Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Институт радиоэлектроники и информационных технологий – РтФ

Школа профессионального и академического образования

Отчет по дисциплине   
«Безопасность компьютерных сетей»

Лабораторная работа №4  
«Специальные техники анализа трафика в Wireshark»

Студенты: Клоченко И.Е., Юсупов Д.А.

Преподаватель: Ваулин С.С.

Группа: РИ-300024

Екатеринбург

2023

**Оглавление**

[1. Просмотр TLS-трафика 3](#_Toc134052604)

[2. Работа с HTTP2 потоком 6](#_Toc134052605)

[3. Дополнительная информация об IP адресах 8](#_Toc134052606)

[4. Другие источники трафика 11](#_Toc134052607)

[Вывод 20](#_Toc134052608)

**Цель работы:**

Знакомство с сетевым анализатором трафика *Wireshark*, изучение приемов анализа трафика.

**Ход работы:**

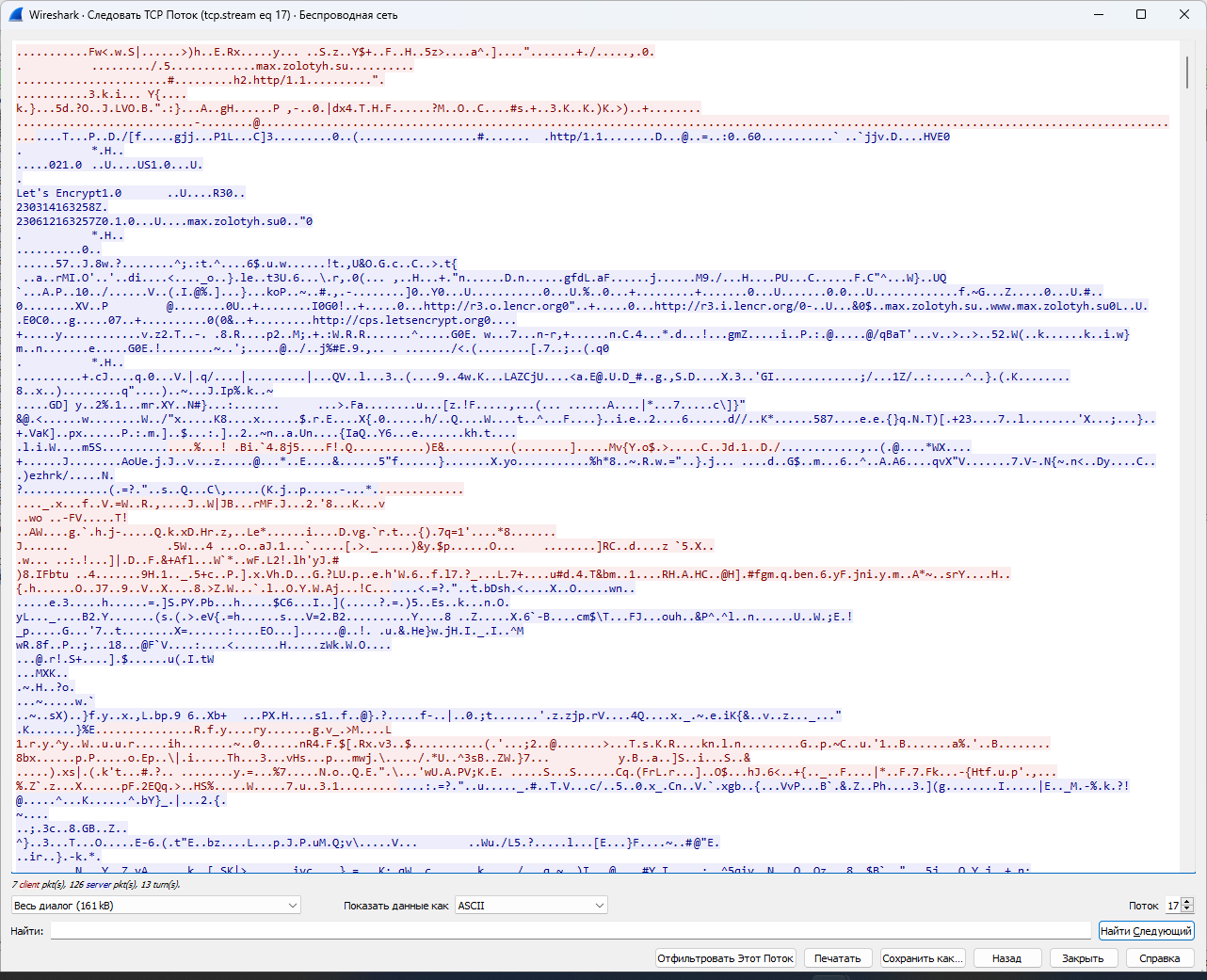
## Просмотр TLS-трафика

* 1. Откроем браузер *FireFox*. В консоли выполним команды:
* *set* *SSLKEYLOGFILE*=%*HOMEDRIVE*%%*HOMEPATH*%*\keys*.*log*
* “%*ProgramFiles*%*\Mozilla* *Firefox\firefox*.*exe*”

Первая строка в файле *keys*.*log*:

# *SSL*/*TLS* *secrets* *log* *file*, *generated* *by* *NSS*.

* 1. Запустим захват трафика и откроем в *FireFox* страницу [*https*://*max*.*zolotyh*.*su*/*lab*03](https://max.zolotyh.su/lab03). В открывшейся форме введем свой псевдоним в качестве имени пользователя и пароль. Остановим захват трафика. Отфильтруем трафик отправки данных на сайт по его *IP*-адресу, а затем отобразим содержимое *TCP*-потока.



