

**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 №4**

# **В РАМКАХ ДИСЦИПЛИНЫ «ТЕХНОЛОГИИ ХРАНЕНИЯ В СИСТЕМАХ КИБЕРБЕЗОПАСНОСТИ»**

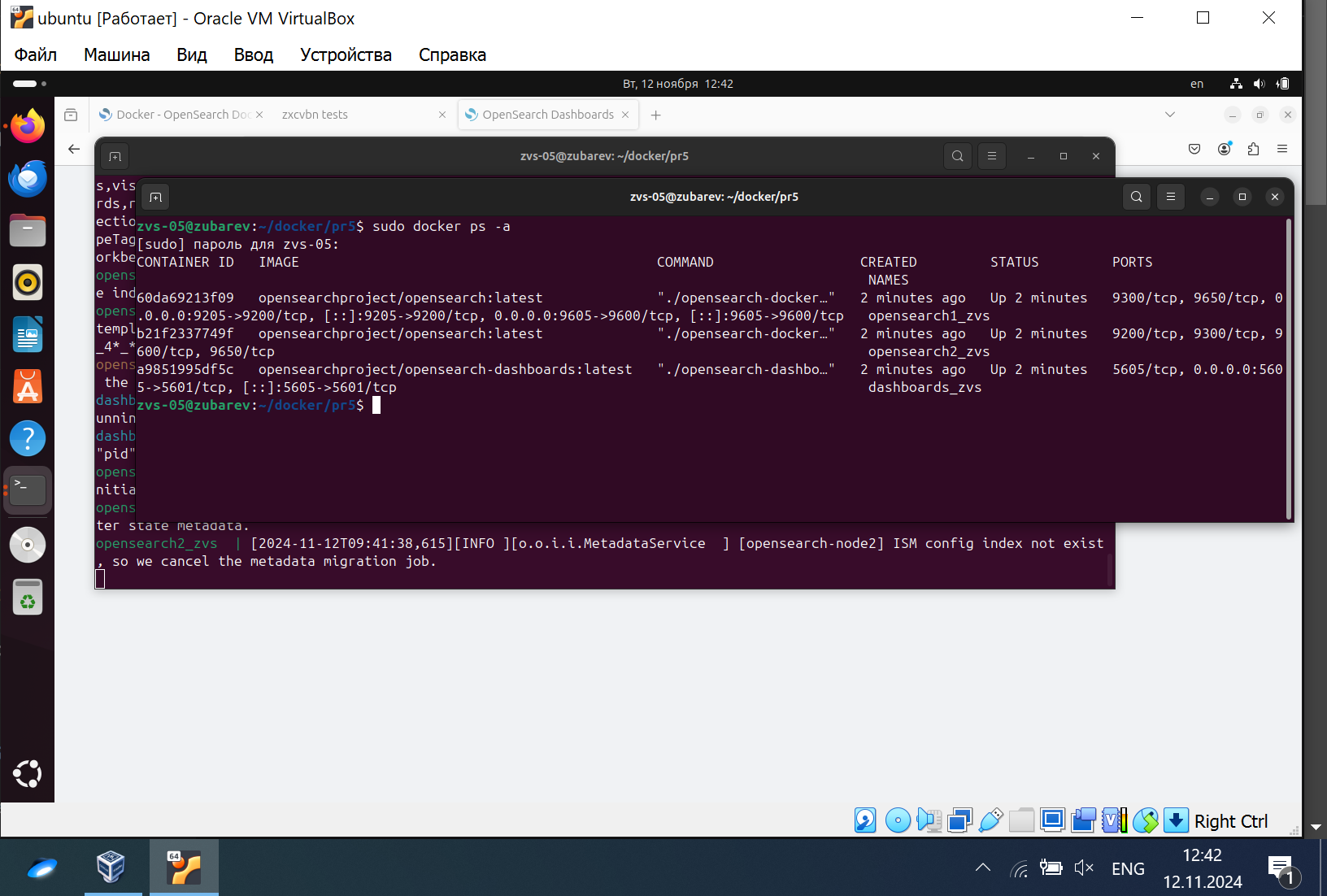
Выполнил:

