — это мощный инструмент командной строки в Linux, который используется для перехвата (захвата) сетевых пакетов. Он помогает системным администраторам и инженерам по сетям анализировать, отслеживать и устранять проблемы в сетевом трафике.
- Применение tcpdump:
 - Мониторинг сети — отслеживание активности и анализа трафика.
 - Безопасность — выявление подозрительных или вредоносных данных.
 - Диагностика сетевых проблем — анализ соединений между клиентом и сервером.
 - Анализ пакетов в Wireshark — сохранение и глубокий анализ сетевых пакетов.



```
(chu@chu)-[~]  
$ sudo apt install tcpdump  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
tcpdump is already the newest version (4.99.4-3).  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

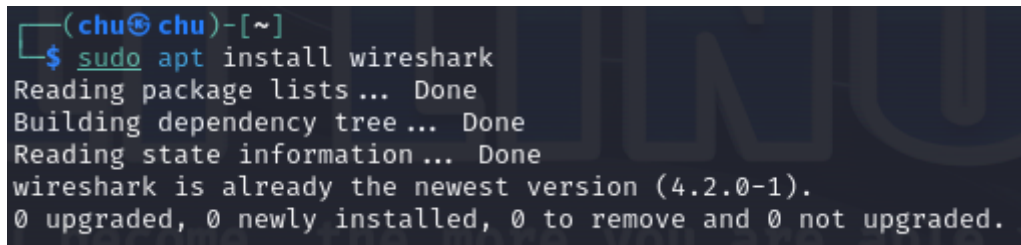
Рисунок 1 – tcpdump

2.1.2. Wireshark

- Wireshark — это мощный анализатор сетевых протоколов (network protocol analyzer), который позволяет перехватывать, исследовать и анализировать пакеты в сети в режиме реального времени. Это популярный инструмент среди сетевых инженеров, специалистов по кибербезопасности и системных

администраторов для диагностики сети, мониторинга трафика и обнаружения проблем с безопасностью.

- Применение Wireshark:
 - Мониторинг сетевого трафика — просмотр и анализ передаваемых данных.
 - Диагностика сетевых проблем — выявление ошибок соединения между клиентом и сервером.
 - Обнаружение атак — анализ DDoS, Man-in-the-Middle (MITM) и других вредоносных действий.
 - Проверка безопасности данных — анализ шифрования и утечек информации.
 - Измерение производительности сети — выявление задержек и узких мест в передаче данных.

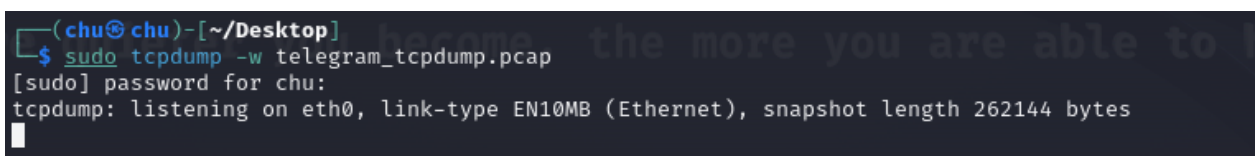


```
(chu@chu)-[~]  
$ sudo apt install wireshark  
Reading package lists ... Done  
Building dependency tree ... Done  
Reading state information ... Done  
wireshark is already the newest version (4.2.0-1).  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Рисунок 2 – wireshark

2.2. Сканирование трафика с помощью tcpdump.

- Запуск программы tcpdump для сканирования трафика показаны на рисунке 3. Мы запишем результаты в файл telegram_tcpdump.pcap.



```
(chu@chu)-[~/Desktop]  
$ sudo tcpdump -w telegram_tcpdump.pcap  
[sudo] password for chu:  
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes  
█
```

Рисунок 3 - Запуска tcpdump

- Затем мы открываем браузер Firefox и ищем сайт Telegram.

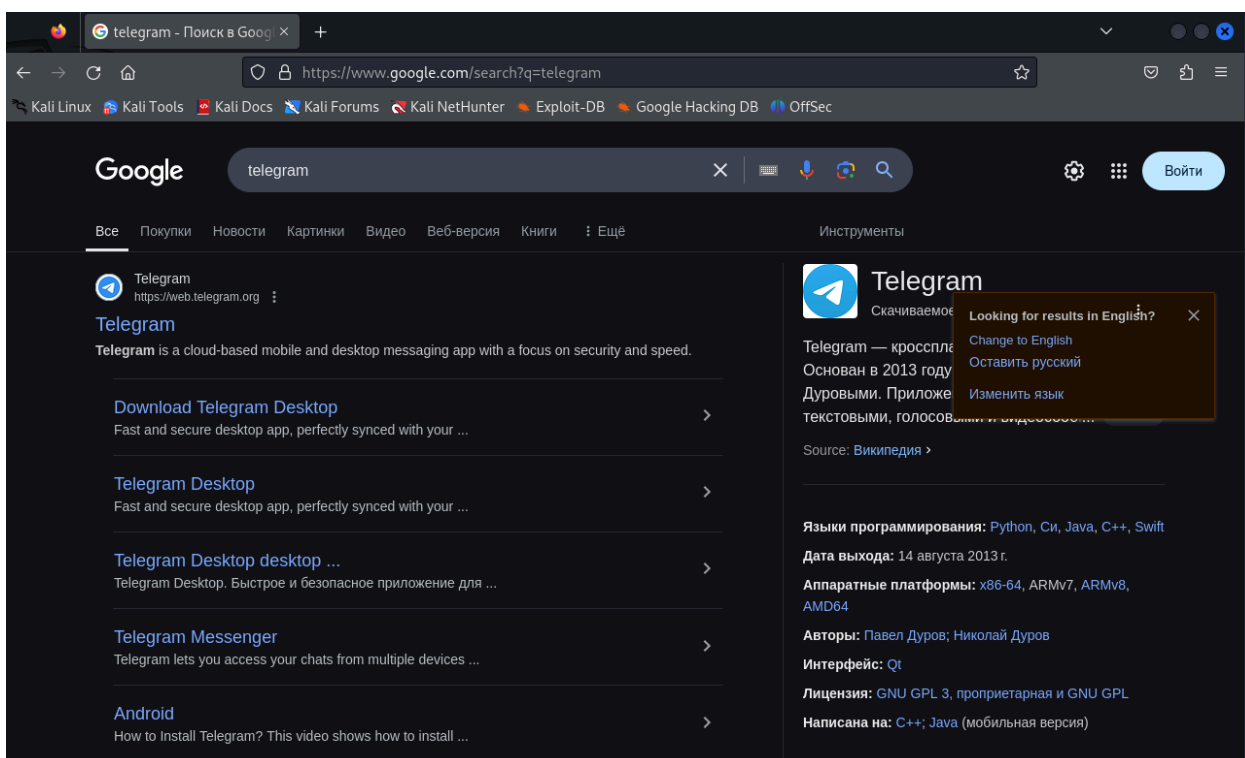


Рисунок 4 - Поиск сайта Telegram

- Затем мы входим в Telegram, отсканировав QR-код.