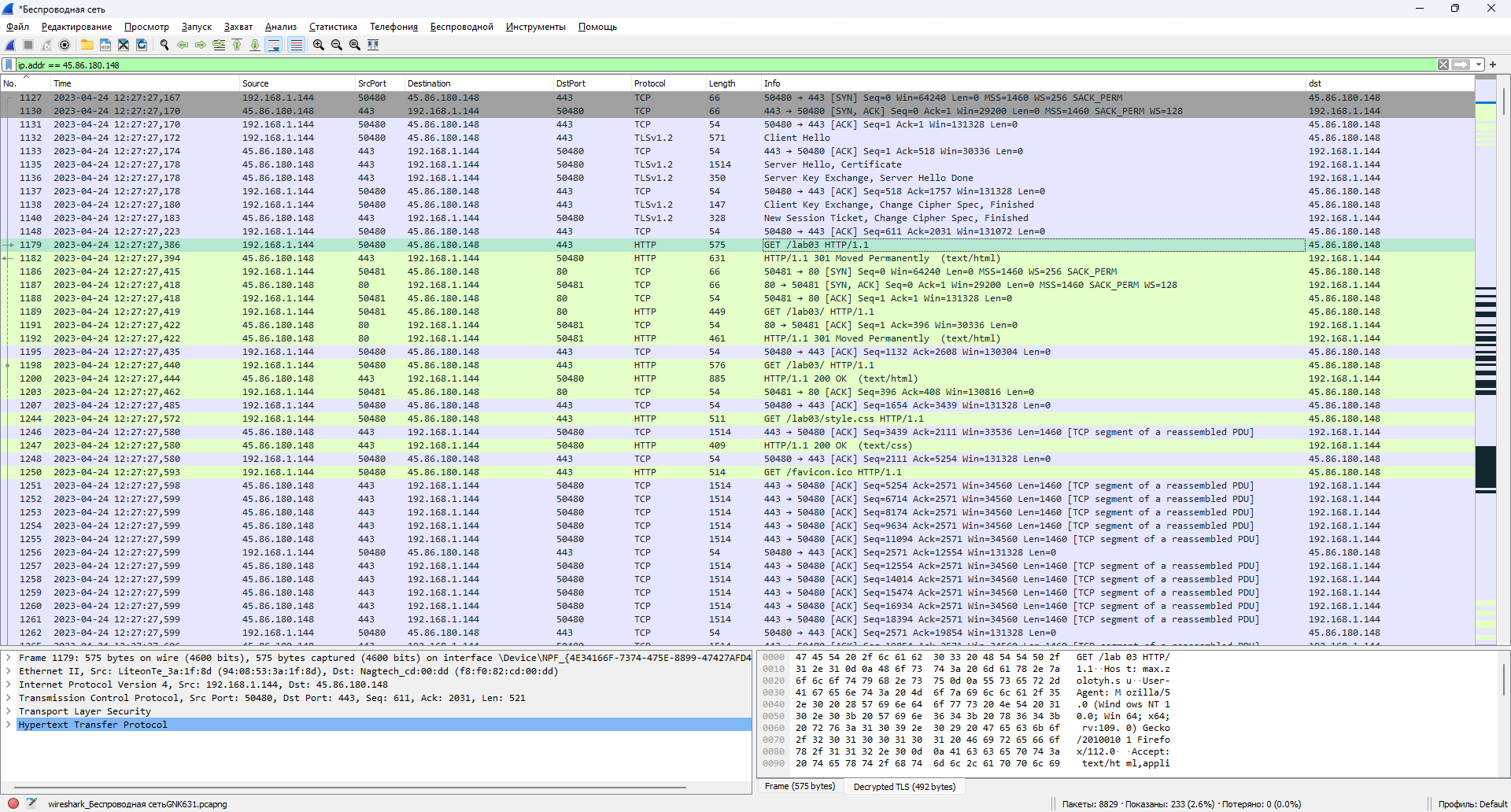
**Рисунок 1**. Содержимое TCP потока

Передаваемая информация не распознаваема.

* 1. Переходим «*Edit* – *Preferences* – *Prot*ОС*ols* – *TLS*». В поле «(*Pre*)-*Master*-*Secret* *log* *filename*» указываем полный путь к файлу ключей, созданному в п1.1.

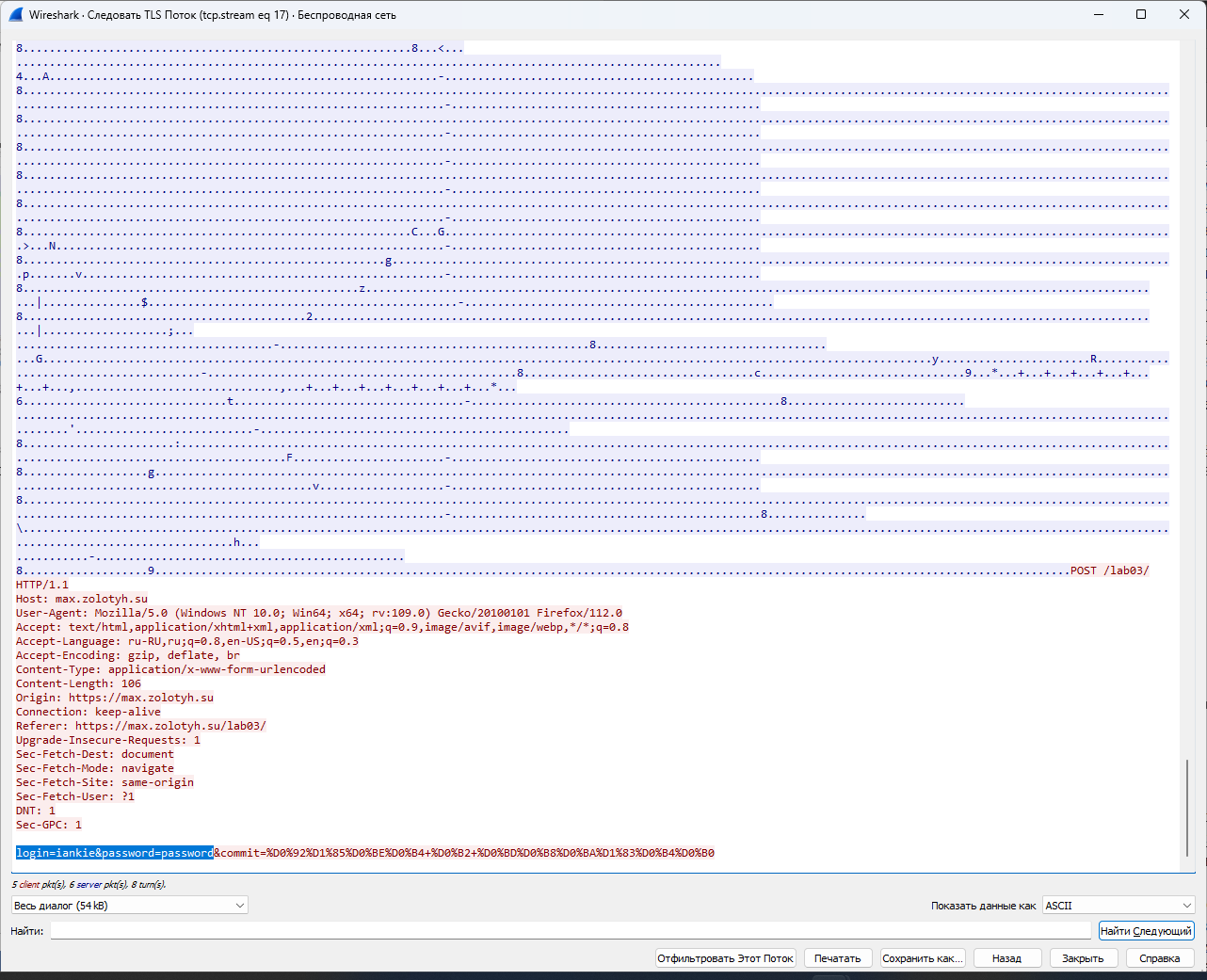
Уберем фильтр по *TCP* потоку и вернем фильтр по *IP* адресу.

В силу большого количества захваченных пакетов, зафиксируем только первые видимые в списке:



**Рисунок 2**. Захваченные пакеты

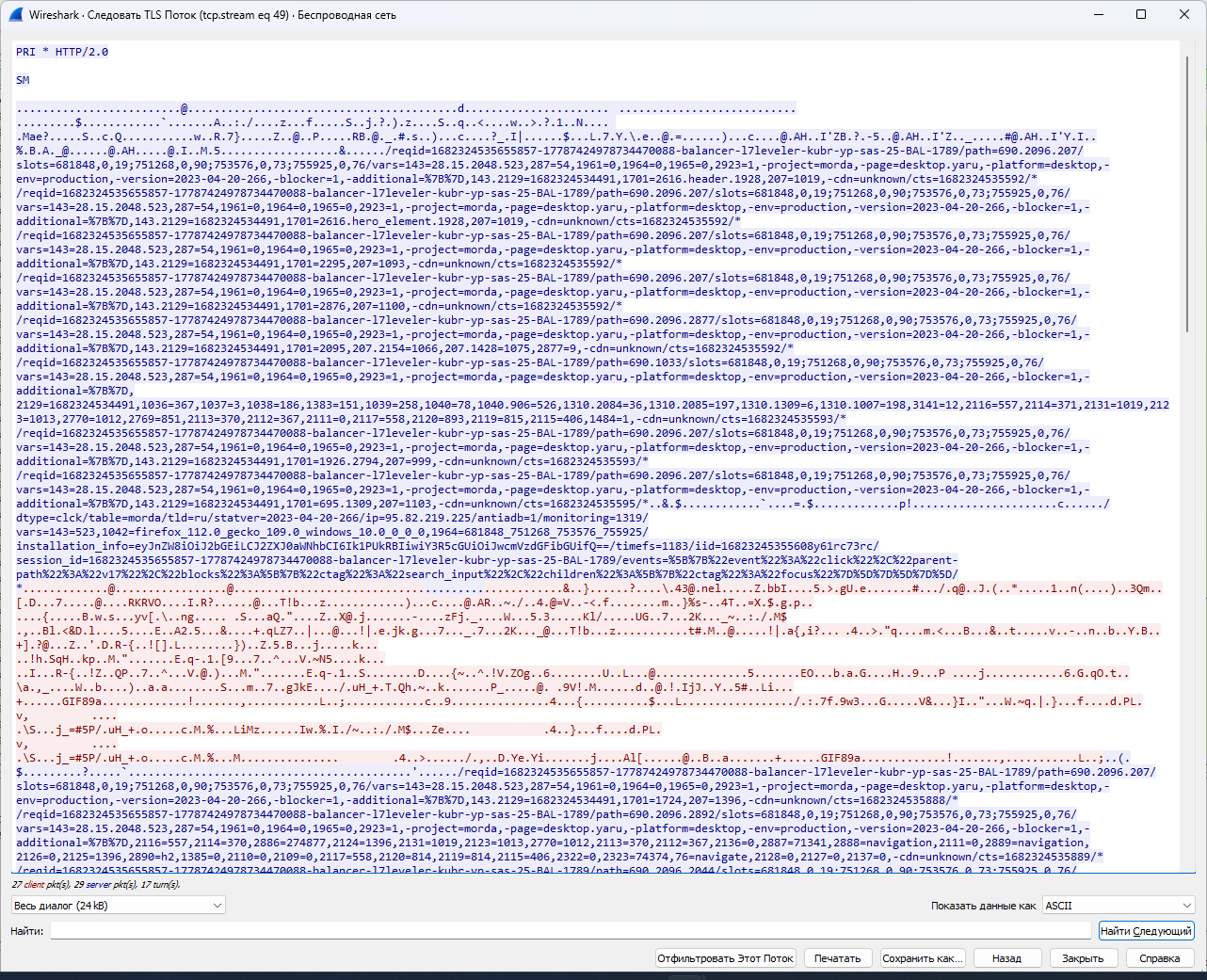
* 1. В списке захваченных пакетов найдем *http* пакет с *POST* запросом. Следуем по *TLS* потоку.



**Рисунок 3**. TLS поток

## Работа с HTTP2 потоком

* 1. Запустим захват трафика и откроем в *FireFox* страницу [*https*://*ya*.*ru*](https://ya.ru). В поисковой форме введем любую комбинацию из латинских символов и выполняем поиск. Останавливаем захват трафика. Восстанавливаем *TLS* поток отправки поискового запроса.



**Рисунок 4**. TLS поток (HTTP2)

* 1. Применим в качестве фильтра отображения выражение «*http*2». Далее ищем пакет, в котором содержится строка, которая была введена в предыдущем пункте с методом *HTTP* запроса *POST*.

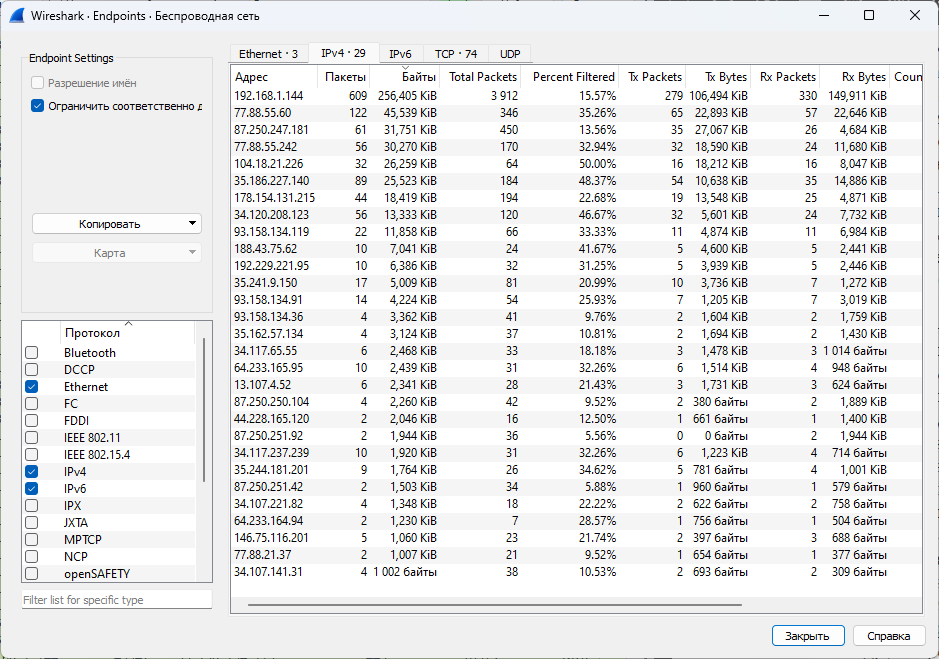
При анализе сайта *max*.*zolotyh*.*su* нам встречались похожие заголовки, например *Accept*, *Accept*-*Language*, *Accept*-*Encoding*.

* 1. Составим фильтр отображения, который будет отображать только *HTTP*2 заголовки, содержащие методы *POST* и *GET*:

(*http*2.*flags* == 0*x*24) *or* (*http*2.*flags* == 0*x*25)

В столбце *Info* можно увидеть различные относительные *URL*, для некоторых из них можно попытаться объяснить их роль:

* /*click*/*click* (или /*click*/*counter*) идут в связке с методом *POST* и, возможно, связаны с «кликами» пользователя в поисковой системе, например нажатие на кнопку «Поиск», переключение по вкладкам «Поиск», «Картинки», «Видео» и тд.
* /*search*/*text*=<*some* *text*>… - собственно перенаправление на страницу поиска по введенному тексту. Тоже идет в связке с методом *POST*.
  1. Применим фильтр отображения «*http*2 || *http*», затем вызовем статистику по отправителям и получателям трафика. Путем перехода на вкладку *IPv*4, включения ограничения отображения фильтра и сортировки по убыванию байт получаем рейтинг узлов.



**Рисунок 5**. Рейтинг узлов по байтам (IPv4)

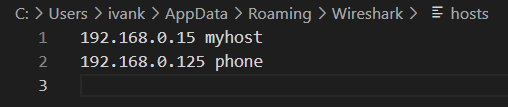
* 1. Откроем анализатор *DNS* пакетов в захваченном трафике. Найдем имена узлов из рейтинга, полученного на предыдущем шаге.

