МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт кибербезопасности и цифровых технологий (ИКБ)

КБ-2 «Информационно-аналитические системы кибербезопасности»

ОТЧЕТ О ВЫПОЛНЕНИИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ №2

В РАМКАХ ДИСЦИПЛИНЫ «ТЕХНОЛОГИИ ХРАНЕНИЯ В СИСТЕМАХ КИБЕРБЕЗОПАСНОСТИ»

Выполнил:

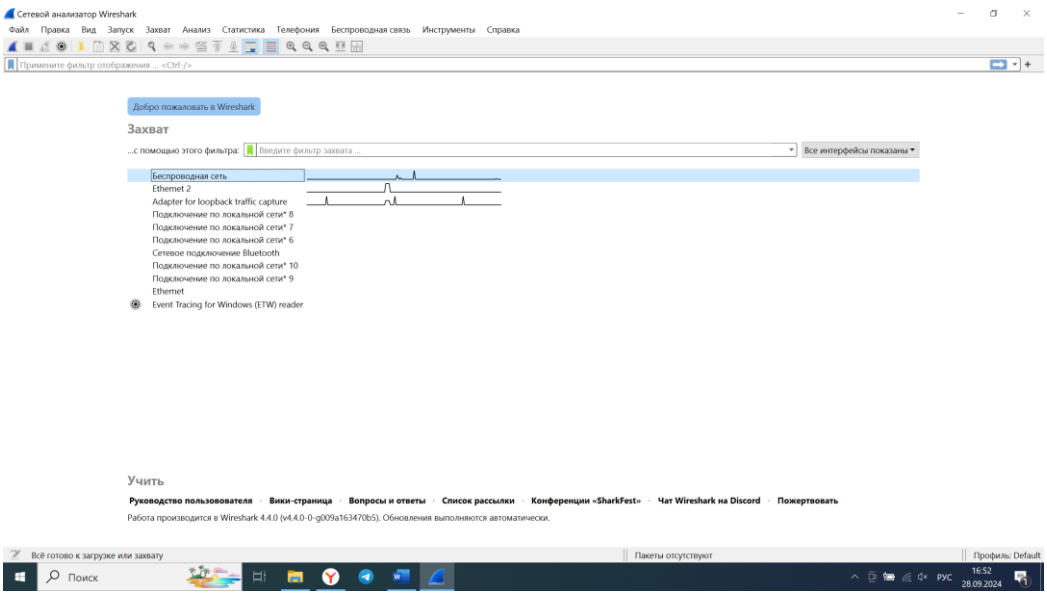
Студент 3-ого курса

Учебной группы БИСО-02-22

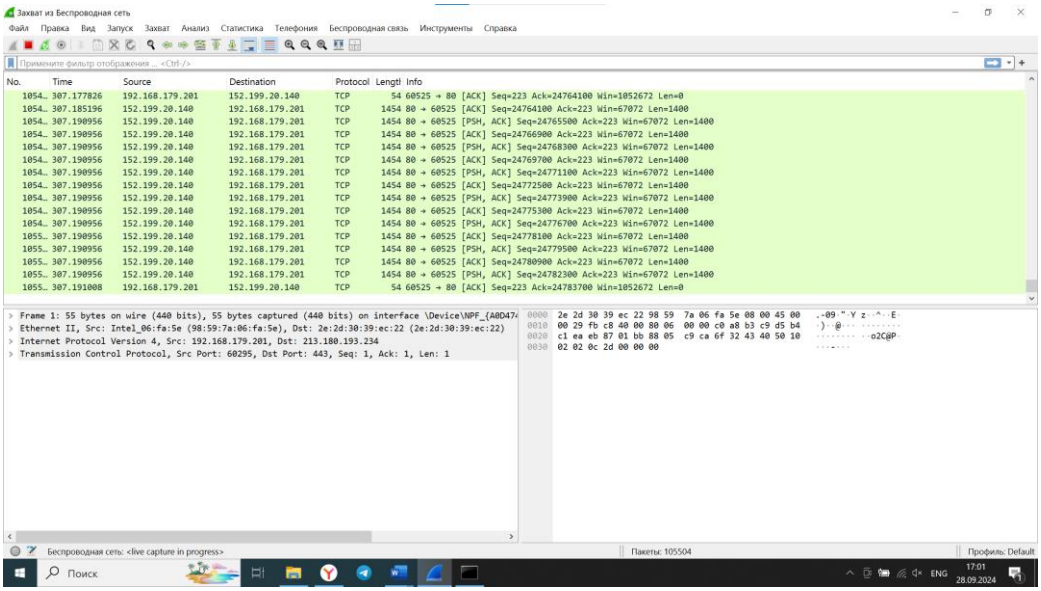
Зубарев В.С.