Студент 3-ого курса

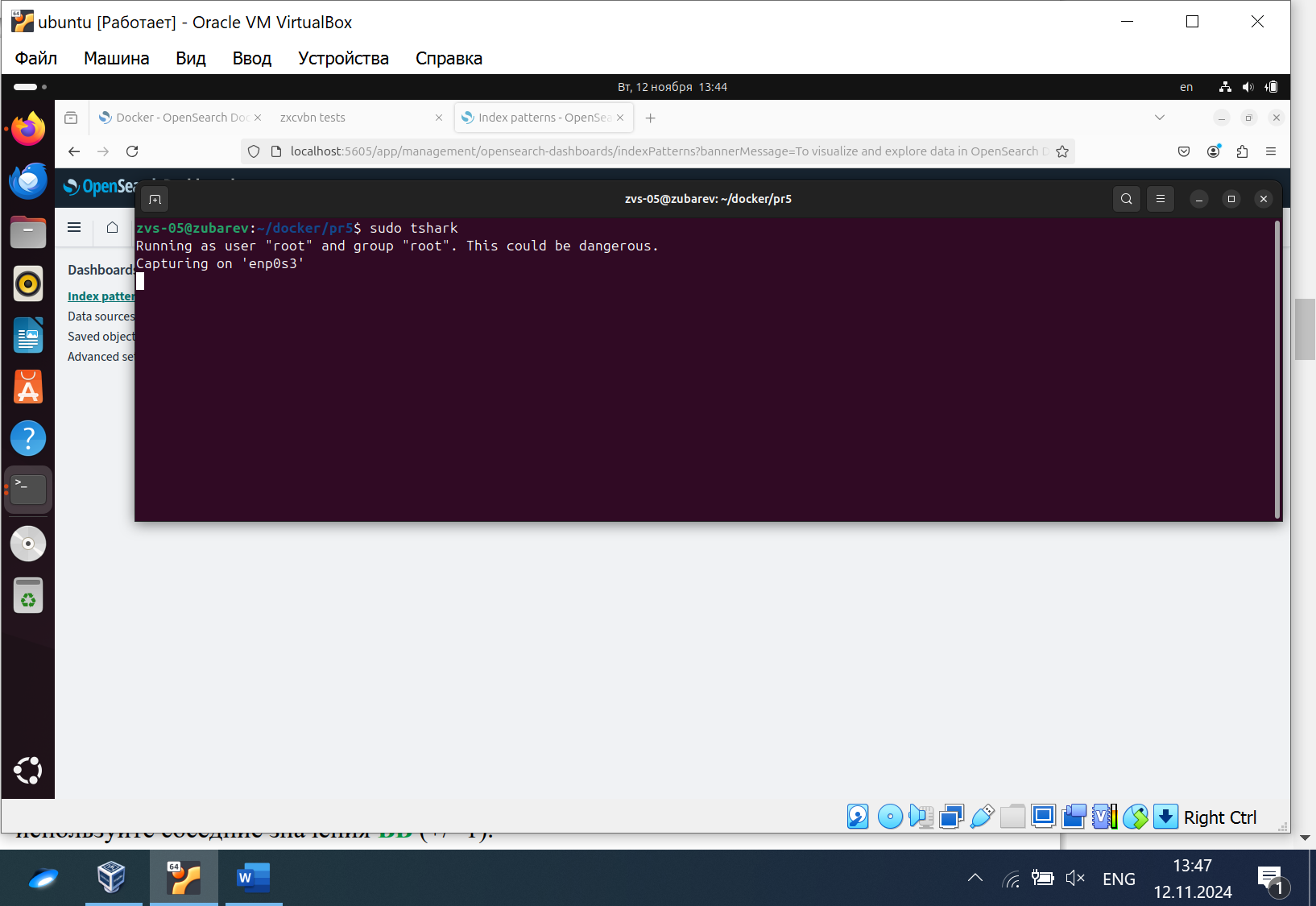
Учебной группы БИСО-02-22

Зубарев В.С.

Разверните OpenSearch и OpenSearch Dashboards с помощью Docker Compose



Установите ПО TShark (инструмент для сетевого анализа пакетов):