Таблица 1. Соответствие IP адресов узлов именам узлов

|  |  |
| --- | --- |
| **IP адрес узла** | **Имя узла** |
| 93.158.134.119 | *mc*.*yandex*.*ru* |
| 93.158.134.36 | *favicon*.*yandex*.*ru* |
| 93.158.134.91 | *yabs*.*yandex*.*ru* |
| 87.250.251.92 | *cryprox*.*yandex*.*net* |
| 87.250.251.42 | *egress*.*yandex*.*net* |
| 87.250.250.104 | *csp*.*yandex*.*net* |
| 87.250.247.181 | *avatars*.*mds*.*yandex*.*net* |
| 77.88.21.60 | *yandex*.*ru* |
| 77.88.21.37 | *mail*.*yandex*.*ru* |

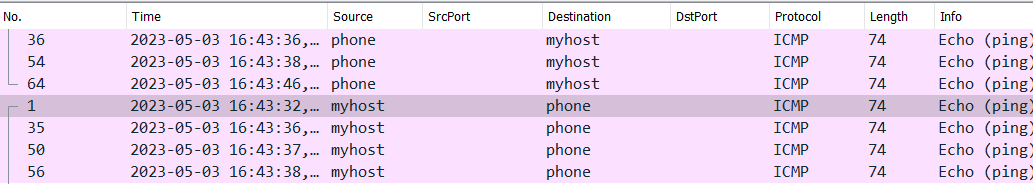
## Дополнительная информация об IP адресах

* 1. Создаем файл *hosts*. В созданный файл вносим информацию о рабочем месте и устройствах, которые есть поблизости - смартфон).



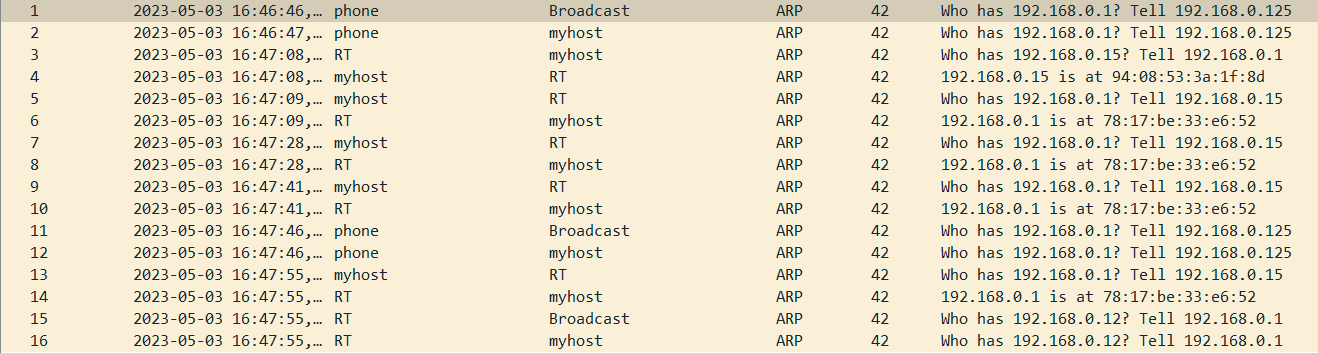
**Рисунок 6**. Содержимое файла hosts

* 1. Перезапускаем *Wireshark* и запускаем захват трафика. Выполняем эхо-тестирование смартфона (запись в *hosts* как *phone*). Включаем разрешение сетевых имен (*Resolve* *Network* *Addresses*). Включение этого параметра отображает в списке захваченных пакетов *ICMP* (конкретно для случая эхо-тестирования) имена устройств из файла *hosts*.



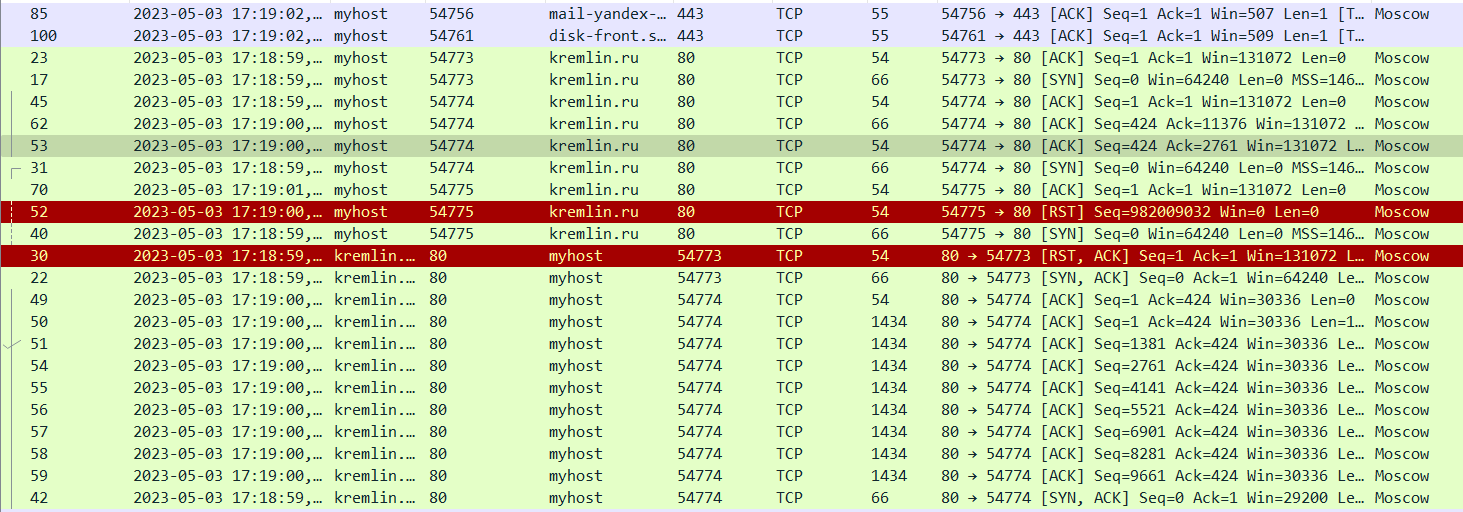
**Рисунок 7**. ICMP пакеты с именами из hosts

* 1. Включаем захват пакетов с фильтром захвата, пропускающим только *ARP* пакеты. Собираем пакеты в течение 2-3 минут. Останавливаем захват. Узлы, проявляющие активность в сети.



**Рисунок 8**. Узлы, проявляющие активность в сети

* 1. Подключаем актуальные базы геолокации к анализатору. Запускаем захват трафика. В браузере открываем сайт *kremlin*.*ru* (при захвате пакетов от узла *yandex*.*ru* не передается город). Останавливаем захват трафика. Находи среди захваченных пакетов, любой отправленный *IP* адресу сайта. В декодере находим информацию о геолокации адреса-получателя. Адрес относится к «*The* *Federal* *Guard* *Service* *of* *the* *Russian* *Federation*». Расположен в Москве. Выбираем строку с указанием города в качестве столбца в списке отображения захваченных пакетов.