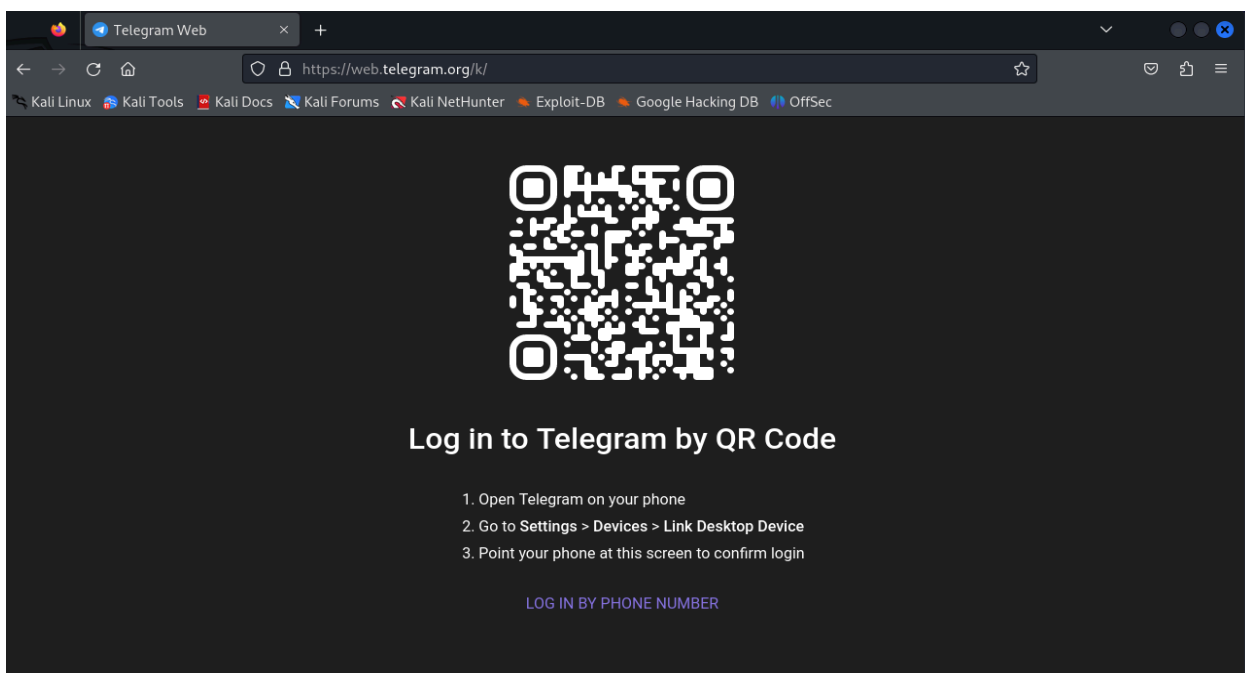


Рисунок 5 - Вход в Telegram

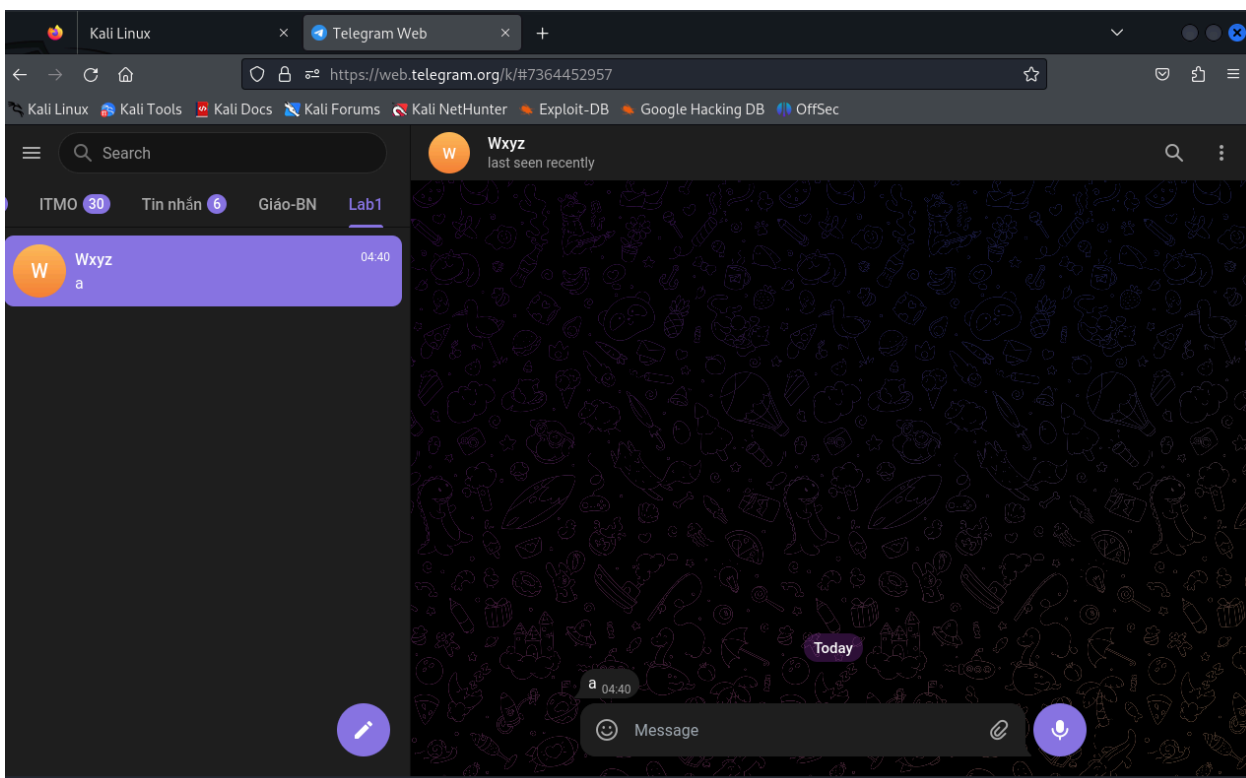


Рисунок 6 - Интерфейс Telegram после входа

- Отправка и получение сообщений для одного пользователя