Москва 2024

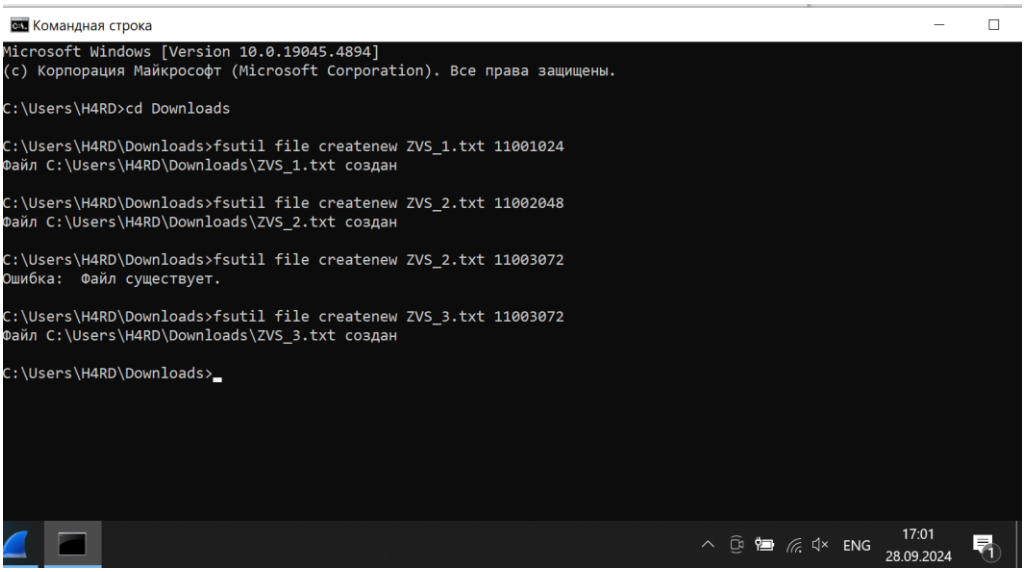
1. Установите и запустите ПО «Wireshark»



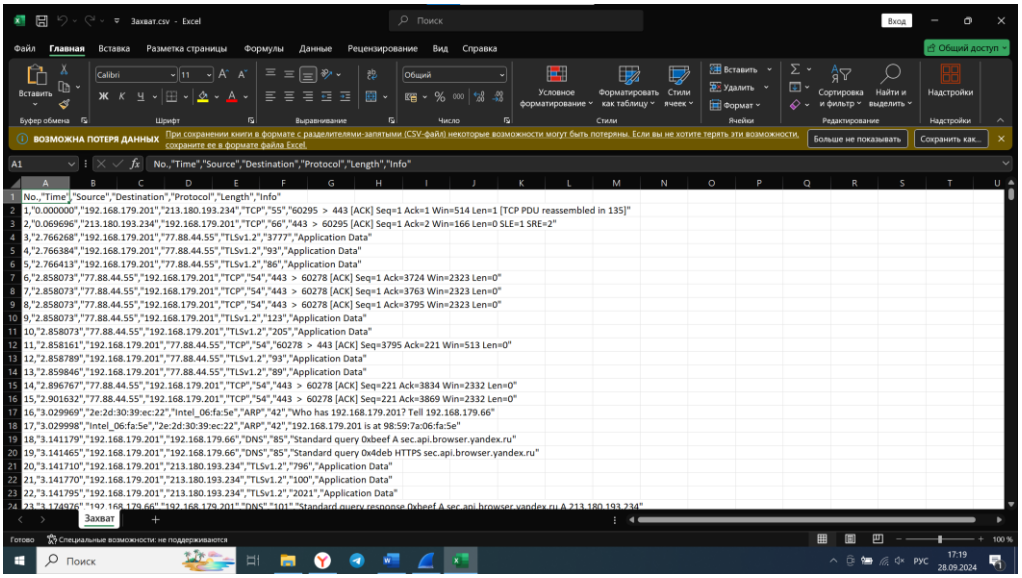
2. Запустите захват пакетов с интерфейса, который используется для выхода в сеть Интернет.



3. Создайте несколько файлов (минимум 3) размером 11000000 + (номер по списку * 1 КБ + N) байт, где N – номер файла. Имя файлов должно иметь следующий вид: FIO_N.txt, где F – первая буква фамилии на латинице; I – первая буква имени на латинице; O – первая буква отчества на латинице (при наличии), N – номер файла (1, 2, 3, …).