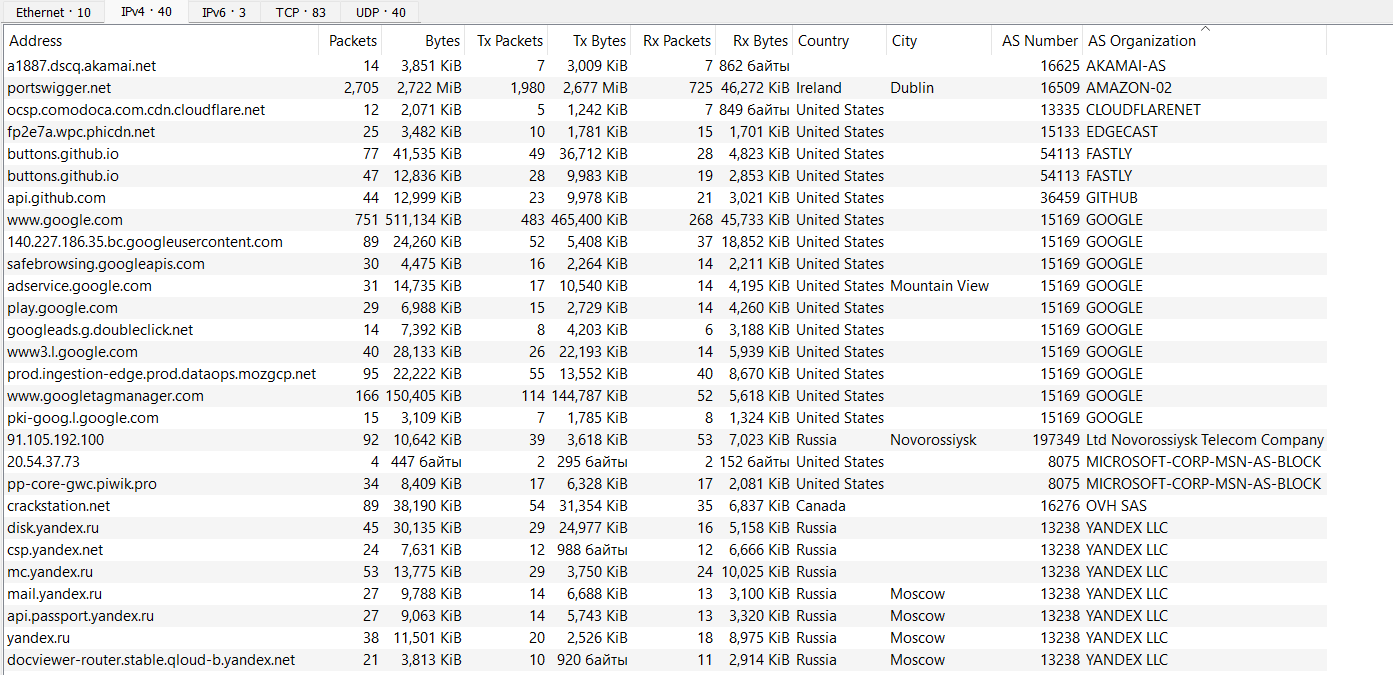


**Рисунок 9**. Установка столбца с названием города узла

* 1. Фильтр отображения, позволяющий выбирать только пакеты, отправленные к *IP* адресам, зарегистрированным за пределами России:

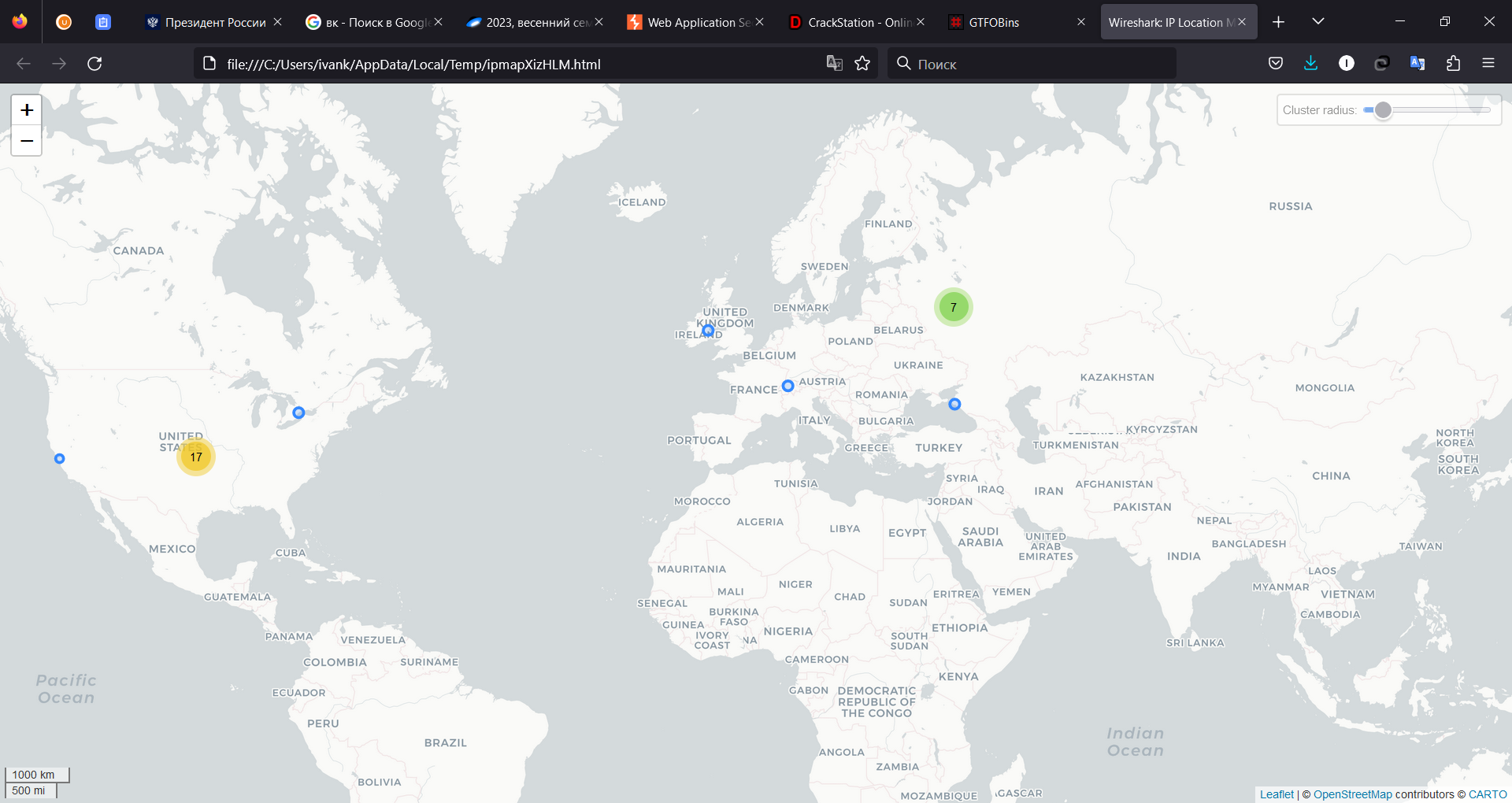
*ip*.*geoip*.*dst\_country* != "*Russia*"

* 1. Удаляем фильтр отображения и запускаем захват пакетов. В течение нескольких минут посещаем несколько различных веб-сайтов. Переходим к конечным точкам. Включаем разрешение имен и сортируем список по организациям-владельцам *AS*.



**Рисунок 10**. Посещенные сайты (Endpoints)

* 1. При помощи *Map* отображаем расположение адресов на карте.



**Рисунок 11**. Адреса посещенных сайтов на карте

Адреса находятся на карте именно там, где и ожидалось их увидеть.

## Другие источники трафика

* 1. Знакомимся с *IP* телефонией. Скачиваем готовый пример для анализа «*MagicJack*+ *short* *test* *call*» и открываем. Восстанавливаем *UDP* потоки для *SIP* и *RTP*.