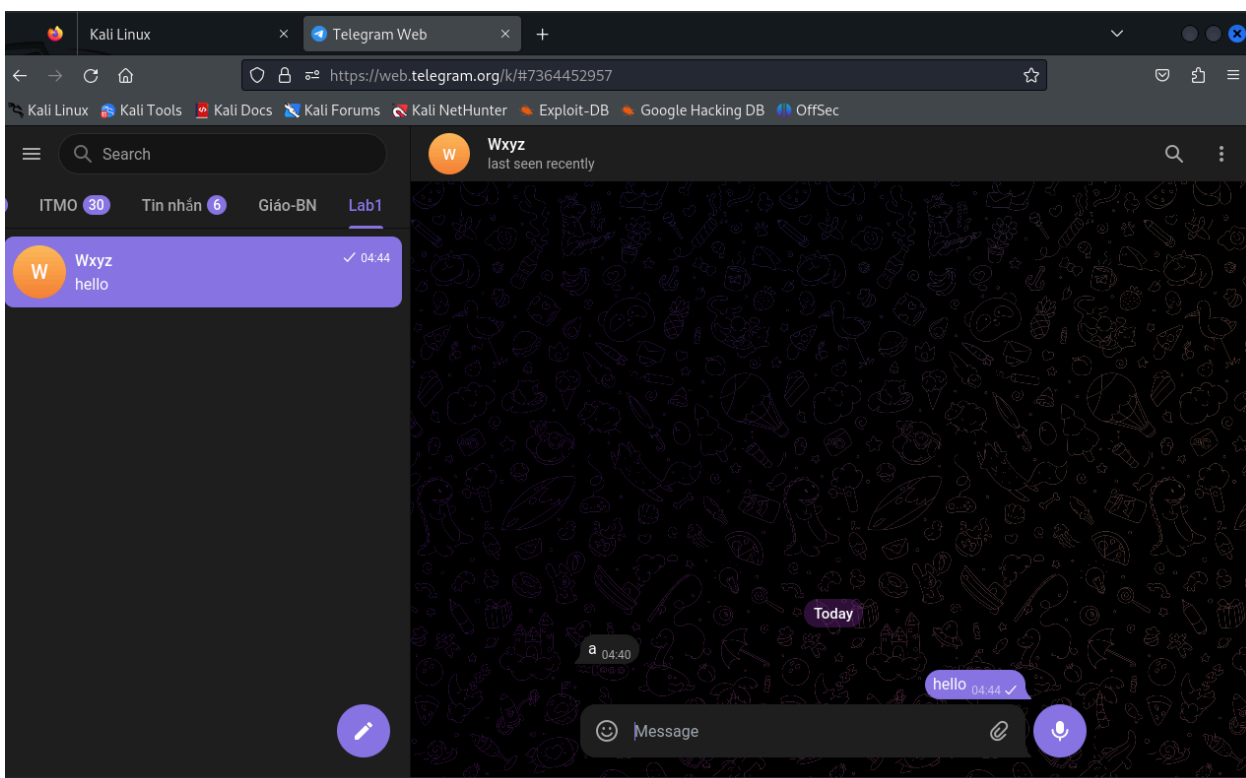


Рисунок 7 - Отправка сообщения

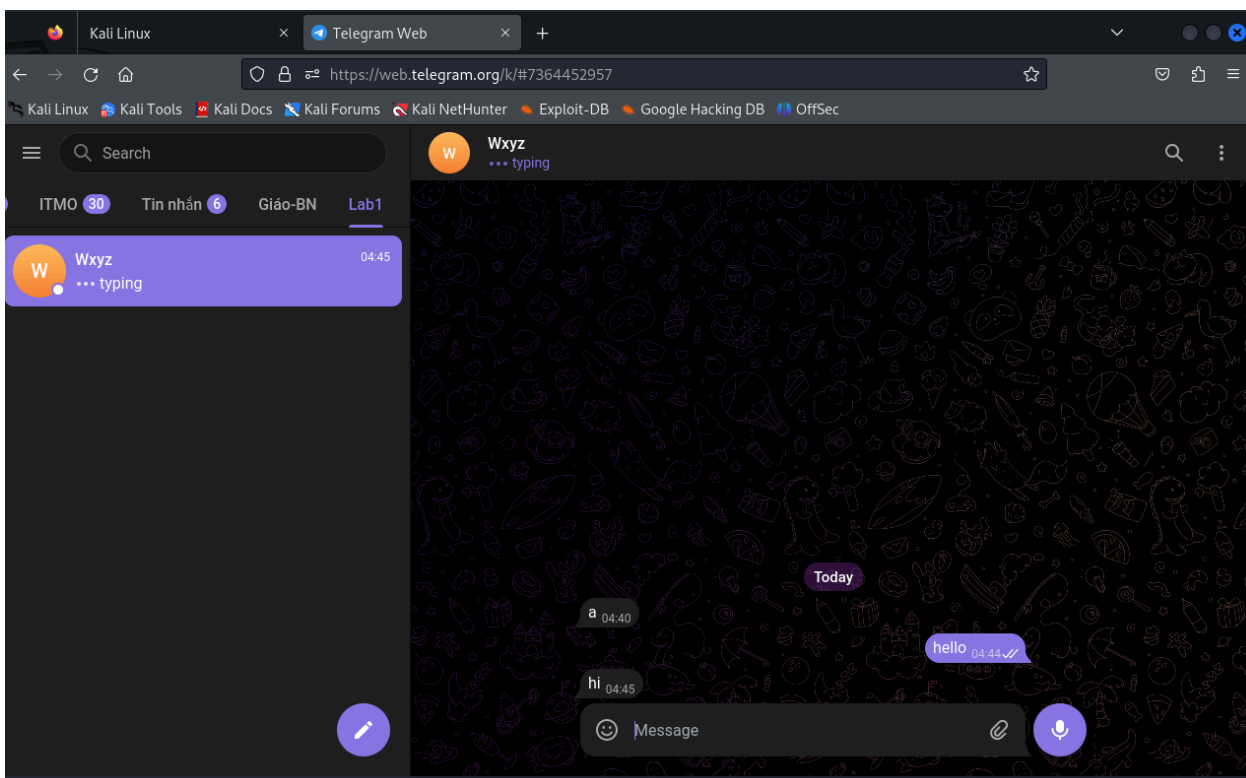


Рисунок 8 - Получение сообщения

- После выполнения всех шагов мы остановили tcpdump с помощью Ctrl + C.

