



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:

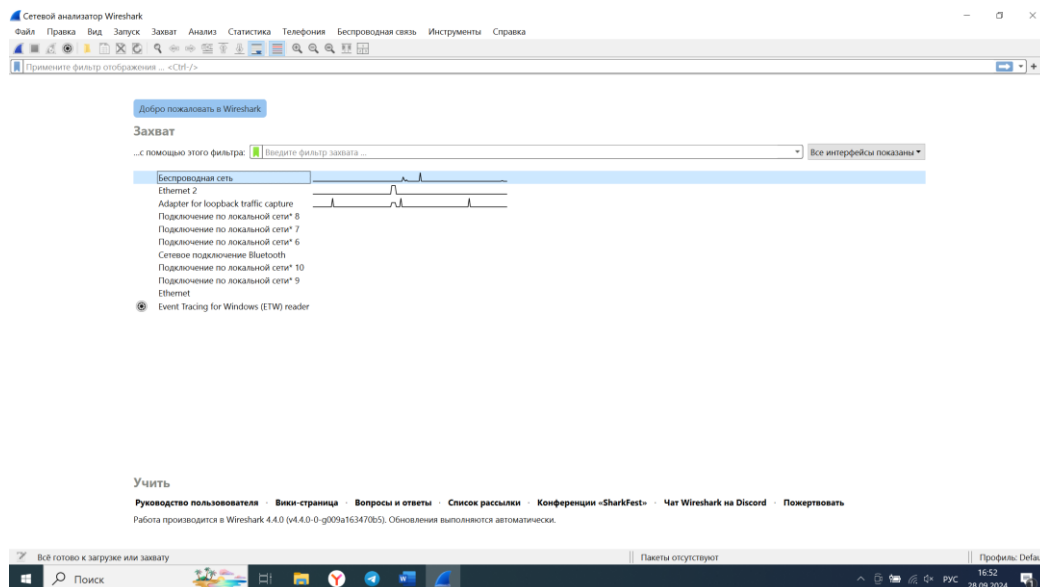
Студент 3-ого курса

Учебной группы БИСО-02-22

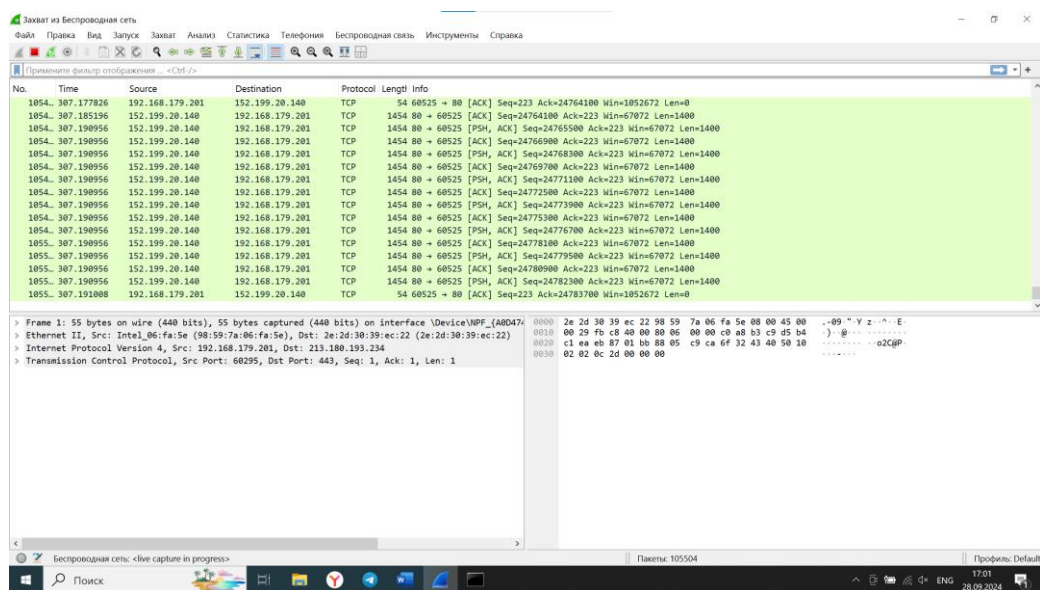
Зубарев В.С.