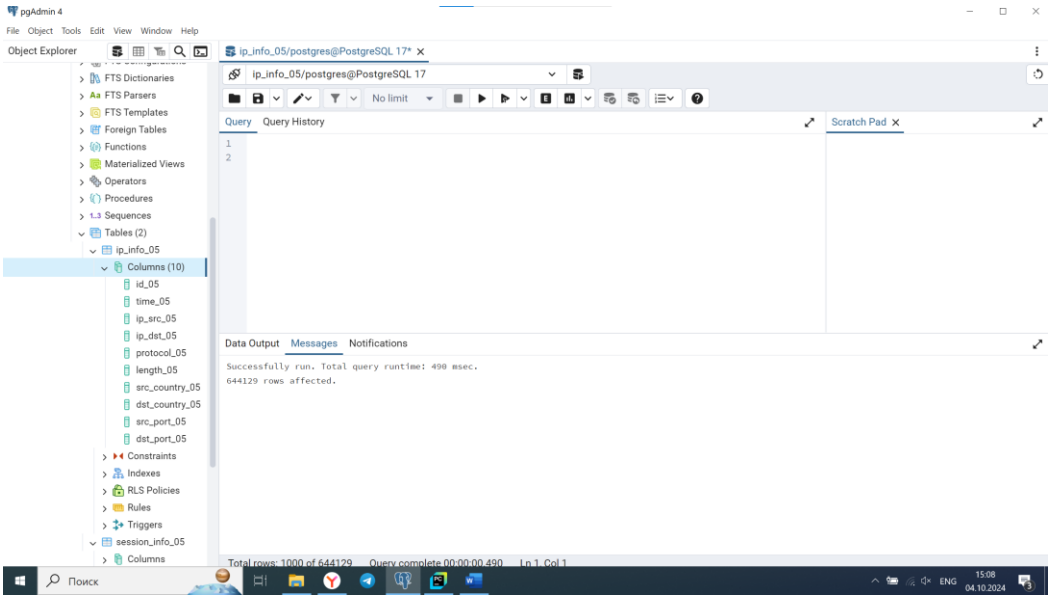
4. Остановите захват пакетов и экспортируйте информацию о сетевых пакетах в формате CSV и JSON. Проанализируйте структуру файлов.



5. Создайте базу данных в PostgreSQL с именем ip_info_XX (XX – порядковый номер студента в группе).

   Создайте таблицу для хранения информации об IP-соединениях. В Приложении 2 приведен пример создания таблицы в pgAdmin. Минимальный набор полей: IPадрес отправителя, IP-адрес получателя, страна адреса отправителя, страна адреса получателя,

порт отправителя, порт получателя, протокол, размер пакета (сессии). К названию каждого поля добавляем «_XX» (XX – порядковый номер студента в группе). Например: ip_src_07, ip_dst_07, country_from_07, country_to_07, length_07 и т.д



6. Напишите программу/скрипт для парсинга информации об IP-соединениях из ранее сохраненного файла CSV или JSON. Определитесь, когда будете вычислять информацию о размере сессии: перед загрузкой в БД или после. Для вычисления размера сессии необходимо просуммировать размеры пакетов, относящиеся к одной сессии. В программе/скрипте необходимо реализовать обогащение IPадресов информацией (как минимум о стране).