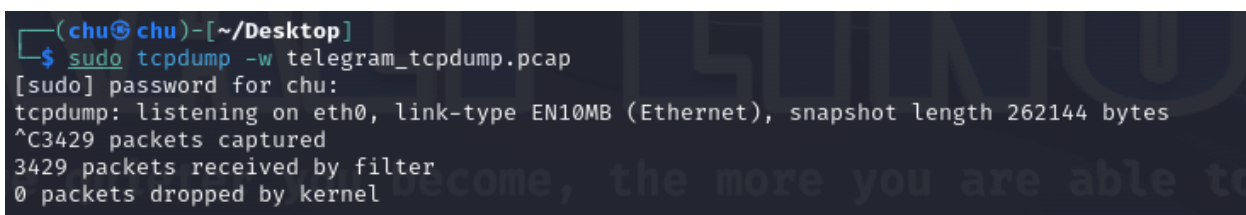


Рисунок 9 - Завершение процесса сканирования трафика

2.3. Сканирование трафика с помощью Wireshark.

- Запускаем сканирование при помощи wireshark

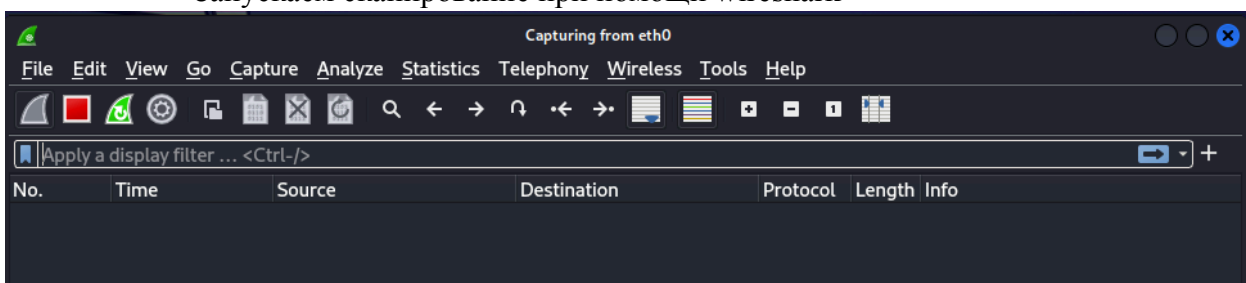


Рисунок 10 - Запуска Wireshark

- Мы продолжаем выполнять шаги так же, как при захвате с помощью tcpdump.