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 $11000000 + (\text{номер по списку} * 1 \text{ КБ} + N)$ 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, ...).

```
Командная строка
Microsoft Windows [Version 10.0.19045.4894]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\H4RD>cd Downloads

C:\Users\H4RD\Downloads>fsutil file createnew ZVS_1.txt 11001024
Файл C:\Users\H4RD\Downloads\ZVS_1.txt создан

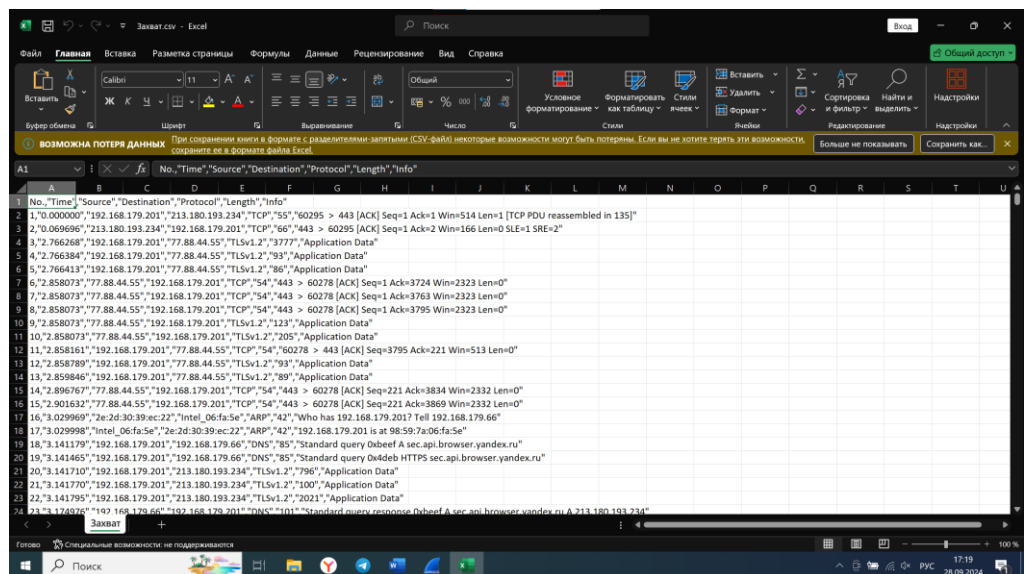
C:\Users\H4RD\Downloads>fsutil file createnew ZVS_2.txt 11002048
Файл C:\Users\H4RD\Downloads\ZVS_2.txt создан

C:\Users\H4RD\Downloads>fsutil file createnew ZVS_2.txt 11003072
Ошибка: Файл существует.

C:\Users\H4RD\Downloads>fsutil file createnew ZVS_3.txt 11003072
Файл C:\Users\H4RD\Downloads\ZVS_3.txt создан

C:\Users\H4RD\Downloads>
```

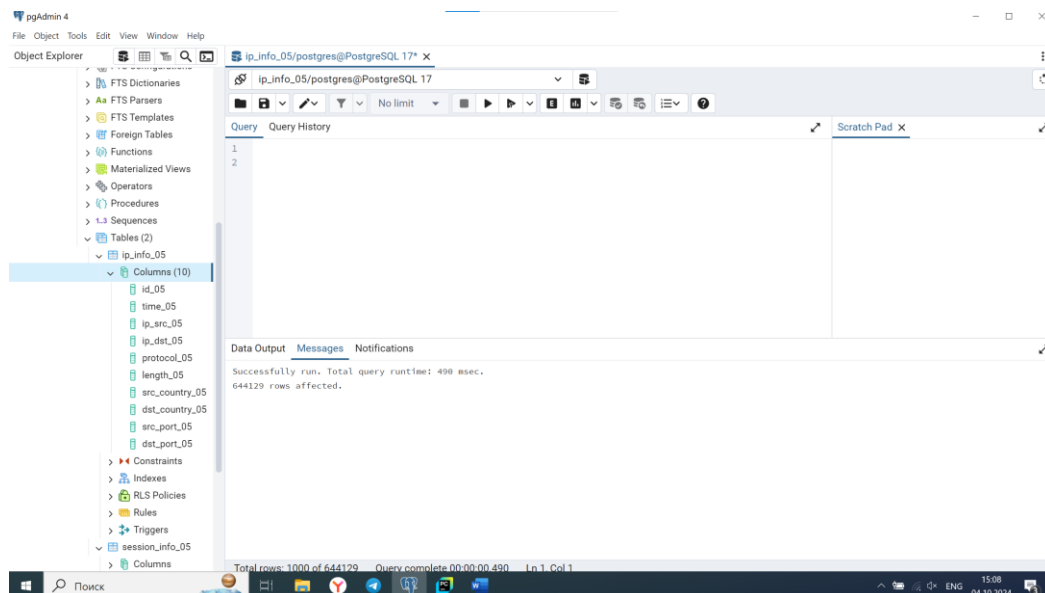
- Остановите захват пакетов и экспортируйте информацию о сетевых пакетах в формате CSV и JSON. Проанализируйте структуру файлов.