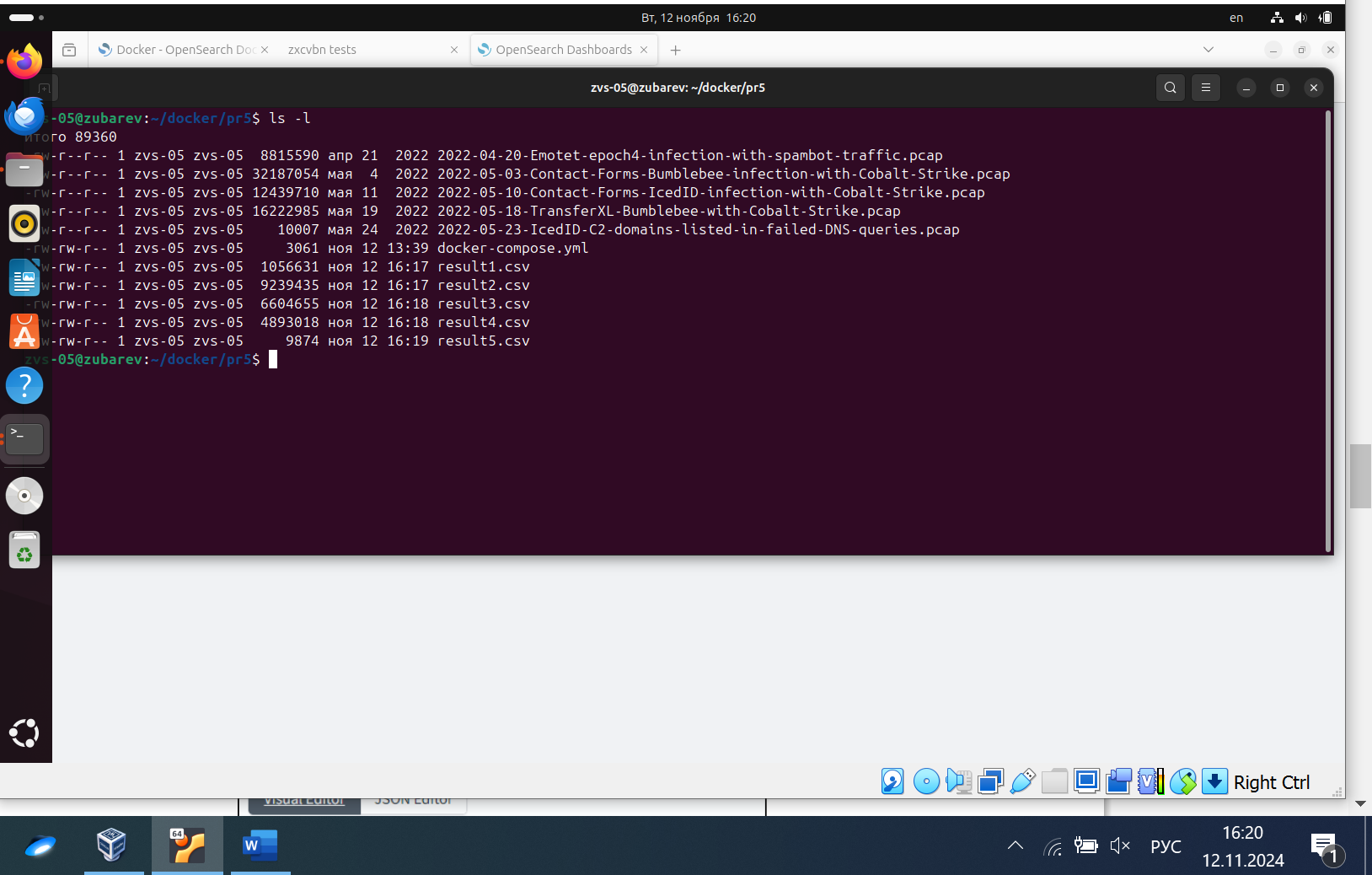


Соберите статистические данные от скачанных PCAP-файлов с помощью TShark.

Я добавил для поля tcp.stream (для связи пакетов в одной tcp-сессии) и tcp.falgs (для идентификации пакетов, что бы можно было понять кое действие совершается)

Команда(для первого файла)

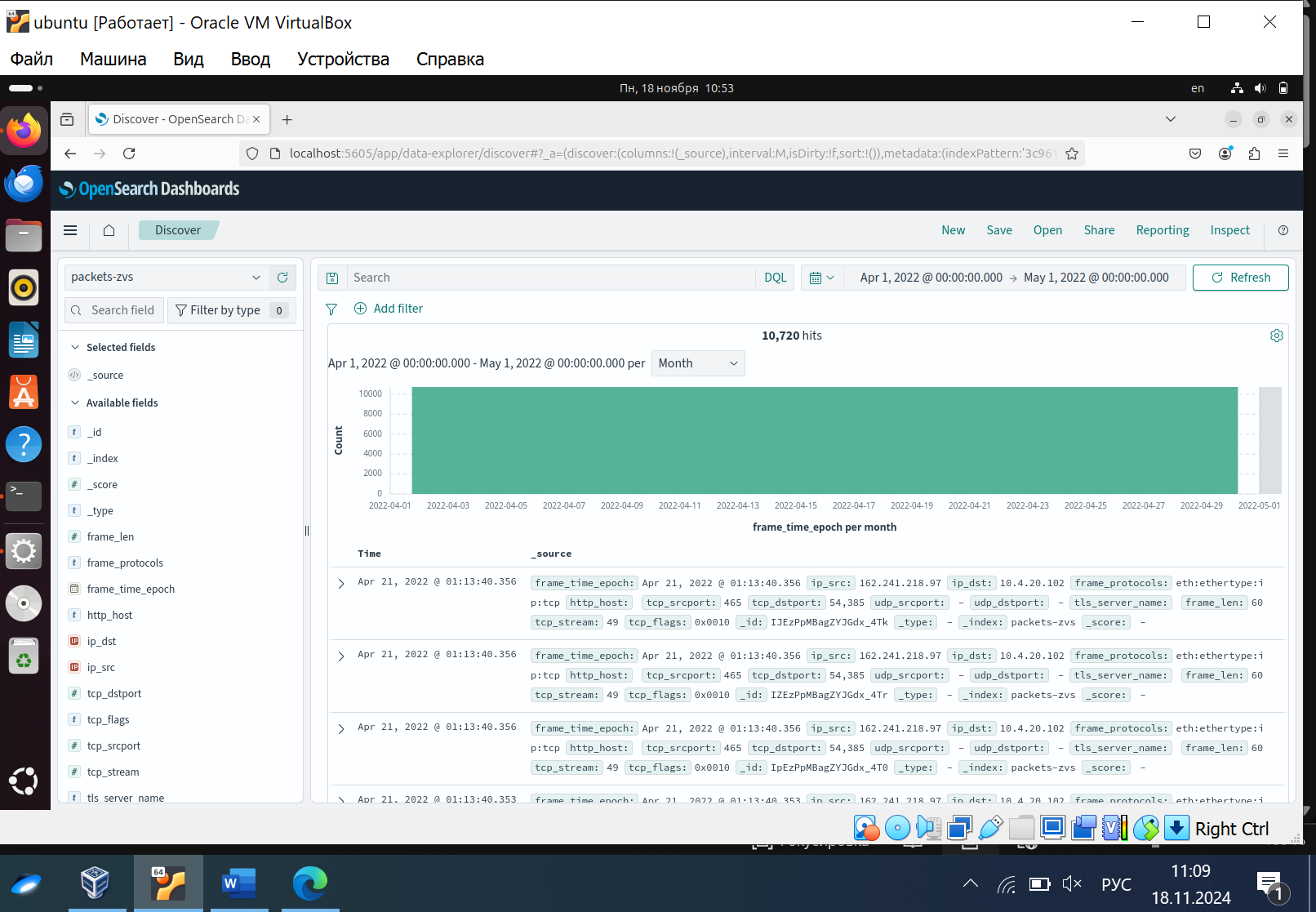
sudo tshark -r 2022-04-20-Emotet-epoch4-infection-with-spambot-traffic.pcap -T fields -e frame.time\_epoch -e ip.src -e ip.dst -e frame.protocols -e http.host -e tcp.srcport -e tcp.dstport -e udp.srcport -e udp.dstport -e tls.handshake.extensions\_server\_name -e frame.len -e tcp.stream -e tcp.flags -E header=y -E separator=\; -E aggregator=, > result1.csv