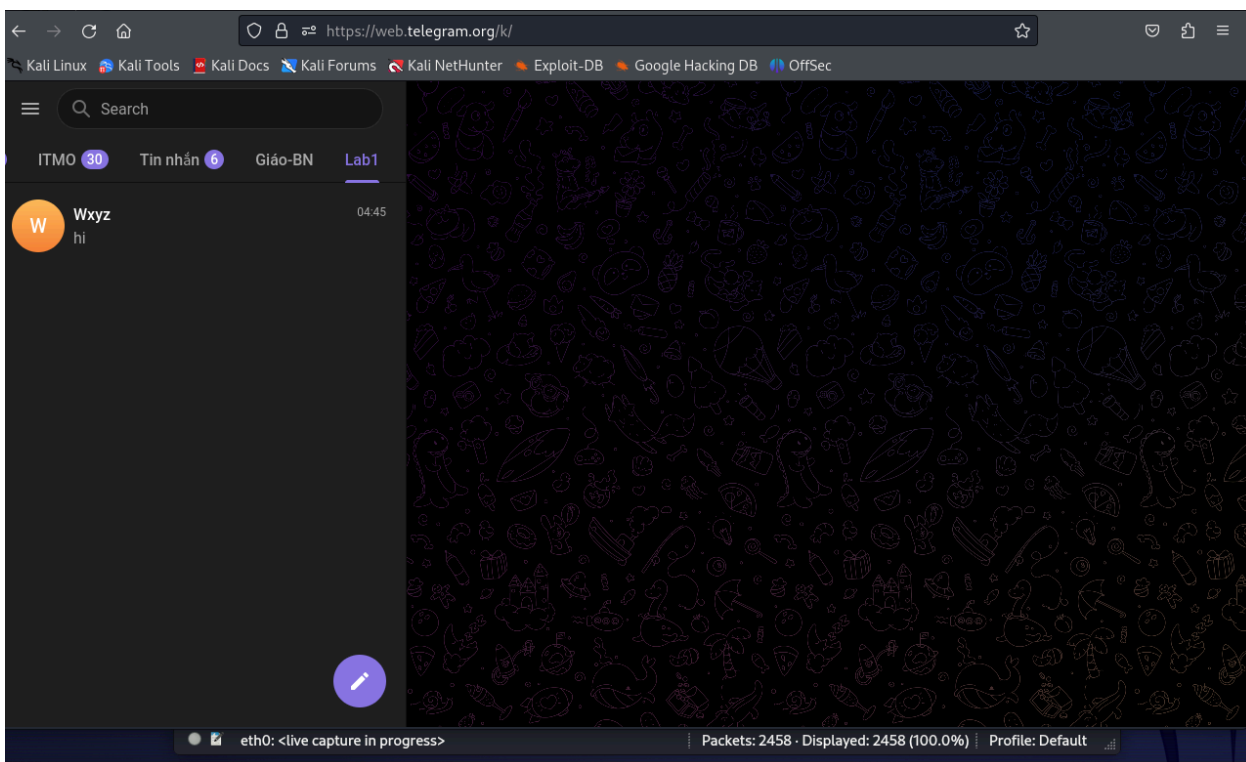


Рисунок 11 - Интерфейс Telegram после входа

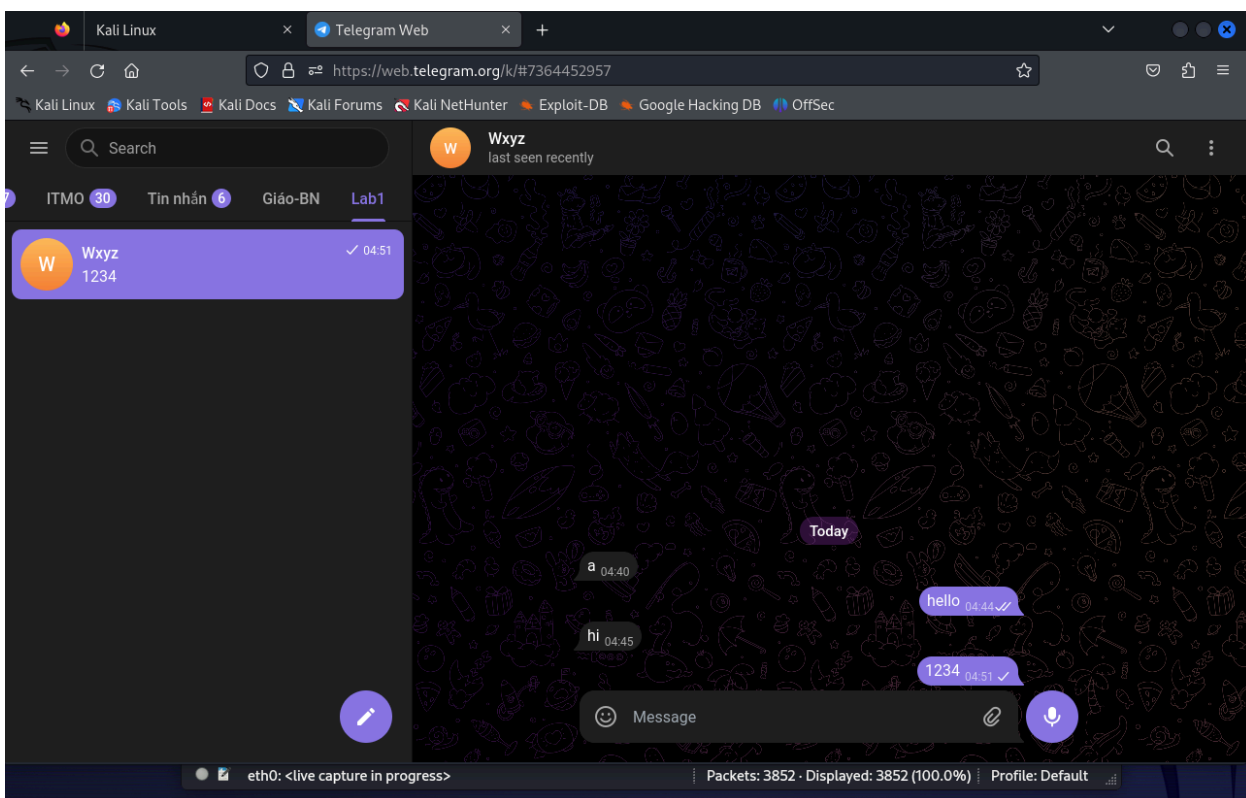


Рисунок 12 - Отправка сообщения

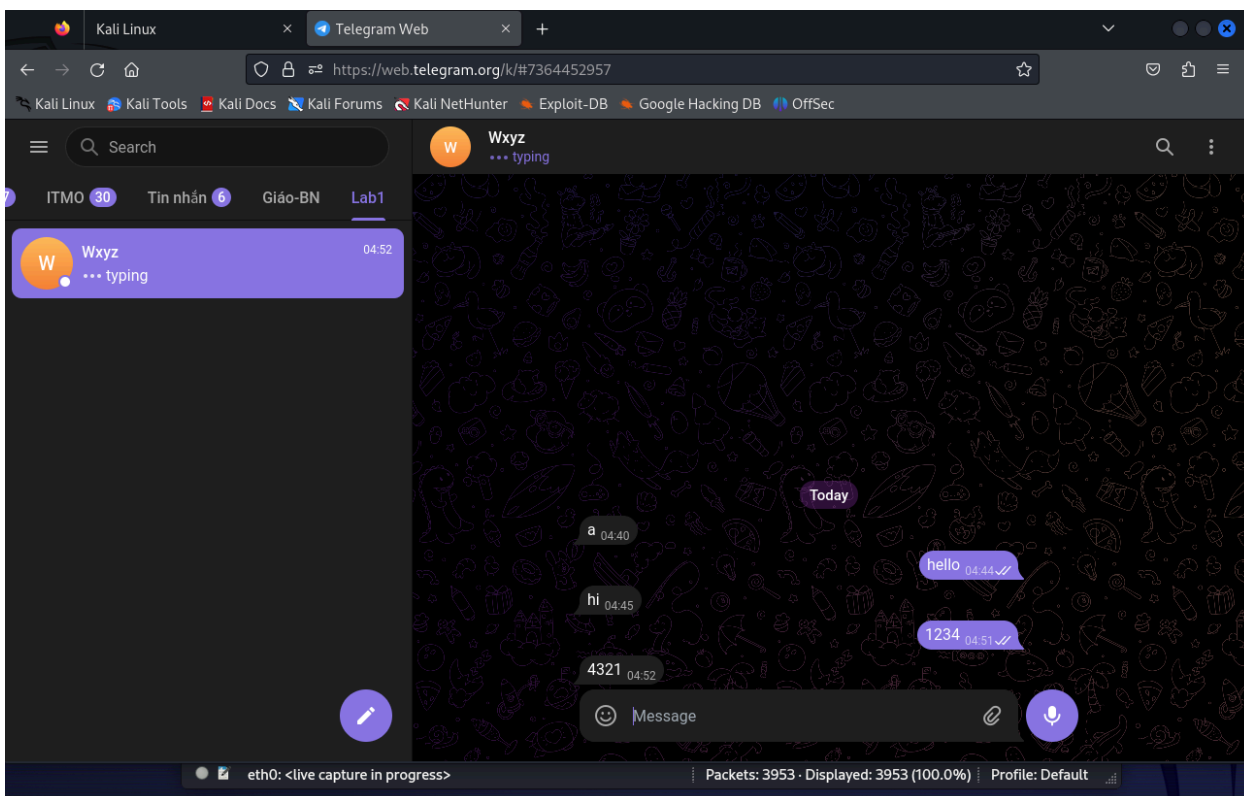


Рисунок 13 - Получение сообщения

- Затем мы закрываем браузер Firefox и останавливаем захват сетевого трафика в Wireshark.