**Рисунок 12**. Восстановленный UDP поток для SIP и RTP (1)



**Рисунок 13**. Восстановленный UDP поток для SIP и RTP (2)

* 1. Очищаем фильтр отображения. Запускаем анализатор звонков и с его помощью прослушиваем переданный разговор. Стенограмма разговора:

216.234.64.16: <Долгий гудок>

192.168.0.10: <короткий гудок>

192.168.0.10: <короткий гудок >

216.234.64.16: <Подняли трубку>

192.168.0.10: <*Test*>

216.234.64.16: <*Test*>

192.168.0.10: <*One*>

216.234.64.16: <*One*>

192.168.0.10: <*Two*>

216.234.64.16: <*Two*>

192.168.0.10: <*Three*>

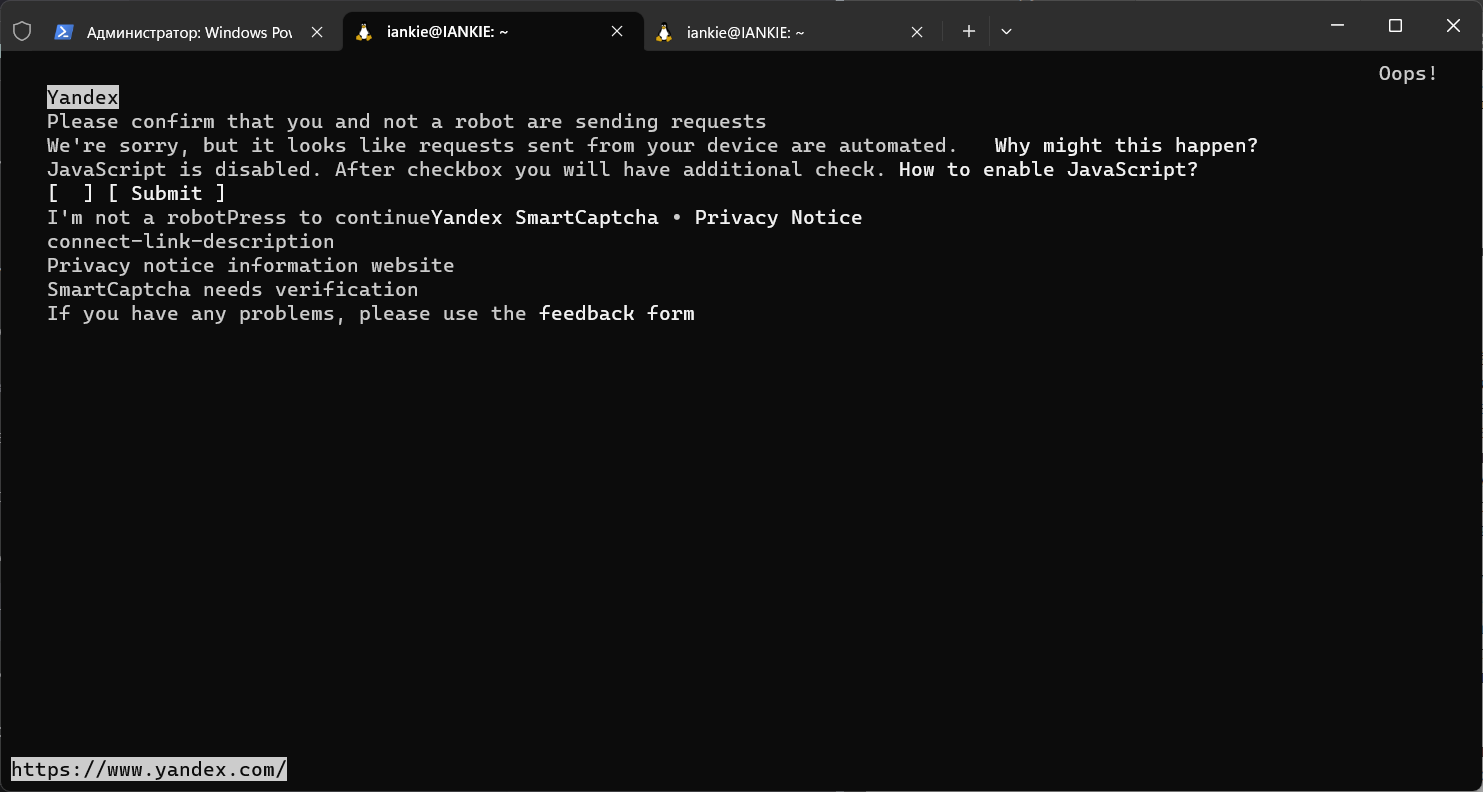
216.234.64.16: <*Three*>

216.234.64.16: <Положили трубку>

* 1. Создаем дамп трафика на серверной *Linux* машине (в силу неизвестных обстоятельств ВМ не хотела проводить трассировку до яндекса, да и до всех узлов, поэтому было принято решение провести эти шаги на *WSL* *Kali* *Linux*).
  2. Используемая команда:

*tcpdump* -*v* *host* 172.26.131.199 *and* *not* *arp* -*w* ./*ivank*.*cap*

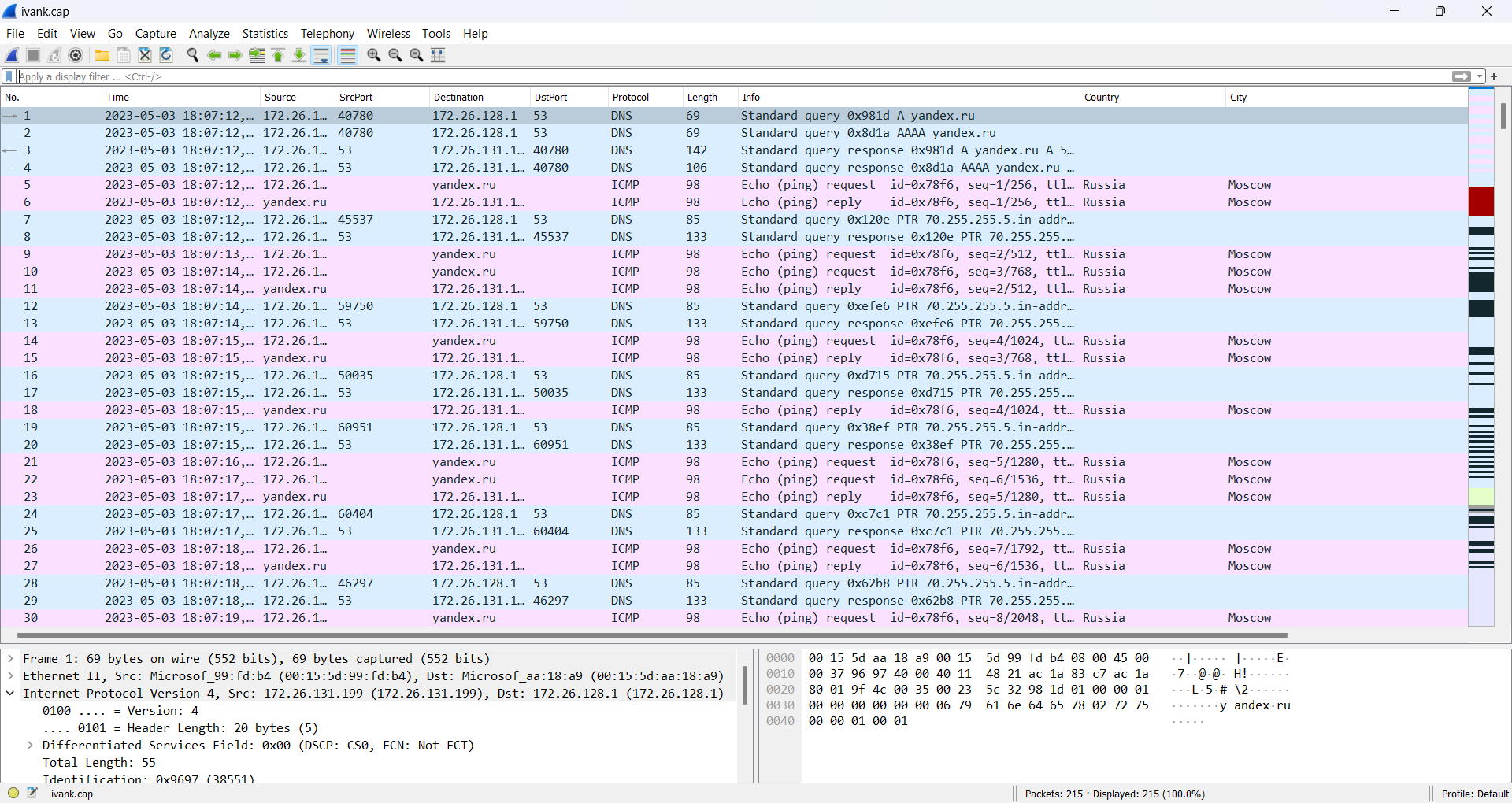
* 1. Не прерывая захват пакетов в первой консоли, входим во вторую консоль и выполняем эхо тестирование узла *yandex*.*ru*, трассировку к нему. Затем открываем заглавную страницу сайта в текстовом браузере *links*.