Загрузите полученные статистические данные в OpenSearch любым удобным способом. Ниже представлен пример на языке Python.

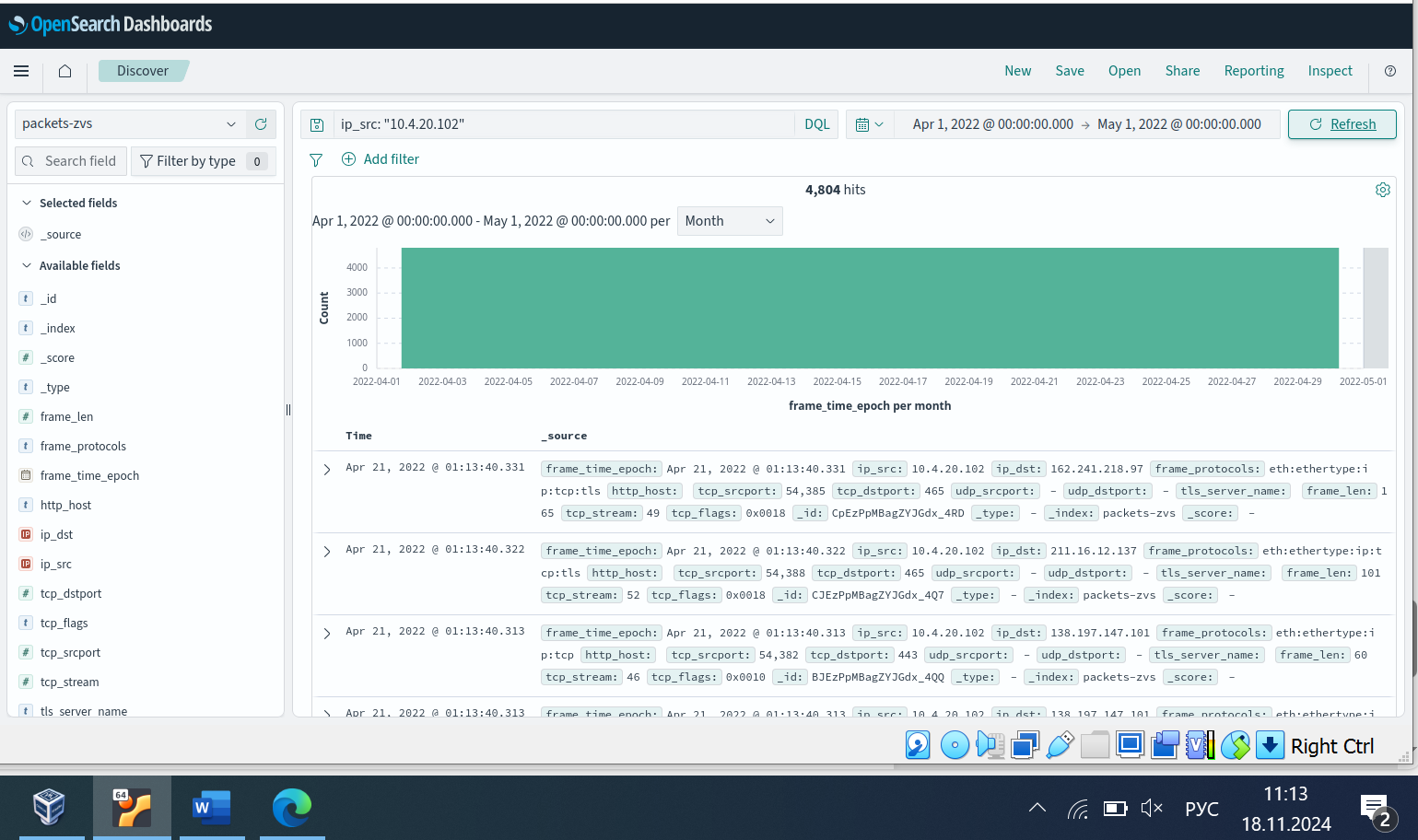
Листинг программы

import csv  
from opensearchpy import OpenSearch  
  
# Настройки подключения к OpenSearch  
OPENSEARCH\_HOST = "http://localhost:9205" # Порт 9205 соответствует вашему docker-compose  
USERNAME = "admin"  
PASSWORD = "Vszub@2409"  
INDEX\_NAME = "packets-zvs" # Имя индекса в OpenSearch  
  
# Инициализация клиента OpenSearch  
client = OpenSearch(  
 hosts=[OPENSEARCH\_HOST],  
 http\_auth=(USERNAME, PASSWORD),  
 use\_ssl=False,  
 verify\_certs=False  
)  
  
# Функция создания индекса  
def create\_index(index\_name):  
 index\_body = {  
 "settings": {  
 "number\_of\_shards": 1,  
 "number\_of\_replicas": 0  
 },  
 "mappings": {  
 "properties": {  
 "frame\_time\_epoch": {"type": "date", "format": "epoch\_second"},  
 "ip\_src": {"type": "ip"},  
 "ip\_dst": {"type": "ip"},  
 "frame\_protocols": {"type": "text"},  
 "http\_host": {"type": "keyword"},  
 "tcp\_srcport": {"type": "integer"},  
 "tcp\_dstport": {"type": "integer"},  
 "udp\_srcport": {"type": "integer"},  
 "udp\_dstport": {"type": "integer"},  
 "tls\_server\_name": {"type": "keyword"},  
 "frame\_len": {"type": "integer"},  
 "tcp\_stream": {"type": "integer"},  
 "tcp\_flags": {"type": "text"}  
 }  
 }  
 }  
 if not client.indices.exists(index=index\_name):  
 response = client.indices.create(index=index\_name, body=index\_body)  
 print(f"Index '{index\_name}' created.")  
 else:  
 print(f"Index '{index\_name}' already exists.")  
  
# Функция загрузки данных из CSV в OpenSearch  
def load\_data\_to\_opensearch(csv\_file, index\_name):  
 with open(csv\_file, mode='r') as file:  
 reader = csv.DictReader(file, delimiter=';')  
 for i, row in enumerate(reader):  
 doc = {  
 "frame\_time\_epoch": float(row.get("frame.time\_epoch", 0)),  
 "ip\_src": row.get("ip.src"),  
 "ip\_dst": row.get("ip.dst"),  
 "frame\_protocols": row.get("frame.protocols"),  
 "http\_host": row.get("http.host"),  
 "tcp\_srcport": int(row.get("tcp.srcport", 0)) if row.get("tcp.srcport") else None,  
 "tcp\_dstport": int(row.get("tcp.dstport", 0)) if row.get("tcp.dstport") else None,  
 "udp\_srcport": int(row.get("udp.srcport", 0)) if row.get("udp.srcport") else None,  
 "udp\_dstport": int(row.get("udp.dstport", 0)) if row.get("udp.dstport") else None,  
 "tls\_server\_name": row.get("tls.handshake.extensions\_server\_name"),  
 "frame\_len": int(row.get("frame.len", 0)),  
 "tcp\_stream": int(row.get("tcp.stream", 0)) if row.get("tcp.stream") else None,  
 "tcp\_flags": row.get("tcp.flags")  
 }  
 response = client.index(index=index\_name, body=doc)  
 if response.get("result") == "created":  
 print(f"Document {i + 1} indexed successfully.")  
 else:  
 print(f"Failed to index document {i + 1}: {response}")  
  
# Основной поток  
if \_\_name\_\_ == "\_\_main\_\_":  
 CSV\_FILE = "result1.csv" # Замените на путь к вашему CSV файлу  
 create\_index(INDEX\_NAME)  
 load\_data\_to\_opensearch(CSV\_FILE, INDEX\_NAME)