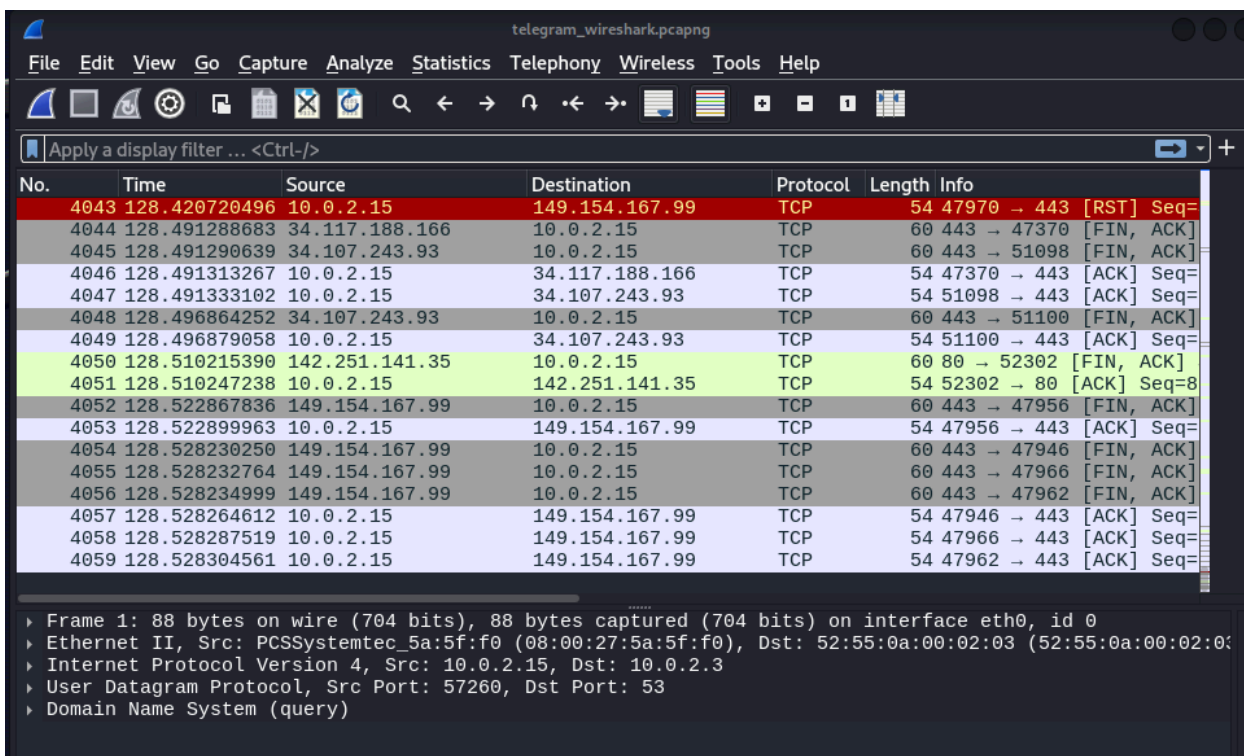
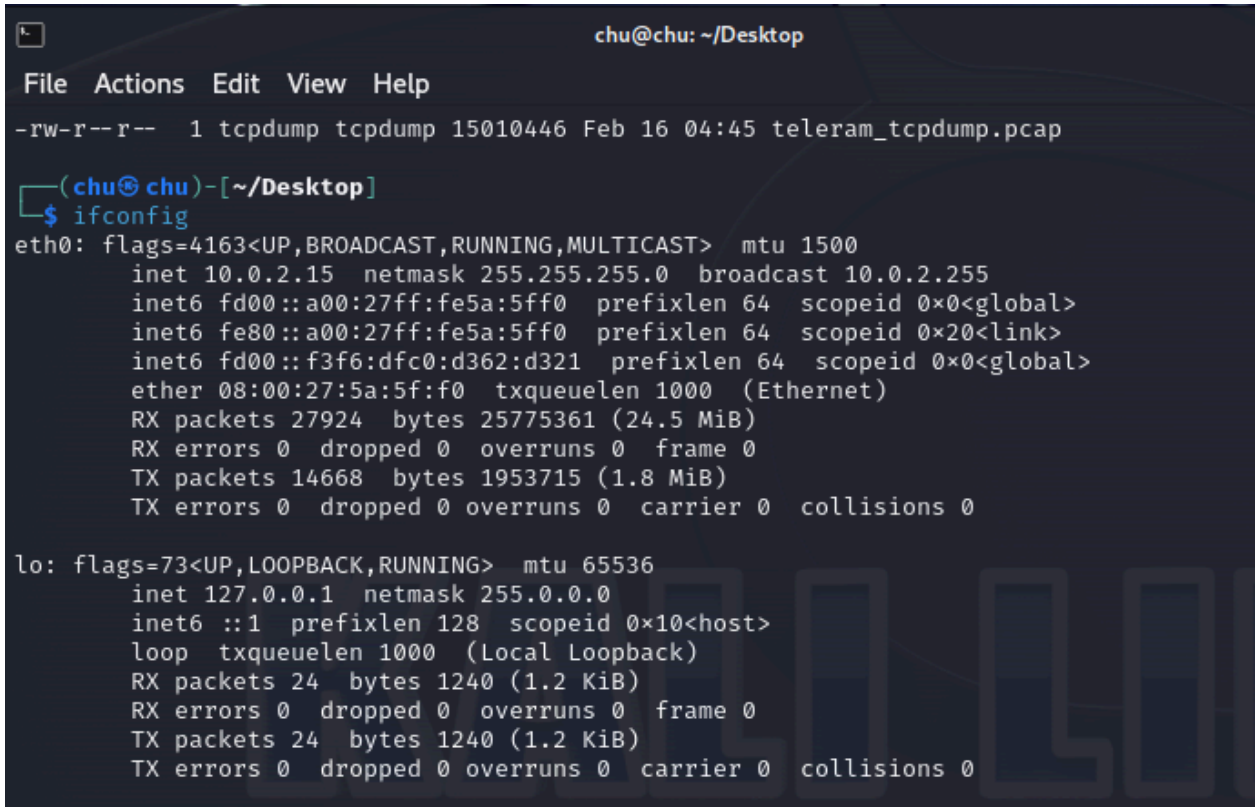


Рисунок 14 - Анализ трафика (Wireshark)

2.4. Проверьте IP-адреса источника и назначения для поиска.

- Мы используем команду `ifconfig`, чтобы просмотреть адрес сетевого интерфейса. Здесь мы видим, что адрес 10.0.2.15 является IP-адресом машины на сетевом интерфейсе `eth0`.