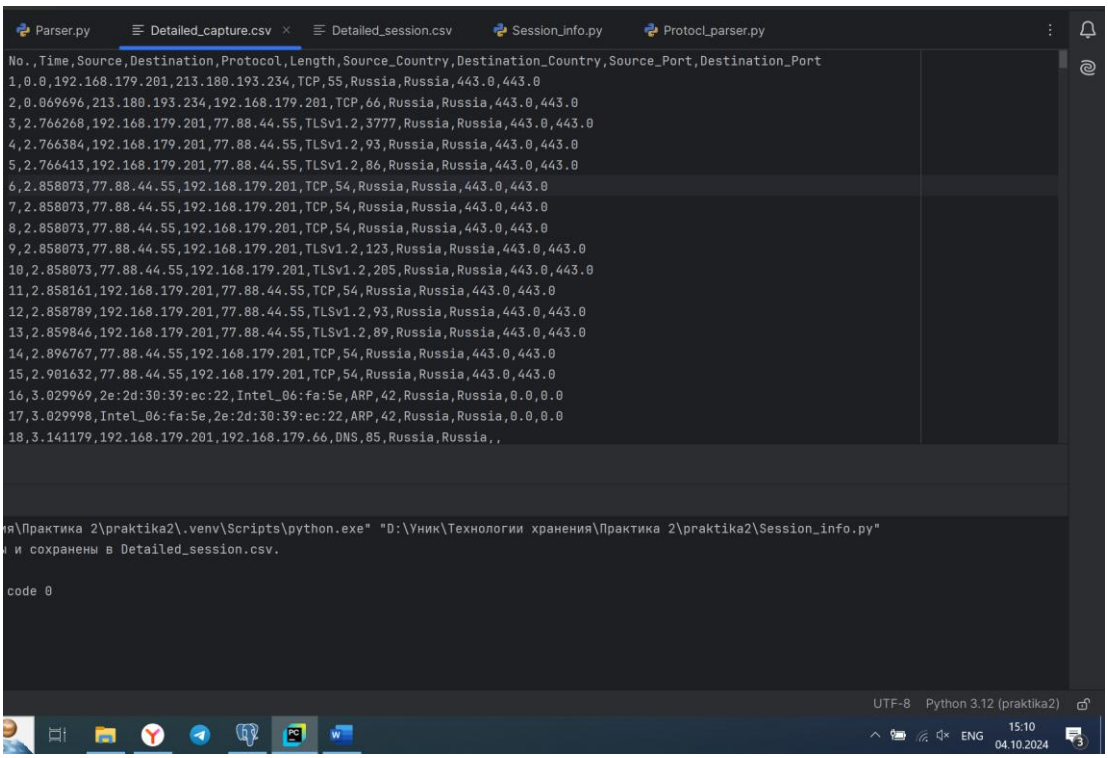
Для реализации обогащения ip адресов был написана программа на языке python.

Листинг программы:

```
import geoip2.database
import pandas as pd

# Путь к базе данных GeoLite2
geoip_db_path = 'GeoLite2-Country.mmdb'

# Путь к входному CSV файлу с данными Wireshark
```

```
input_csv_path = 'Capture.csv'

# Путь к выходному CSV файлу с добавленной информацией о стране и портах
output_csv_path = 'Detailed_capture.csv'


# Функция для получения информации о стране по IP адресу
def get_country_by_ip(ip_address, reader):
    try:
        response = reader.country(ip_address)
        return response.country.name
    except Exception:
        return "Russia"


# Функция для получения порта по протоколу
def get_ports_by_protocol(protocol):
    protocol_ports = {
        'TCP': (443,443),  # HTTPS (443)
        'ARP':(0,0), #Протокол канального уровня
        'TLSv1.2': (443, 443),  # TLS работает на 443 порту
        'SSDP': (1900,1900), #1900(UDP)
        'TLSv1.3': (443, 443),  # TLS работает на 443 порту
        'MDNS': (5353,5353), #UDP
        'QUIC': (443,443), #UDP
        'SSLv2': (443,443), #443(HTTPS)
        'UDP': (53, 53),  # DNS обычно использует UDP на порту 53
    }
    return protocol_ports.get(protocol, (None, None))  # Если протокол
неизвестен, вернем None


# Открываем базу данных GeoLite2
with geoip2.database.Reader(geoip_db_path) as reader:
    # Читаем данные из CSV файла
    df = pd.read_csv(input_csv_path)

    # Добавляем информацию о стране для Source и Destination IP-адресов
    df['Source_Country'] = df['Source'].apply(lambda x: get_country_by_ip(x,
reader))
    df['Destination_Country'] = df['Destination'].apply(lambda x:
get_country_by_ip(x, reader))

    # Добавляем информацию о портах, исходя из протокола
    df[['Source_Port', 'Destination_Port']] = df['Protocol'].apply(lambda x:
pd.Series(get_ports_by_protocol(x)))

# Сохраняем результат в новый CSV файл
df.to_csv(output_csv_path, index=False)

print(f"Данные с информацией о стране и портах успешно сохранены в
{output_csv_path}")
```

Результат работы (чтение полученного файла):