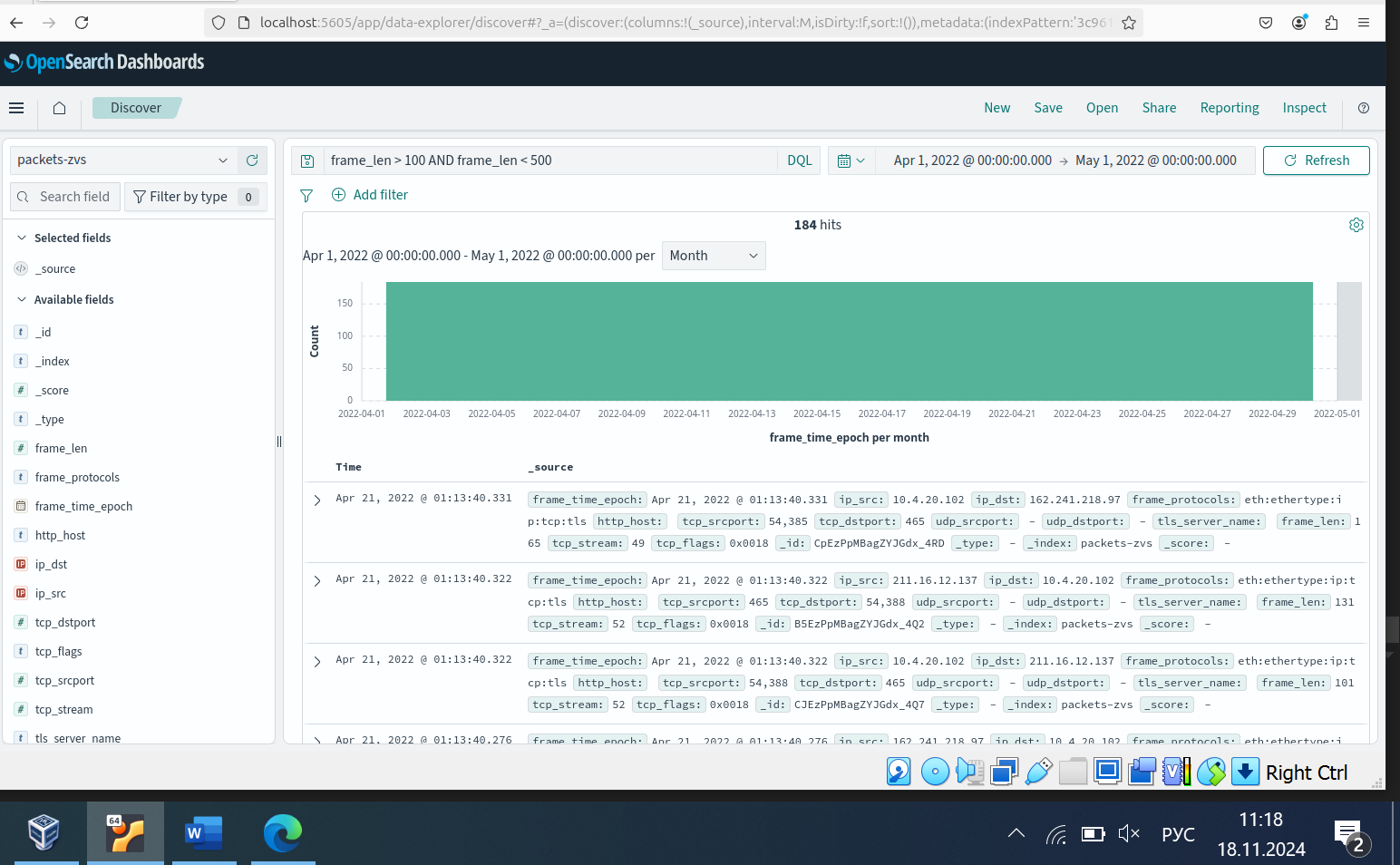


Изучите возможности поиска и анализа данных в разделе «Discover» (не менее 5 различных запросов).

