

Введение

Тема: анализ трафика сети посредством Wireshark.

Цель: научиться устанавливать, проводить базовые настройки и проверять работоспособность Wireshark.

Задачи:

1. Установить wireshark (если ещё не установлен);
2. Выдать права на выполнение (если ещё не выданы);
3. Отправить запрос на свое доменное имя; 4. Убедиться в том, что wireshark поймал трафик;
5. Найти трафик, используя фильтр.

Оборудование, ПО

Таблица 1 – информация об оборудовании

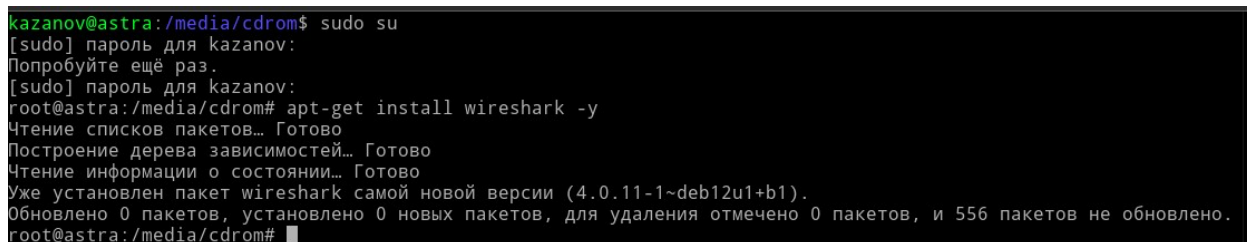
Устройство	Операционная система	IP адрес/Маска	Шлюз	DNS
CLI_A1	Astra Linux SE 1.8.x	10.0.0.1/24	-	au1au.team.lab
CLI_A2	Astra Linux SE 1.8.x	10.0.0.2/24	-	au2au.team.lab
CLI_A3	Astra Linux SE 1.8.x	10.0.0.3/24	-	au3au.team.lab

Выполнение лабораторной работы

Wireshark – это мощный сетевой анализатор, который может использоваться для анализа трафика, проходящего через сетевой интерфейс вашего компьютера. Он может понадобиться для обнаружения и решения проблем с сетью, отладки ваших веб-приложений, сетевых программ или сайтов. Wireshark позволяет полностью просматривать содержимое пакета на

всех уровнях: так вы сможете лучше понять как работает сеть на низком уровне.

Если Wireshark ещё не установлен, это можно сделать при помощи следующей команды см.рис.1: **apt-get install wireshark**

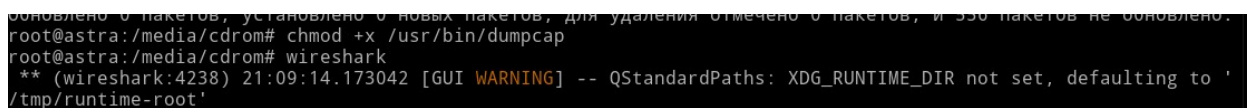


```
kazanov@astra:/media/cdrom$ sudo su
[sudo] пароль для kazanov:
Попробуйте ещё раз.
[sudo] пароль для kazanov:
root@astra:/media/cdrom# apt-get install wireshark -y
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Уже установлен пакет wireshark самой новой версии (4.0.11-1~deb12u1+b1).
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 556 пакетов не обновлено.
root@astra:/media/cdrom#
```

Рисунок 1 – Установка Wireshark

Во время установки нужно выбрать «Да». Далее выдадим права на выполнение см.рис.2:

chmod +x /usr/bin/dumpcap



```
root@astra:/media/cdrom# chmod +x /usr/bin/dumpcap
root@astra:/media/cdrom# wireshark
** (wireshark:4238) 21:09:14.173042 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```

Рисунок 2 – Выдача прав на выполнение

Теперь запустим программу командой и выберем отслеживание трафика со всех сетевых интерфейсов см.рис.3: **wireshark**

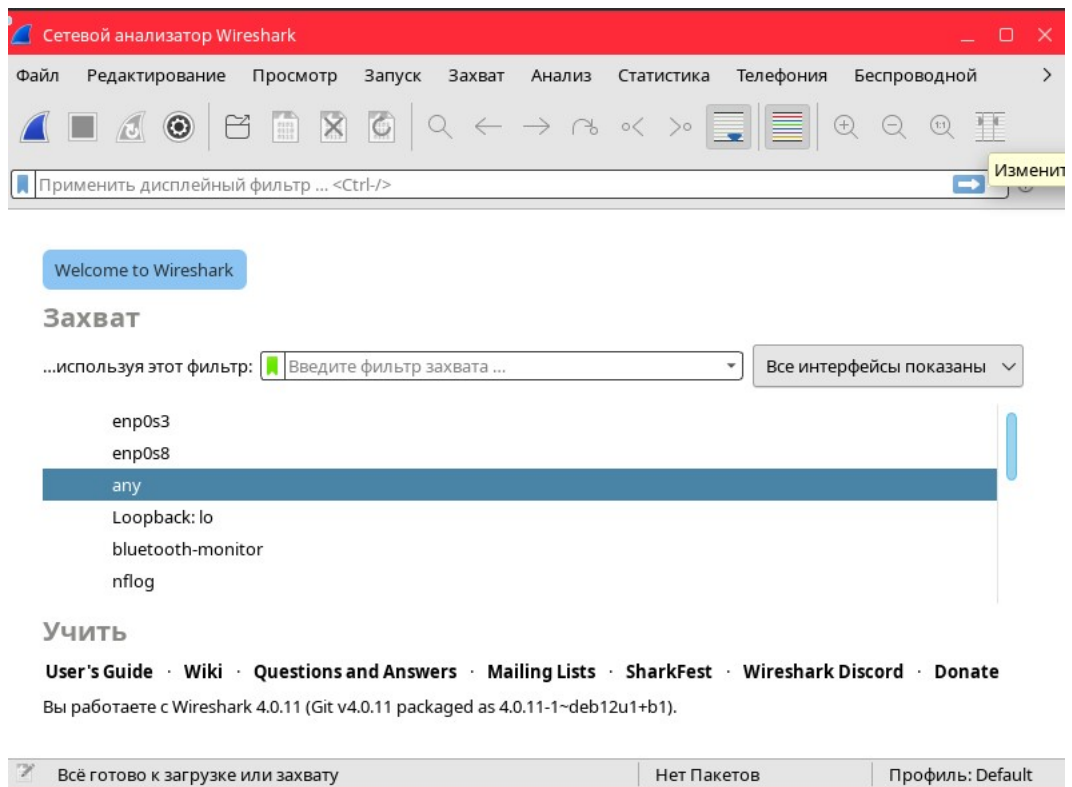


Рисунок 3 – Выбор отслеживания трафика со всех сетевых интерфейсов

Следует произвести пинг своего доменного имени, чтобы убедиться, что wireshark отслеживает трафик

Проверим, что wireshark поймал трафик см.рис.4:

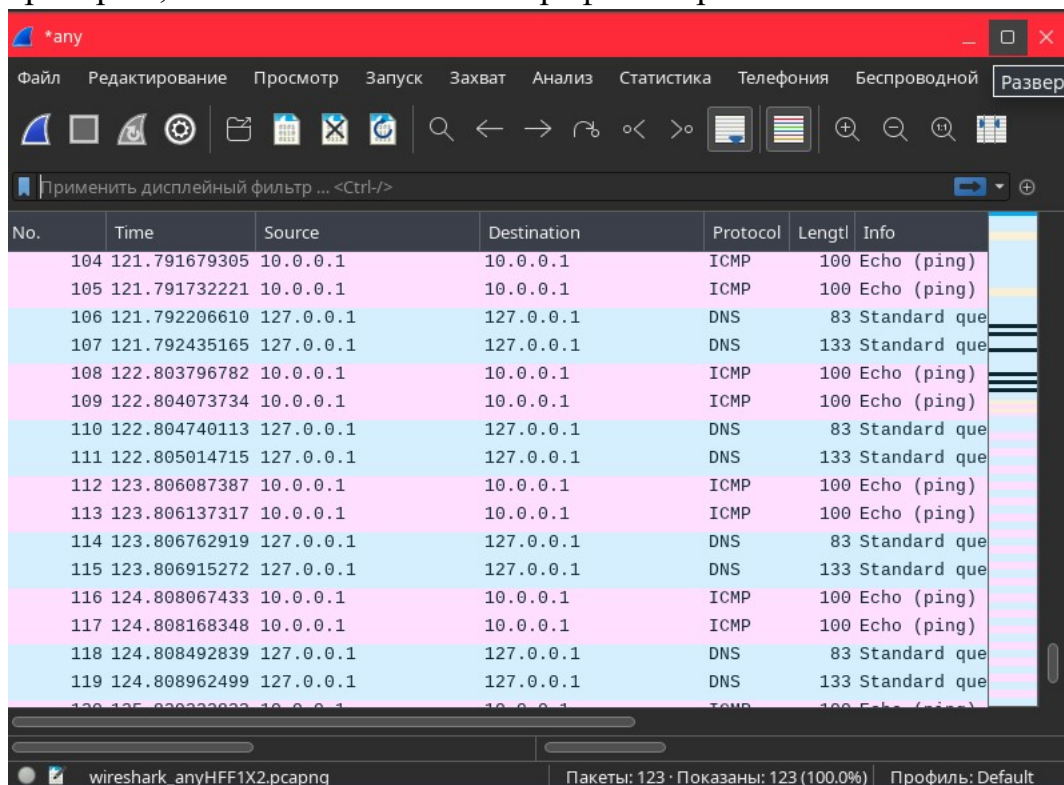


Рисунок 4 – Трафик в wireshark

Зайдём на наш сайт astra и остановим поиск трафика см.рис.6:

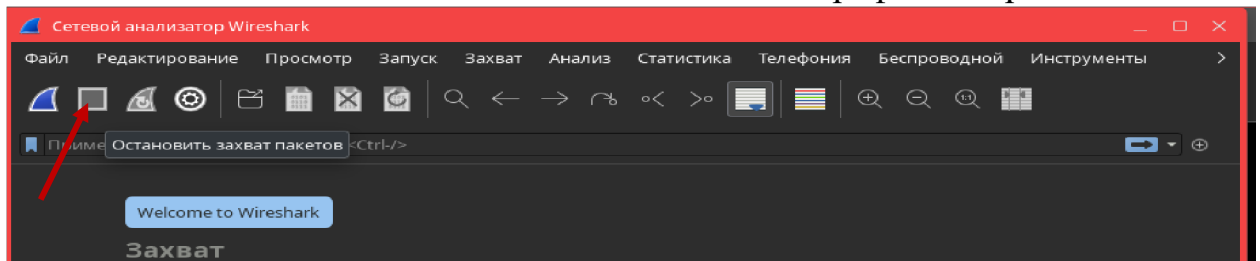


Рисунок 5 – Отключение поиска трафика

С помощью фильтра найдём весь трафик, отправленный с адреса 10.0.0.1 см.рис.6: **Ip.src==10.0.0.1**

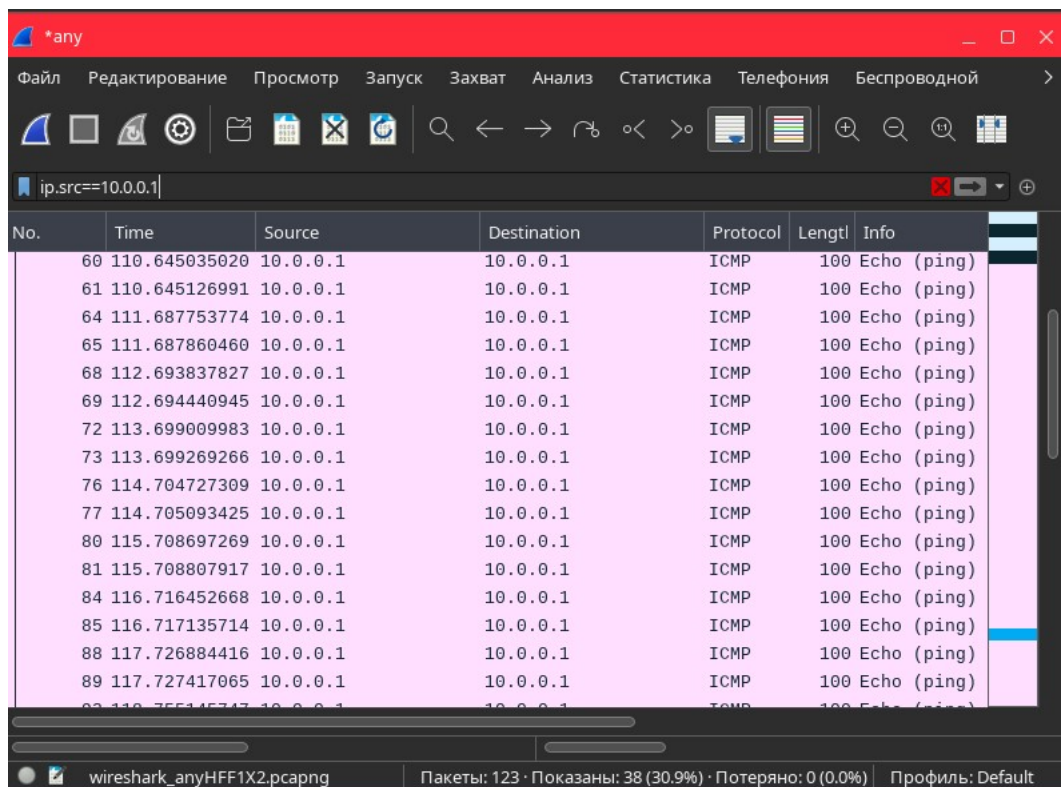


Рисунок 6 – Трафик, отправленный с адреса 10.0.0.1

Далее с помощью фильтра найдём весь трафик, где нет протокола ICMP см.рис.7:

!icmp

13	13.440747409	10.0.2.15	10.0.2.3	DNS	74 Standard query 0x27
14	13.440838838	10.0.2.15	10.0.2.3	DNS	74 Standard query 0xcf
15	13.517643906	10.0.2.3	10.0.2.15	DNS	142 Standard query resp