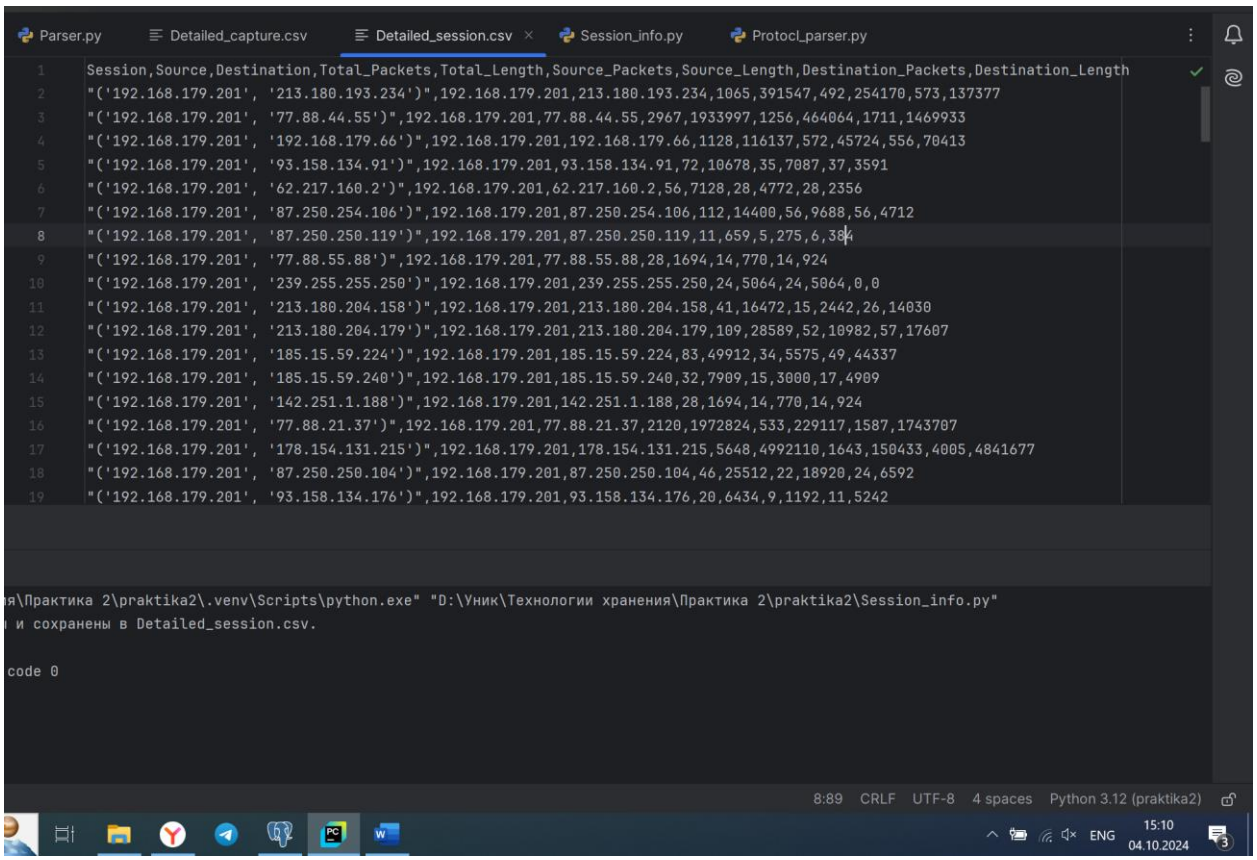
Для подсчета размера сессии была написана вторая программа на языке python

Листинг программы:

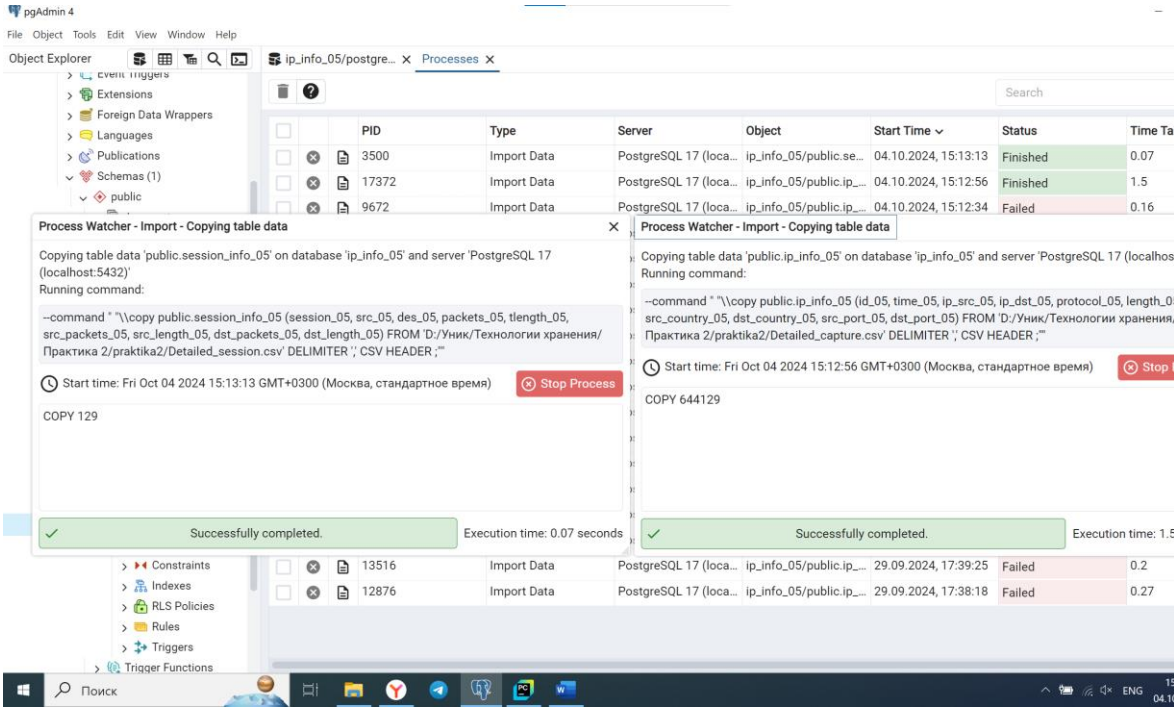
```
import pandas as pd

# IP-адрес хоста, который всегда должен быть Source
host_ip = "192.168.179.201"

# Чтение исходного CSV файла
input_csv_path = 'Detailed_capture.csv'  # Укажите путь к вашему файлу
df = pd.read_csv(input_csv_path)

# Создаем словарь для хранения информации о сессиях
sessions = {}

# Проходим по каждой строке исходного файла
for index, row in df.iterrows():
    src = row['Source']
    dst = row['Destination']
    length = int(row['Length'])
    # Определяем, кто из них является хостом (192.168.179.201)
    if src == host_ip:
        # Хост отправляет данные, сервер принимает
        session_key = (src, dst)
        source_packets = 1
        source_length = length
        destination_packets = 0
        destination_length = 0
    elif dst == host_ip:
        # Сервер отправляет данные, хост принимает
        session_key = (dst, src)
        source_packets = 0
```

```
        source_length = 0
        destination_packets = 1
        destination_length = length
    else:
        # Если ни один из IP не является хостом, пропускаем строку
        continue

    # Обновляем информацию о сессии
    if session_key not in sessions:
        sessions[session_key] = {
            'Source_Packets': source_packets,
            'Source_Length': source_length,
            'Destination_Packets': destination_packets,
            'Destination_Length': destination_length,
        }
    else:
        sessions[session_key]['Source_Packets'] += source_packets
        sessions[session_key]['Source_Length'] += source_length
        sessions[session_key]['Destination_Packets'] += destination_packets
        sessions[session_key]['Destination_Length'] += destination_length

# Формируем данные для записи в CSV
session_data = []
for session_key, data in sessions.items():
    session_data.append({
        'Session': f"{session_key}",
        'Source': session_key[0],  # Хост (192.168.179.201)
        'Destination': session_key[1],  # Сервер
        'Total_Packets': data['Source_Packets'] +
data['Destination_Packets'],
        'Total_Length': data['Source_Length'] + data['Destination_Length'],
        'Source_Packets': data['Source_Packets'],
        'Source_Length': data['Source_Length'],
        'Destination_Packets': data['Destination_Packets'],
        'Destination_Length': data['Destination_Length']
    })

# Создаем DataFrame с результатами
output_df = pd.DataFrame(session_data)

# Сохраняем результат в новый CSV файл
output_csv_path = 'Detailed_session.csv'  # Укажите путь для сохранения файла
output_df.to_csv(output_csv_path, index=False)

print(f"Сессии успешно сформированы и сохранены в {output_csv_path}.")
```