**Рисунок 14**. Текстовый браузер links

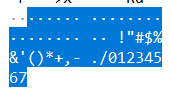
Возвращаемся в первую консоль и останавливаем захват трафика. Было захвачено 215 пакетов.

* 1. Запускать приложение *WinSCP* не пришлось потому что в проводнике *Windows* можно довольно просто найти файл дампа на *WSL* машине. Открываем файл дампа в *Wireshark*.



**Рисунок 15**. Пакеты, захваченные tcpdump

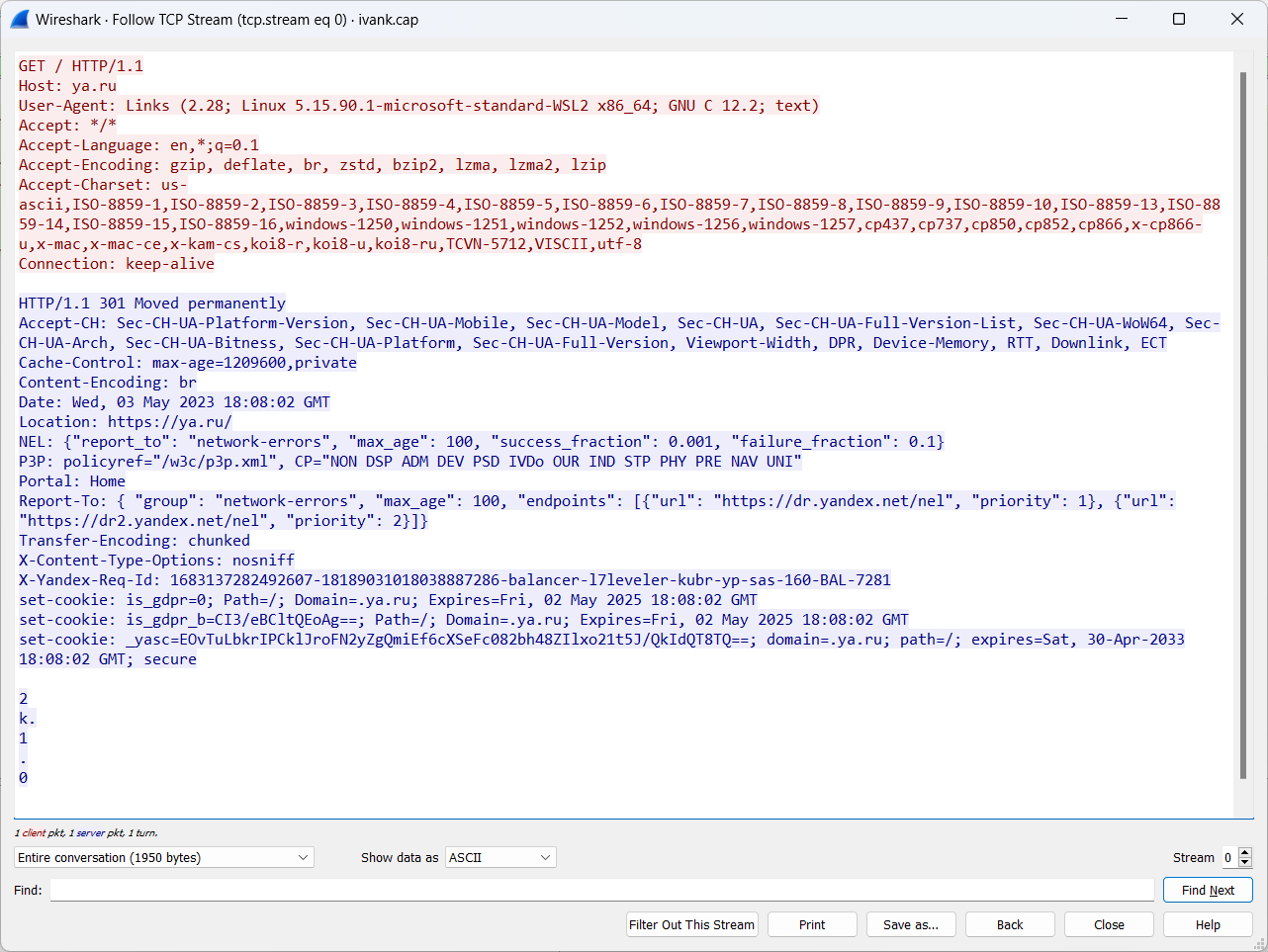
* 1. Находим в списке захваченных пакетов эхо-запросы. Длина поля данных 48 байт. Содержимое поля (кроме заголовков) – перечисление спецсимволов и цифр по порядку следования таблицы *ASCII*.



**Рисунок 16**. Содержимое поля данных

В утилите *Windows* *ping* длина 32 байта и содержимое это латинские символы от *a* до *z* по кругу до полного заполнения.

* 1. Сравниваем *IP* заголовки. Значение *TTL* от *Windows* – 128, *Linux* – 64.
  2. Исследуем трафик утилиты *traceroute*. Утилита отправляет зондирующий пакет *UDP* (по 3 одинаковых пакета). Промежуточные пакеты-ответы это *Time*-*to*-*live* *exceeded*, что служит индикатором продолжать повышать *TTL* и искать путь. Пакет *Destination* *unreachable* (*Port* *unreachable*) служит указателем окончания трассировки.
  3. Находим и восстанавливаем *TCP* поток, соответствующий запросу страницы [*http*://*ya*.*ru*](http://ya.ru).