16	13.703693917	10.0.2.3	10.0.2.15	DNS	142 Standard query resp
17	18.567805800	PcsCompu_cf:ea:0e		ARP	44 Who has 10.0.2.3? T
18	18.569182101	52:55:0a:00:02:03		ARP	66 10.0.2.3 is at 52:5
19	21.638045278	10.0.2.15	10.0.2.3	DNS	78 Standard query 0x7d
20	21.638133726	10.0.2.15	10.0.2.3	DNS	78 Standard query 0x90
21	21.694723264	10.0.2.3	10.0.2.15	DNS	153 Standard query resp
22	21.807915826	10.0.2.3	10.0.2.15	DNS	153 Standard query resp

Рисунок 7 – Трафик, где нет протокола ICMP

Эксперимент:

- Проведём эксперимент по захвату кадров за определенный временной промежуток. Для того чтобы захватить все кадры, начиная с некоторого момента, стоит выполнить команду **frame.time_relative**. Например, попробуем пингануть 8.8.8.8, зайти в браузер и ввести любой запрос, а после выделим кадры, захваченные после 5 секунд после начала захвата см.рис.8:

No.	Time	Source	Destination	Protocol	Length	Info
11	5.009237974	10.0.2.15	8.8.8.8	ICMP	100	Echo (ping)
12	5.028072994	8.8.8.8	10.0.2.15	ICMP	100	Echo (ping)
13	6.010562174	10.0.2.15	8.8.8.8	ICMP	100	Echo (ping)
14	6.037128263	8.8.8.8	10.0.2.15	ICMP	100	Echo (ping)
15	7.011990046	10.0.2.15	8.8.8.8	ICMP	100	Echo (ping)
16	7.040225982	8.8.8.8	10.0.2.15	ICMP	100	Echo (ping)
17	7.327018476	10.0.2.15	10.0.2.3	DNS	86	Standard que
18	7.327098064	10.0.2.15	10.0.2.3	DNS	86	Standard que
19	7.338343575	10.0.2.3	10.0.2.15	DNS	197	Standard que