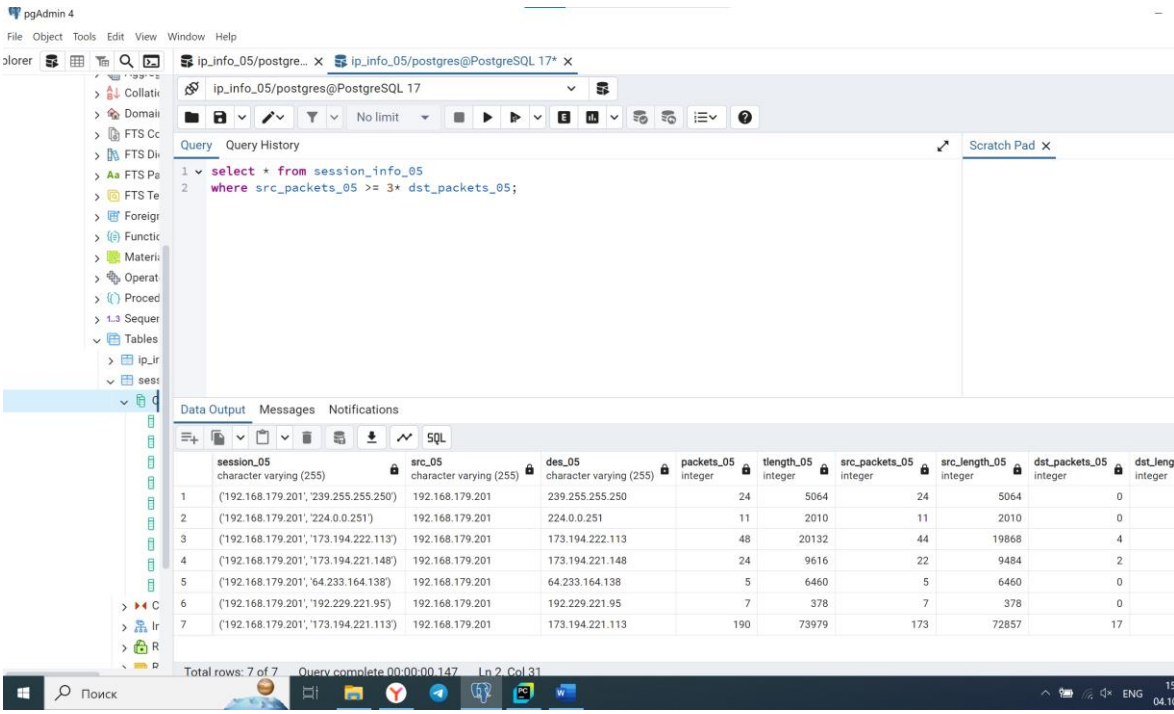
Результат работы программы (чтение файла):