- 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: IP-адрес 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 IP-адресов информацией (как минимум о стране).

Для реализации обогащения IP-адресов была 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):

```
Parser.py x Detailed_capture.csv Detailed_session.csv Session_info.py Protocol-parser.py
No.,Time,Source,Destination,Protocol,Length,Source_Country,Destination_Country,Source_Port,Destination_Port
1,0.0,192.168.179.201,213.180.193.234,TCP,55,Russia,Russia,443.0,443.0
2,0.069696,213.180.193.234,192.168.179.201,TCP,66,Russia,Russia,443.0,443.0
3,2.766268,192.168.179.201,77.88.44.55,TLSv1.2,3777,Russia,Russia,443.0,443.0
4,2.766384,192.168.179.201,77.88.44.55,TLSv1.2,93,Russia,Russia,443.0,443.0
5,2.766413,192.168.179.201,77.88.44.55,TLSv1.2,86,Russia,Russia,443.0,443.0
6,2.858073,77.88.44.55,192.168.179.201,TCP,54,Russia,Russia,443.0,443.0
7,2.858073,77.88.44.55,192.168.179.201,TCP,54,Russia,Russia,443.0,443.0
8,2.858073,77.88.44.55,192.168.179.201,TCP,54,Russia,Russia,443.0,443.0
9,2.858073,77.88.44.55,192.168.179.201,TLSv1.2,123,Russia,Russia,443.0,443.0
10,2.858073,77.88.44.55,192.168.179.201,TLSv1.2,205,Russia,Russia,443.0,443.0
11,2.858161,192.168.179.201,77.88.44.55,TCP,54,Russia,Russia,443.0,443.0
12,2.858789,192.168.179.201,77.88.44.55,TLSv1.2,93,Russia,Russia,443.0,443.0
13,2.859846,192.168.179.201,77.88.44.55,TLSv1.2,89,Russia,Russia,443.0,443.0
14,2.896767,77.88.44.55,192.168.179.201,TCP,54,Russia,Russia,443.0,443.0
15,2.901632,77.88.44.55,192.168.179.201,TCP,54,Russia,Russia,443.0,443.0
16,3.029969,2e:2d:30:39:ec:22,Intel_06:fa:5e:42,Russia,Russia,0.0,0.0
17,3.029998,Intel_06:fa:5e:2e:2d:30:39:ec:22,ARP,42,Russia,Russia,0.0,0.0
18,3.141179,192.168.179.201,192.168.179.66,DNS,85,Russia,Russia,,
D:\Практика 2\praktika2\.venv\Scripts\python.exe "D:\Уник\Технологии хранения\Практика 2\praktika2\Session_info.py"
и сохранены в Detailed_session.csv.
code 0
UTF-8 Python 3.12 (praktika2) 15:10 04.10.2024
```

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):

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>).