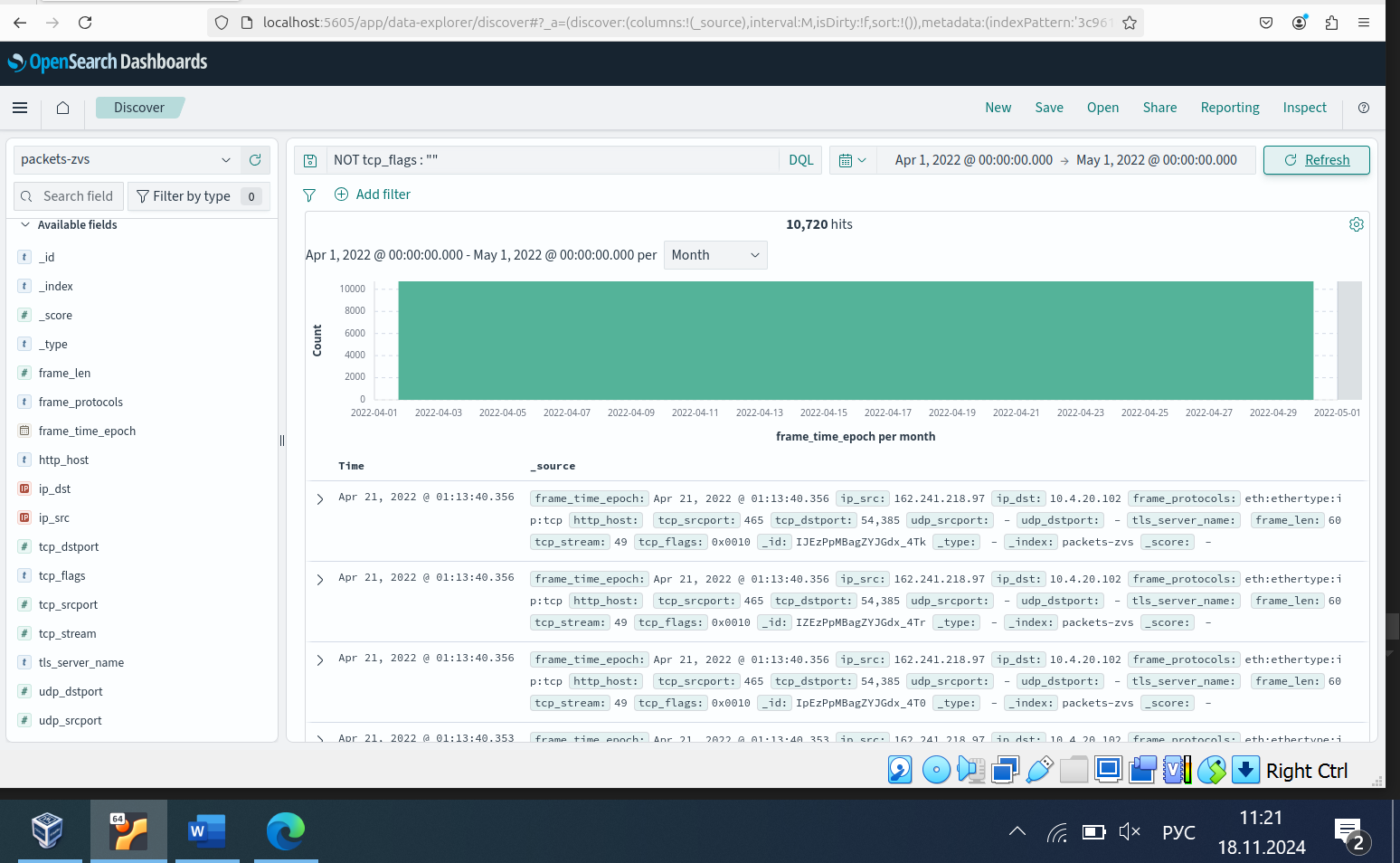
Ip src= 10.4.20.102



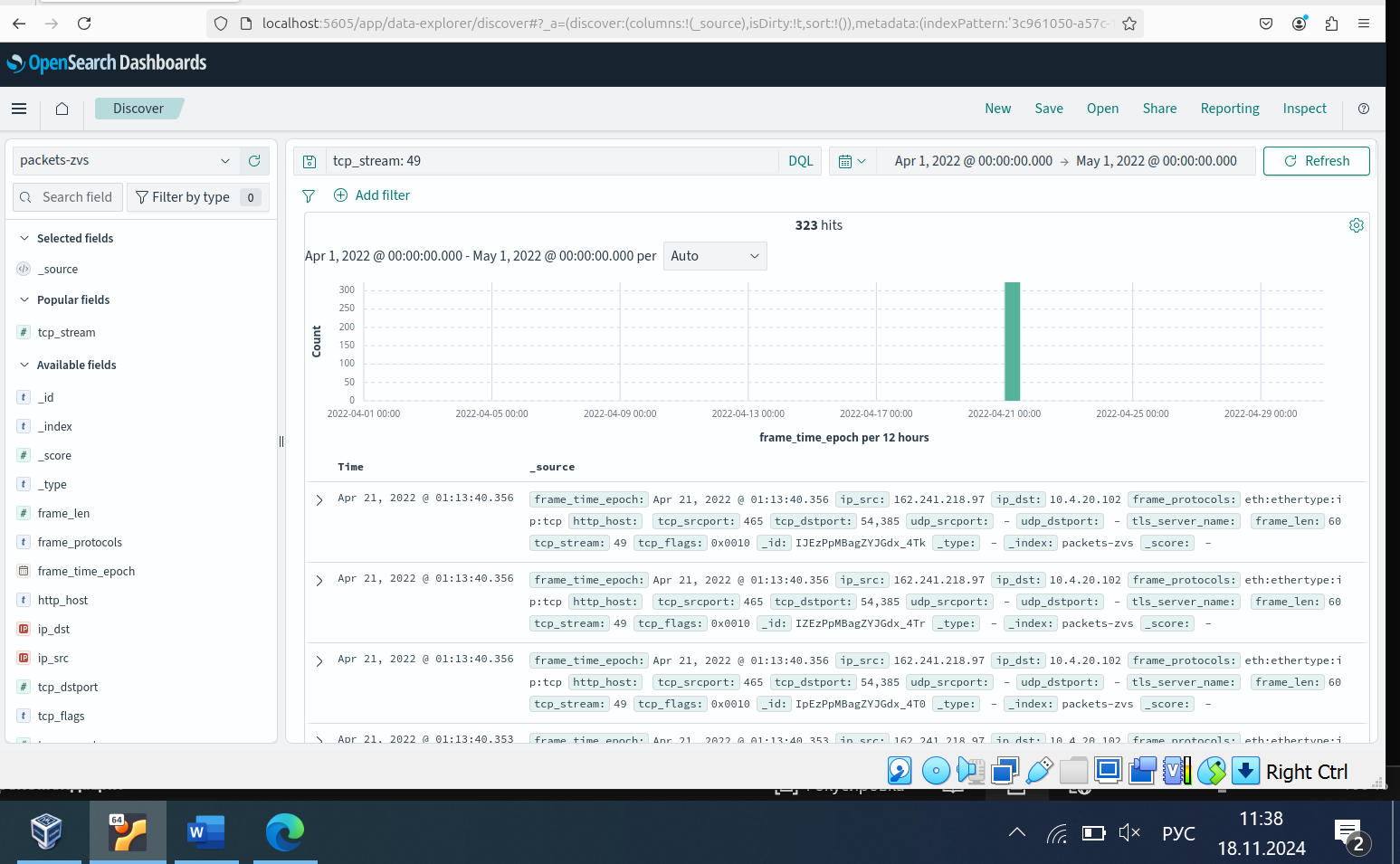
Поиск пакетов где 100<=frame\_length<=500