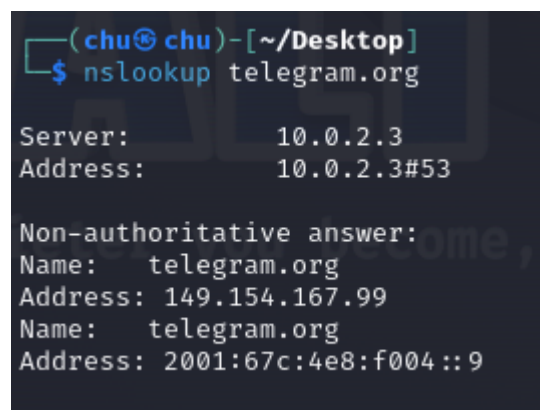
```
chu@chu: ~/Desktop
File Actions Edit View Help
-rw-r--r-- 1 tcpdump tcpdump 15010446 Feb 16 04:45 teleram_tcpdump.pcap

(chu@chu)-[~/Desktop]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fd00::a00:27ff:fe5a:5ff0 prefixlen 64 scopeid 0<global>
    inet6 fe80::a00:27ff:fe5a:5ff0 prefixlen 64 scopeid 0<link>
    inet6 fd00::f3f6:dfc0:d362:d321 prefixlen 64 scopeid 0<global>
    ether 08:00:27:5a:5f:f0 txqueuelen 1000 (Ethernet)
    RX packets 27924 bytes 25775361 (24.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14668 bytes 1953715 (1.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 24 bytes 1240 (1.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 1240 (1.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рисунок 15 - IP-адрес машины

- Далее мы используем `nslookup`, чтобы найти IP-адрес веб-сайта Telegram.
- Мы получаем IP-адрес Telegram: 149.154.167.99



```
(chu@chu)-[~/Desktop]
$ nslookup telegram.org

Server:      10.0.2.3
Address:     10.0.2.3#53

Non-authoritative answer:
Name:   telegram.org
Address: 149.154.167.99
Name:   telegram.org
Address: 2001:67c:4e8:f004::9
```

Рисунок 16 - IP-адрес Telegram

2.5. Анализ полученных результатов с помощью tcpdump

- Мы читаем последние 20 строк файла

```
(chu@chu)-[~/Desktop]
$ tcpdump -r telegram_tcpdump.pcap | tail -n 20
reading from file telegram_tcpdump.pcap, link-type EN10MB (Ethernet), snapshot length 262144
06:09:59.010199 IP 149.154.167.99.https > 10.0.2.15.52176: Flags [F.], seq 7026, ack 1254, win 65535, length 0
06:09:59.010199 IP 149.154.167.99.https > 10.0.2.15.45040: Flags [F.], seq 8060, ack 1475, win 65535, length 0
06:09:59.010200 IP 149.154.167.99.https > 10.0.2.15.45024: Flags [F.], seq 218003, ack 1849, win 65535, length 0
06:09:59.010215 IP 10.0.2.15.52176 > 149.154.167.99.https: Flags [.], ack 7027, win 63360, length 0
06:09:59.010228 IP 10.0.2.15.45040 > 149.154.167.99.https: Flags [.], ack 8061, win 63360, length 0
06:09:59.010232 IP 10.0.2.15.45024 > 149.154.167.99.https: Flags [.], ack 218004, win 36640, length 0
06:09:59.025994 IP 149.154.174.100.https > 10.0.2.15.48886: Flags [P.], seq 5704:6552, ack 1410, win 65535, length 848
06:09:59.025995 IP 149.154.167.99.https > 10.0.2.15.52184: Flags [F.], seq 6995, ack 1249, win 65535, length 0
06:09:59.026015 IP 10.0.2.15.48886 > 149.154.174.100.https: Flags [R], seq 2078181254, win 0, length 0
06:09:59.026037 IP 10.0.2.15.52184 > 149.154.167.99.https: Flags [.], ack 6996, win 63360, length 0
06:09:59.148877 IP 149.154.174.100.https > 10.0.2.15.44660: Flags [P.], seq 93234:93296, ack 5343, win 65535, length 62
06:09:59.148877 IP 149.154.174.100.https > 10.0.2.15.44660: Flags [F.], seq 93296, ack 5343, win 65535, length 0
06:09:59.148896 IP 10.0.2.15.44660 > 149.154.174.100.https: Flags [R], seq 605923747, win 0, length 0
06:09:59.148916 IP 10.0.2.15.44660 > 149.154.174.100.https: Flags [R], seq 605923747, win 0, length 0
06:09:59.149028 IP 149.154.174.100.https > 10.0.2.15.44660: Flags [R.], seq 4208439295, ack 5343, win 0, length 0
06:09:59.156335 IP 149.154.174.100.https > 10.0.2.15.48864: Flags [P.], seq 124218:124280, ack 5522, win 65535, length 62
06:09:59.156335 IP 149.154.174.100.https > 10.0.2.15.48864: Flags [F.], seq 124280, ack 5522, win 65535, length 0
06:09:59.156349 IP 10.0.2.15.48864 > 149.154.174.100.https: Flags [R], seq 112037241, win 0, length 0
06:09:59.156365 IP 10.0.2.15.48864 > 149.154.174.100.https: Flags [R], seq 112037241, win 0, length 0
06:09:59.156494 IP 149.154.174.100.https > 10.0.2.15.48864: Flags [R.], seq 4207479295, ack 5522, win 0, length 0
```

Рисунок 17 - Содержимое файла telegram_tcpdump.pcap

- Мы нашли полный процесс трехстороннего рукопожатия TCP. Мы обнаружили пакеты SYN, отправленные с машины 10.0.2.15 на 149.154.167.99.