The screenshot shows the pgAdmin 4 interface. The query editor contains the following SQL query:

```
1 select * from session_info_05
2 where src_packets_05 >= 3* dst_packets_05;
```

The query results are displayed in the Data Output tab, showing 7 rows of data. The columns are: session_05, src_05, des_05, packets_05, tlength_05, src_packets_05, src_length_05, dst_packets_05, and dst_length_05.

	session_05	src_05	des_05	packets_05	tlength_05	src_packets_05	src_length_05	dst_packets_05	dst_length_05
1	('192.168.179.201','239.255.255.250')	192.168.179.201	239.255.255.250	24	5064	24	5064	0	
2	('192.168.179.201','224.0.0.251')	192.168.179.201	224.0.0.251	11	2010	11	2010	0	
3	('192.168.179.201','173.194.222.113')	192.168.179.201	173.194.222.113	48	20132	44	19868	4	
4	('192.168.179.201','173.194.221.148')	192.168.179.201	173.194.221.148	24	9616	22	9484	2	
5	('192.168.179.201','64.233.164.138')	192.168.179.201	64.233.164.138	5	6460	5	6460	0	
6	('192.168.179.201','192.229.221.95')	192.168.179.201	192.229.221.95	7	378	7	378	0	
7	('192.168.179.201','173.194.221.113')	192.168.179.201	173.194.221.113	190	73979	173	72857	17	

Total rows: 7 of 7. Query complete 00:00:00.147. Ln 2, Col 31.

173.194.222.113

“ ip: "173.194.222.113",

“ hostname: "lo-in-f113.1e100.net",

“ city: "Hamina",

“ region: "Kymenlaakso",

“ country: "FI",

“ loc: "60.5697,27.1979",

“ org: "AS15169 Google LLC",

“ postal: "49400",

Your IP

8.8.4.4

AS15169

11.114

AS45194

68.87.41.40

15:17

04.10.2024

3

64.233.164.138

“ ip: "64.233.164.138",

“ hostname: "lf-in-f138.1e100.net",

“ city: "Kotka",

“ region: "Kymenlaakso",

“ country: "FI",

“ loc: "60.4664,26.9458",

“ org: "AS15169 Google LLC",

“ postal: "48130",

Your IP

8.8.4.4

AS15169

11.114

AS45194

68.87.41.40

15:18

04.10.2024

3

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

The screenshot shows the pgAdmin 4 interface. The left pane displays the 'Object Explorer' with the following structure:

- Servers (1)
 - PostgreSQL 17
 - Databases (3)
 - OAS
 - ip_info_05
 - postgres
 - Login/Group Roles
 - Tablespaces

The main pane shows a SQL query in the 'Query' tab:

```
1 select * from ip_info_05;
```

The 'Data Output' pane displays the results of the query in a table with 10 columns:

	id_05 [PK] integer	time_05 numeric	ip_src_05 character varying (255)	ip_dst_05 character varying (255)	protocol_06 character varying (50)	length_05 integer	src_country_05 character varying (100)	dst_country_05 character varying (100)	src_port double precision
1	1	0.0	192.168.179.201	213.180.193.234	TCP	55	LocalIP	Russia	
2	2	0.069696	213.180.193.234	192.168.179.201	TCP	66	Russia	LocalIP	
3	3	2.766268	192.168.179.201	77.88.44.55	TLShv1.2	3777	LocalIP	Russia	
4	4	2.766384	192.168.179.201	77.88.44.55	TLShv1.2	93	LocalIP	Russia	
5	5	2.766413	192.168.179.201	77.88.44.55	TLShv1.2	86	LocalIP	Russia	
6	6	2.858073	77.88.44.55	192.168.179.201	TCP	54	Russia	LocalIP	
7	7	2.858073	77.88.44.55	192.168.179.201	TCP	54	Russia	LocalIP	

The status bar at the bottom indicates: 'Total rows: 1000 of 644129 Query complete 00:00:00.478 Ln 1, Col 9'.

Изменение данных

The screenshot shows the pgAdmin 4 interface. The left pane displays the 'Object Explorer' with the same structure as the previous screenshot.

The main pane shows an update query in the 'Query' tab:

```
1 update ip_info_05
2 set protocol_05 = 'TASK'
3 where id_05 = 2
```

The 'Messages' pane displays an error message:

```
ERROR: нет доступа к таблице ip_info_05
ОШИБКА: нет доступа к таблице ip_info_05
SQL state: 42503
```

The status bar at the bottom indicates: 'Total rows: 1000 of 644129 Query complete 00:00:00.080 Ln 2, Col 25'.