20	7.338343830	10.0.2.3	10.0.2.15	DNS	209	Standard que
21	7.340719728	10.0.2.15	34.107.221.82	TCP	76	53464 → 80 [
22	7.366108162	34.107.221.82	10.0.2.15	TCP	62	80 → 53464 [
23	7.366269805	10.0.2.15	34.107.221.82	TCP	56	53464 → 80 [
24	7.366423172	10.0.2.15	34.107.221.82	HTTP	370	GET /canonic
25	7.367428112	34.107.221.82	10.0.2.15	TCP	62	80 → 53464 [
26	7.392459363	34.107.221.82	10.0.2.15	HTTP	354	HTTP/1.1 200

wireshark_anyGZZZ02.pcapng | Пакеты: 4164 · Показаны: 4154 (99.8%) · Потеряно: 0 (0.0%) | Профиль: Default

Рисунок 8 – Захват кадров, начиная с 5 секунд

Для того чтобы сделать фильтр по номеру кадра, нужно выполнить команду **frame.number**. Например, выделим кадры с 300 до 550 см.рис.9:

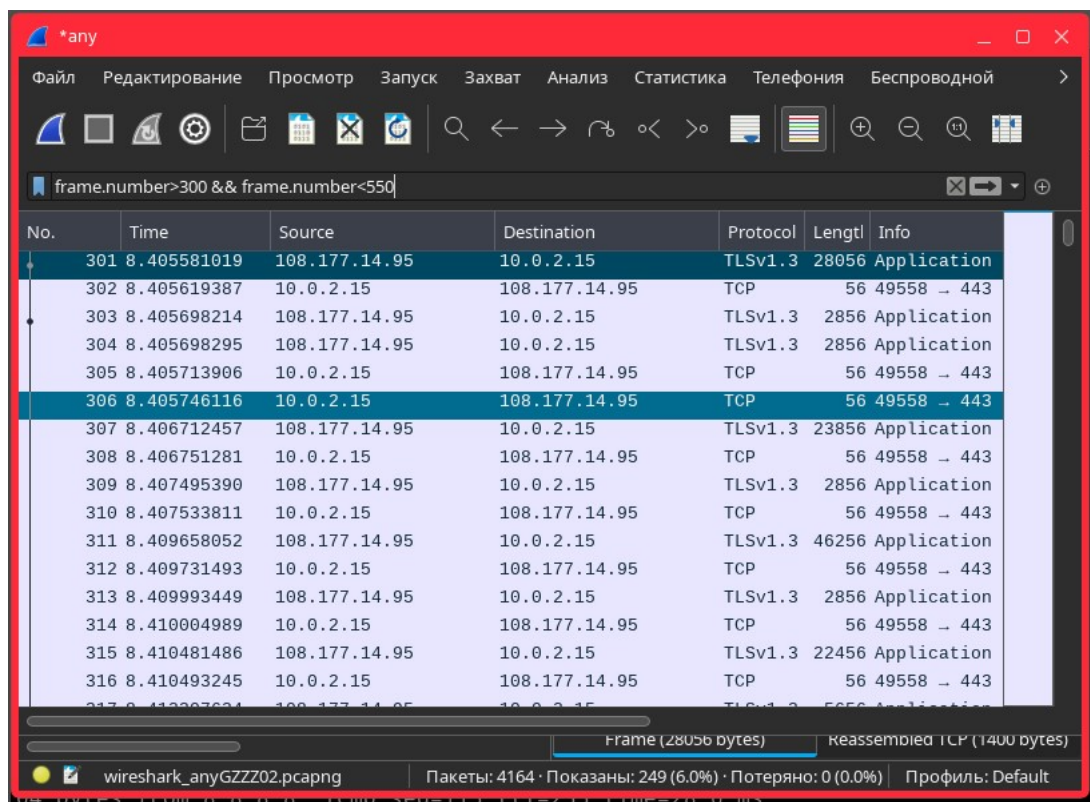


Рисунок 9 – Выделение кадров с 300 до 550

В ходе эксперимента с использованием Wireshark была проанализирована сеть. Были зафиксированы пакеты протокола ICMP, свидетельствующие об успешной передаче и получении запросов и ответов при попытке «пингануть» 8.8.8.8. При входе на веб-страницу в Wireshark также был отслежен трафик, использующий протокол TCP, что подтверждает корректную работу HTTP/HTTPS-запросов. Для анализа использовались фильтры `frame.time_relative` – для захвата кадров с какого-то времени, а также `frame.number` – для захвата кадров по номеру.