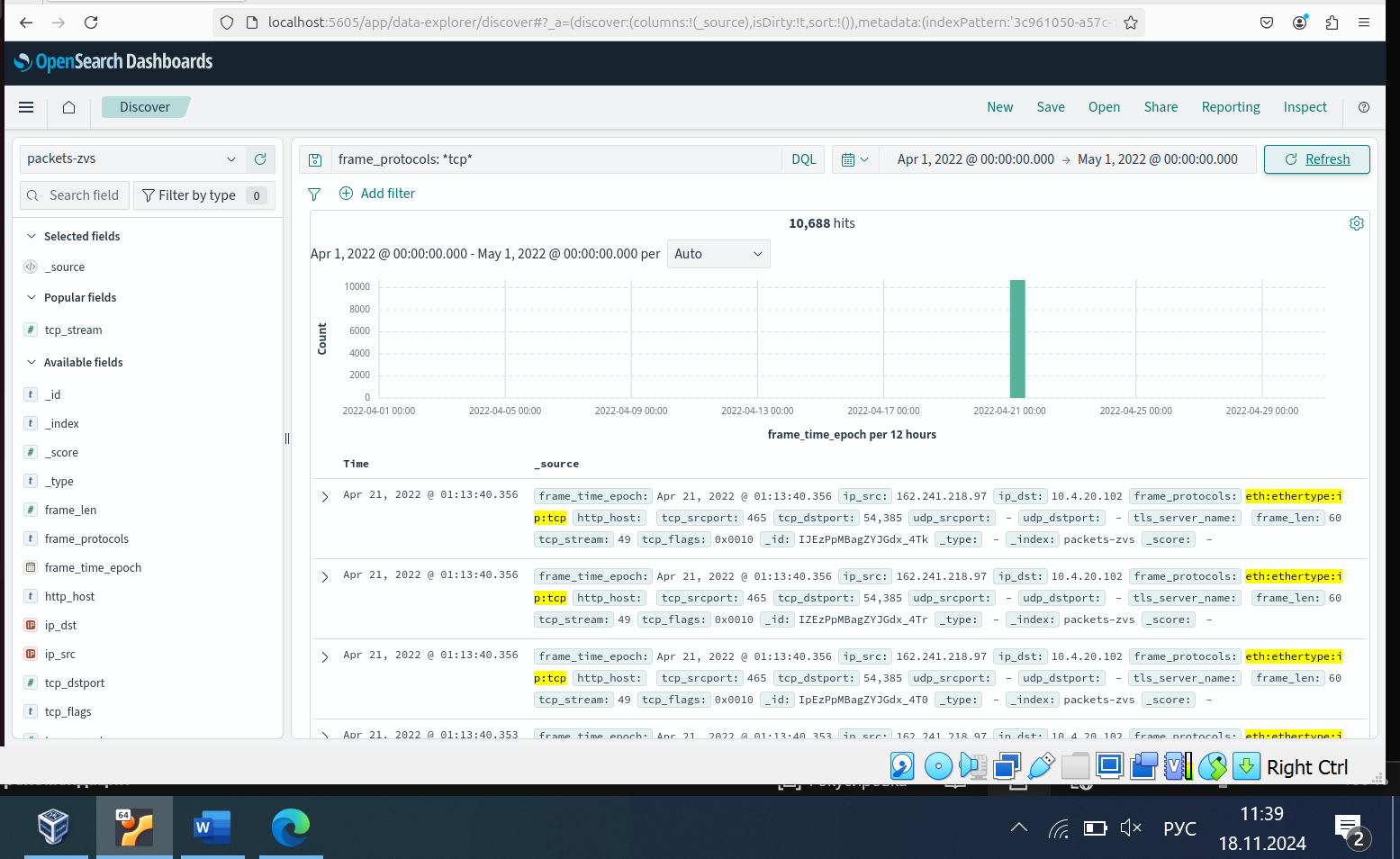
Поиск вывод записей действий (tcp\_flags не путое)



Поиск всех пакетов сессии 49



Все пакеты, где использовался tcp



Сформируйте не менее 5 различных дашбордов (раздел «Dashboards»). Каждый должен иметь уникальное значение. Поясните их назначение и какие выводы вы сделали по результатам анализа данных.