```
(chu@chu)-[~/Desktop]
$ tcpdump -r telegram_tcpdump.pcap 'tcp[tcpflags] & (tcp-syn[tcp-ack] &= 0)' and host 10.0.2.15 and host 149.154.167.99 | head -n 20
reading from file telegram_tcpdump.pcap, link-type EN10MB (Ethernet), snapshot length 262144
06:09:06.910684 IP 10.0.2.15.45018 > 149.154.167.99.https: Flags [S], seq 640055681, win 64240, options [mss 1460,sackOK,TS val 3749582594 ecr 0,nop,wscale 7], length 0
06:09:06.910779 IP 10.0.2.15.45024 > 149.154.167.99.https: Flags [S], seq 2799946822, win 64240, options [mss 1460,sackOK,TS val 3749582594 ecr 0,nop,wscale 7], length 0
06:09:06.929934 IP 10.0.2.15.45038 > 149.154.167.99.https: Flags [S], seq 1681089595, win 64240, options [mss 1460,sackOK,TS val 3749582614 ecr 0,nop,wscale 7], length 0
06:09:07.025976 IP 149.154.167.99.https > 10.0.2.15.45024: Flags [S.], seq 79552001, ack 2799946823, win 65535, options [mss 1460], length 0
06:09:07.026011 IP 10.0.2.15.45024 > 149.154.167.99.https: Flags [.], ack 1, win 64240, length 0
06:09:07.028124 IP 10.0.2.15.45024 > 149.154.167.99.https: Flags [P.], seq 1:518, ack 1, win 64240, length 517
06:09:07.028312 IP 149.154.167.99.https > 10.0.2.15.45024: Flags [.], ack 518, win 65535, length 0
06:09:07.028565 IP 149.154.167.99.https > 10.0.2.15.45018: Flags [S.], seq 79616001, ack 640055682, win 65535, options [mss 1460], length 0
06:09:07.028582 IP 10.0.2.15.45018 > 149.154.167.99.https: Flags [.], ack 1, win 64240, length 0
06:09:07.029663 IP 10.0.2.15.45018 > 149.154.167.99.https: Flags [P.], seq 1:518, ack 1, win 64240, length 517
06:09:07.029810 IP 149.154.167.99.https > 10.0.2.15.45018: Flags [.], ack 518, win 65535, length 0
06:09:07.038079 IP 149.154.167.99.https > 10.0.2.15.45038: Flags [S.], seq 79688001, ack 1681089596, win 65535, options [mss 1460], length 0
06:09:07.038111 IP 10.0.2.15.45038 > 149.154.167.99.https: Flags [.], ack 1, win 64240, length 0
06:09:07.041155 IP 10.0.2.15.45038 > 149.154.167.99.https: Flags [P.], seq 1:518, ack 1, win 64240, length 517
06:09:07.041384 IP 149.154.167.99.https > 10.0.2.15.45038: Flags [.], ack 518, win 65535, length 0
06:09:07.141572 IP 149.154.167.99.https > 10.0.2.15.45024: Flags [P.], seq 1:2457, ack 518, win 65535, length 2456
06:09:07.141806 IP 10.0.2.15.45024 > 149.154.167.99.https: Flags [.], ack 2457, win 63360, length 0
06:09:07.143437 IP 149.154.167.99.https > 10.0.2.15.45024: Flags [P.], seq 2457:4097, ack 518, win 65535, length 1640
06:09:07.143451 IP 10.0.2.15.45024 > 149.154.167.99.https: Flags [.], ack 4097, win 63360, length 0
06:09:07.145148 IP 149.154.167.99.https > 10.0.2.15.45024: Flags [P.], seq 4097:5698, ack 518, win 65535, length 1601
tcpdump: Unable to write output: Broken pipe
```

Рисунок 18 - Поиск процесса тройного рукопожатия (Three-way Handshake)

2.6. Анализ полученных результатов с помощью Wireshark

- Мы использовали фильтр: `ip.addr==149.154.167.99` и увидели процесс трехстороннего рукопожатия TCP (TCP 3-Way Handshake) между нашим компьютером (10.0.2.15) и сервером Telegram (149.154.167.99).

ip.addr==149.154.167.99						
No.	Time	Source	Destination	Protocol	Length	Info
159	9.452334863	10.0.2.15	149.154.167.99	TCP	54	47946 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
160	9.453744260	10.0.2.15	149.154.167.99	TLSv1.3	571	Client Hello (SNI=web.telegram.org)
161	9.453981162	149.154.167.99	10.0.2.15	TCP	60	443 → 47946 [ACK] Seq=1 Ack=518 Win=65535 Len=0
162	9.456397162	149.154.167.99	10.0.2.15	TCP	60	443 → 47930 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
163	9.456398229	10.0.2.15	149.154.167.99	TCP	54	47930 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
164	9.457952337	10.0.2.15	149.154.167.99	TLSv1.3	571	Client Hello (SNI=web.telegram.org)
165	9.458164096	149.154.167.99	10.0.2.15	TCP	60	443 → 47930 [ACK] Seq=1 Ack=518 Win=65535 Len=0
166	9.555781073	10.0.2.15	149.154.167.99	TCP	74	47954 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2651963202 TSecr=0 WS=128
167	9.567842002	149.154.167.99	10.0.2.15	TLSv1.3	1282	Server Hello, Change Cipher Spec, Application Data
168	9.567866865	10.0.2.15	149.154.167.99	TCP	54	47946 → 443 [ACK] Seq=518 Ack=1229 Win=63856 Len=0
169	9.568610814	149.154.167.99	10.0.2.15	TCP	4190	443 → 47946 [PSH, ACK] Seq=1229 Ack=518 Win=65535 Len=4096 [TCP segment of a reassembled PDU]
170	9.568625341	10.0.2.15	149.154.167.99	TCP	54	47946 → 443 [ACK] Seq=518 Ack=5325 Win=61920 Len=0
171	9.568763906	149.154.167.99	10.0.2.15	TLSv1.3	427	Application Data, Application Data, Application Data
172	9.568772567	10.0.2.15	149.154.167.99	TCP	54	47946 → 443 [ACK] Seq=518 Ack=5698 Win=63360 Len=0
173	9.579266359	149.154.167.99	10.0.2.15	TLSv1.3	1282	Server Hello, Change Cipher Spec, Application Data
174	9.579289546	10.0.2.15	149.154.167.99	TCP	54	47930 → 443 [ACK] Seq=518 Ack=1229 Win=63856 Len=0
175	9.579907502	149.154.167.99	10.0.2.15	TCP	2922	443 → 47930 [PSH, ACK] Seq=1229 Ack=518 Win=65535 Len=2868 [TCP segment of a reassembled PDU]
176	9.579921190	10.0.2.15	149.154.167.99	TCP	54	47930 → 443 [ACK] Seq=518 Ack=4097 Win=63360 Len=0
177	9.580089013	149.154.167.99	10.0.2.15	TLSv1.3	1655	Application Data, Application Data, Application Data
178	9.5800824378	10.0.2.15	149.154.167.99	TCP	54	47930 → 443 [ACK] Seq=518 Ack=5698 Win=63360 Len=0
181	9.669249303	149.154.167.99	10.0.2.15	TCP	60	443 → 47954 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
182	9.669283665	10.0.2.15	149.154.167.99	TCP	54	47954 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
183	9.670533545	10.0.2.15	149.154.167.99	TLSv1.3	571	Client Hello (SNI=web.telegram.org)

Рисунок 19 - Поиск процесса тройного рукопожатия (Three-way Handshake)

2.7. Сравнение Wireshark и tcpdump

Критерий	Wireshark	tcpdump
Интерфейс	Графический (GUI)	Командная строка (CLI)
Удобство	Интуитивно понятный интерфейс	Требуются знания команд
Функционал	Визуализация данных, фильтрация, статистика	Выводит "сырые" пакеты
ОС	Windows, Linux, macOS	Linux, Unix, macOS, Windows (WSL)
Формат файлов	Открывает и редактирует pcap	В основном для захвата трафика

- ➔ Если нужен удобный анализатор — выбирайте Wireshark.
- ➔ Если нужен быстрый анализ в терминале — используйте tcpdump.

ЗАКЛЮЧЕНИЕ

Было выполнено сканирование сетевого трафика в соответствии с вторым вариантом с помощью инструментов tcpdump и Wireshark. Было изучено использование фильтров Wireshark.

Это позволили получить навыки анализа сетевого трафика.