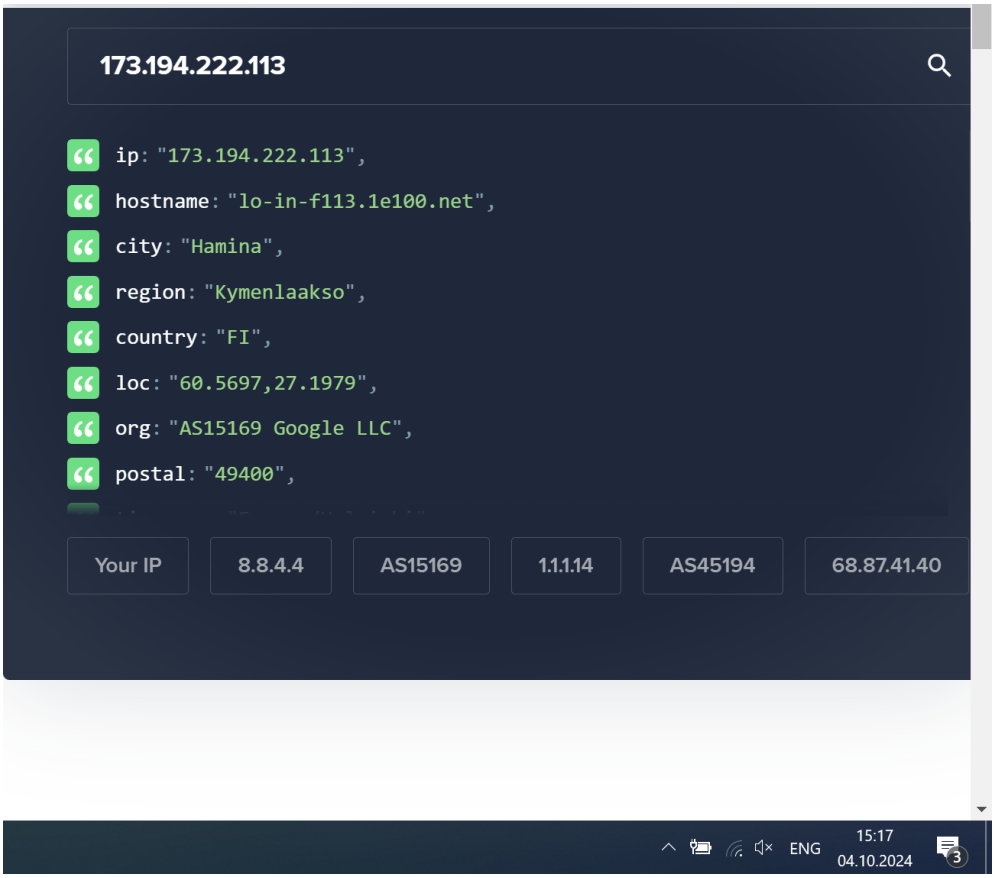


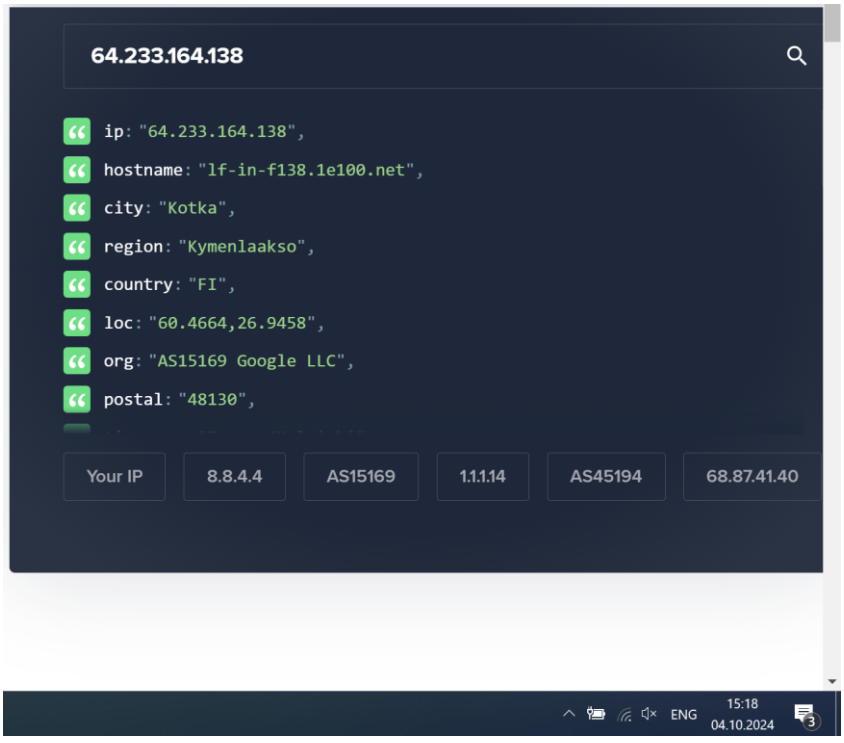
7. Загрузите данные в БД.



8. Напишите программу/скрипт для вычисления потенциальных утечек информации на основе собранных данных (объем данных, передающихся на сервер, во много раз больше объема данных, полученных от сервера). Соберите информацию о выявленных IP-адресах серверов (например, с сайта https://ipinfo.io).



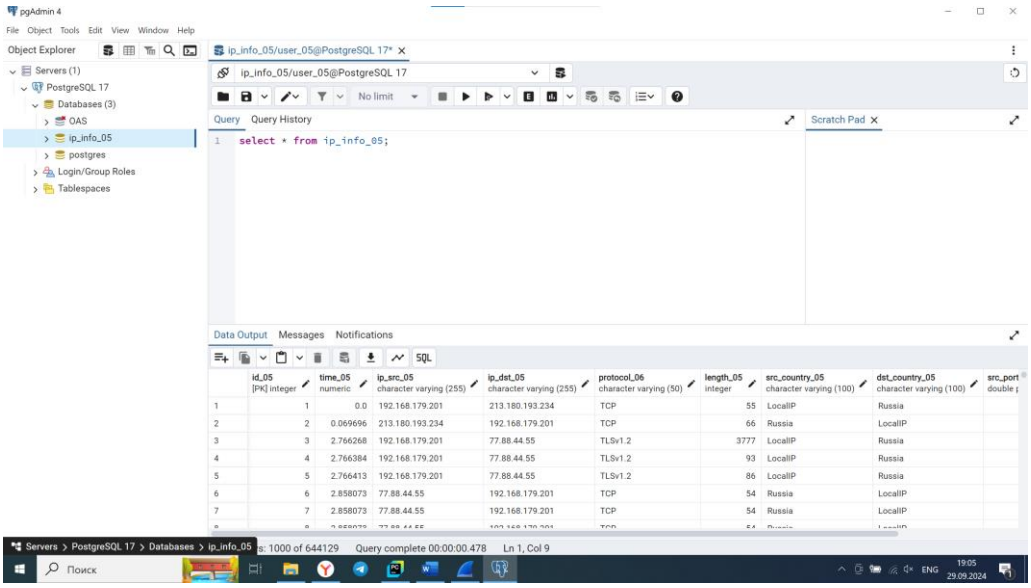




9. Создайте пользователя user_XX (XX – порядковый номер студента в группе), имеющего права только на чтение данных.

Авторизуйтесь под созданным пользователем. Выполните запросы на чтение и удаление/изменение данных.

Чтение данных



Изменение данных