**Рисунок 17**. Восстановленный TCP поток для запроса страницы ya.ru

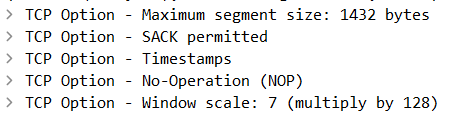
Код ответа 301 *Moved* *permanently*, который пришел от сервера, означает, что запрашиваемый ресурс был перемещен по другому адресу, после чего идет перенаправление. Поэтому в ответе только заголовки, сервер не может найти запрашиваемый ресурс и перенаправляет на актуальный ресурс.

* 1. Не убирая фильтра отображения, найдем в потоке первый пакет, с помощью которого клиент инициирует соединение. Значения:

*Windows* *size* *value* (есть только *Window*): 64440

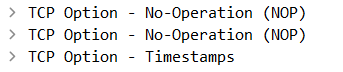
*Calculated* *window* *size*: 64440.

Опции, установленные *TCP* драйвером:



**Рисунок 18**. Опции TCP драйвера

Находим второй пакет, отправленный клиентом. По сравнению с первым пакетом изменились значения *Window* (стало 540), *Calculated* *window* *size* (стало 64512), а также опции *TCP* драйвера:

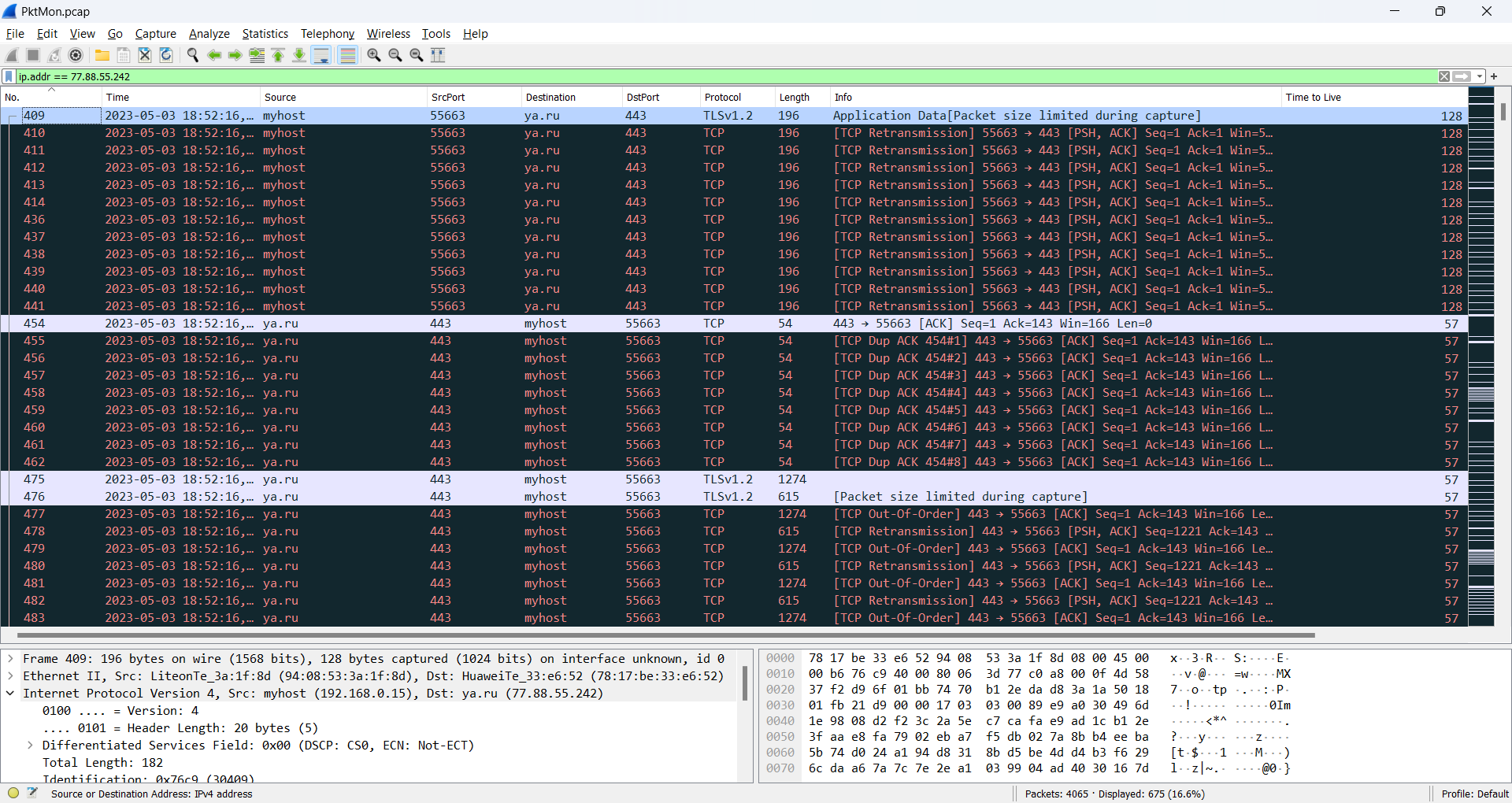


**Рисунок 19**. Опции TCP драйвера во втором пакете

Опций стало меньше, изменились типы опций.

* 1. В *Windows* запускаем командную строку от имени администратора. Запускаем захват пакетов утилитой *pktmon*.

В браузере обращаемся к *ya*.*ru* и останавливаем захват трафика. Преобразуем полученный файл в .*cap*. Открываем полученный файл в новом окне *Wireshark*.



**Рисунок 20**. Захваченные пакеты при обращении к ya.ru

* 1. Для первого *TCP* соединения (все красно-черные одинаковые по значениям и опциям) с сайтом *ya*.*ru* выписываем значения:

*Window*: 1027

*Calculated* *window* *size*: 1027

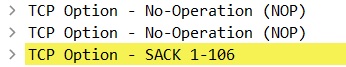
*TCP* опции: отсутствуют.

Найдем второй пакет:

*Window*: 1026

*Calculated* *window* *size*: 1026

*TCP* опции:



**Рисунок 21**. TCP опции второго пакета от клиента

Размеры окон почти не изменились, добавились *TCP* опции во втором пакете.

* 1. Сравним поведение *TCP* драйвера и трассировщика в *Linux* и *Windows*:

Опции *TCP* драйвера отличаются. Трассировщик использует разные зондирующие пакеты (*Linux* *UDP*, *Windows* *ICMP*).

Можно определить ОС по размерам окна (значения для *Linux* и *Windows* сильно отличаются друг от друга), по *TTL* (64 *Linux* и 128 *Windows*). Исходя из этих сравнений, сервер может определить, какая ОС отправляет ему запрос.

## Вывод

В ходе выполнения лабораторной работы, были освоены продвинутые навыки работы с сетевым анализатором *Wireshark*.

Была произведена работа по просмотру содержимого *TLS* трафика. Получилось восстановить данные, отправленные *POST* запросом серверу.

Была проведена работа с *HTTP*2 трафиком. Попытались по аналогии с *HTTP* и *TLS* восстановить поток и данные, безрезультатно, в этом заслуга *HTTP*2.

В работе использовали базу геолокаций *GeoLite*2 для определения геолокаций узлов, а именно стран, городов, владельцев *AS*. Создана карта местоположений посещенных сайтов.

Рассмотрены и практически закреплены навыки работы с альтернативными источниками трафика, такими, как *VoIP*, дампы трафика на разных ОС (*Windows*, *Linux*).

Проведен анализ пакетов при эхо-тестировании, трассировке и посещении сайта с разных ОС. Сделаны выводы по вопросу возможности определения сервером ОС, которая присылает ему пакеты.