В ходе эксперимента я выделил, что следует использовать временные фильтры для анализа подозрительной активности, если таковая имеется.

Вывод

В ходе лабораторной работы была изучена установка и настройка Wireshark для анализа сетевого трафика на платформе Astra Linux. Были выполнены базовые операции с использованием командной строки, включая пинг доменного имени и фильтрации трафика. Изучение различных фильтров Wireshark показало их значимость для быстрого поиска необходимой информации.

Контрольные вопросы

1. Для чего используется Wireshark?

Wireshark используется для анализа сетевого трафика, проходящего через сетевой интерфейс компьютера. Он позволяет выявлять и устранять проблемы в сети, отлаживать веб-приложения, сетевые программы и сайты, а также исследовать работу сетевых протоколов на низком уровне.

2. Что такое фильтры?

Фильтры в Wireshark – это инструменты, которые позволяют выделять определённые пакеты из общего потока трафика, основываясь на заданных критериях. Они упрощают анализ данных, сокращая объём просматриваемой информации до наиболее релевантной.

3. Какие протоколы существуют?

Существует множество сетевых протоколов, включая ARP, ICMP, TCP, UDP, HTTP, HTTPS, FTP, DNS, BGP, OSPF, EIGRP, VRRP, а также протоколы для работы с VLAN (например, 802.1Q) и беспроводные протоколы (например, Wi-Fi).