МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт кибербезопасности и цифровых технологий
(наименование института, филиала)

Кафедра КБ-2«Информационно-аналитические системы кибербезопасности»
(наименование кафедры)

ОТЧЕТ

по дисциплине: «Технологии хранения в системах кибербезопасности»

Задание получил:

III курс, группа БИСО-02-23 _____
Подпись

Макаревич Сергей Витальевич
ФИО

Проверил:
« » 2025 г. _____
Дата *Подпись*

_____ Селин А. А.
Отметка / результат *ФИО*

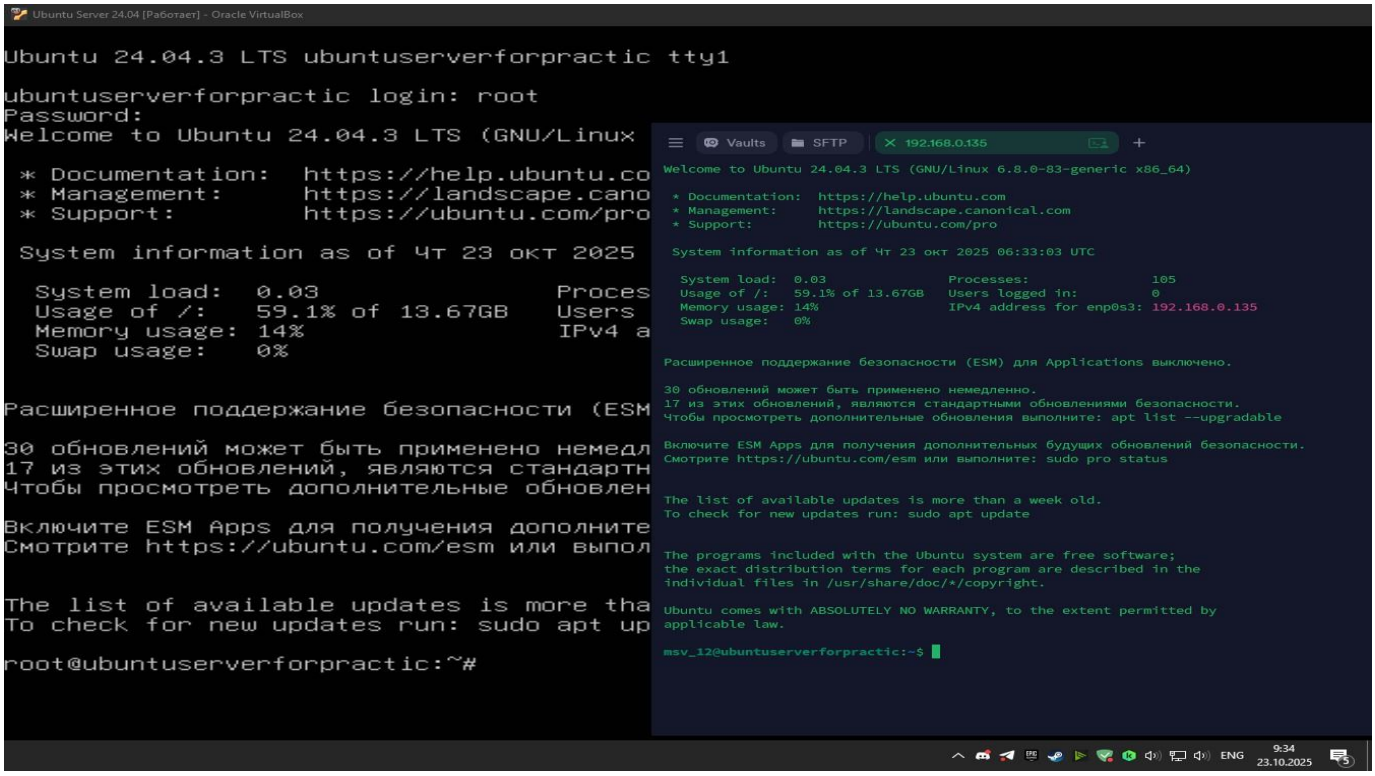
Москва 2025 г.

Оглавление

ПРАКТИЧЕСКАЯ РАБОТА № 4
«Знакомство с инструментами для работы с частично структурированными данными на примере документо-ориентированной СУБД MongoDB»

Цель работы – получение навыков развертывания приложений с использованием Docker.

1. Запуск виртуальной машины с Ubuntu Server 24.04, подключение к ней по ssh используя ПО Termius для удобства взаимодействия.



2. Создание пользователя и настройка окружения

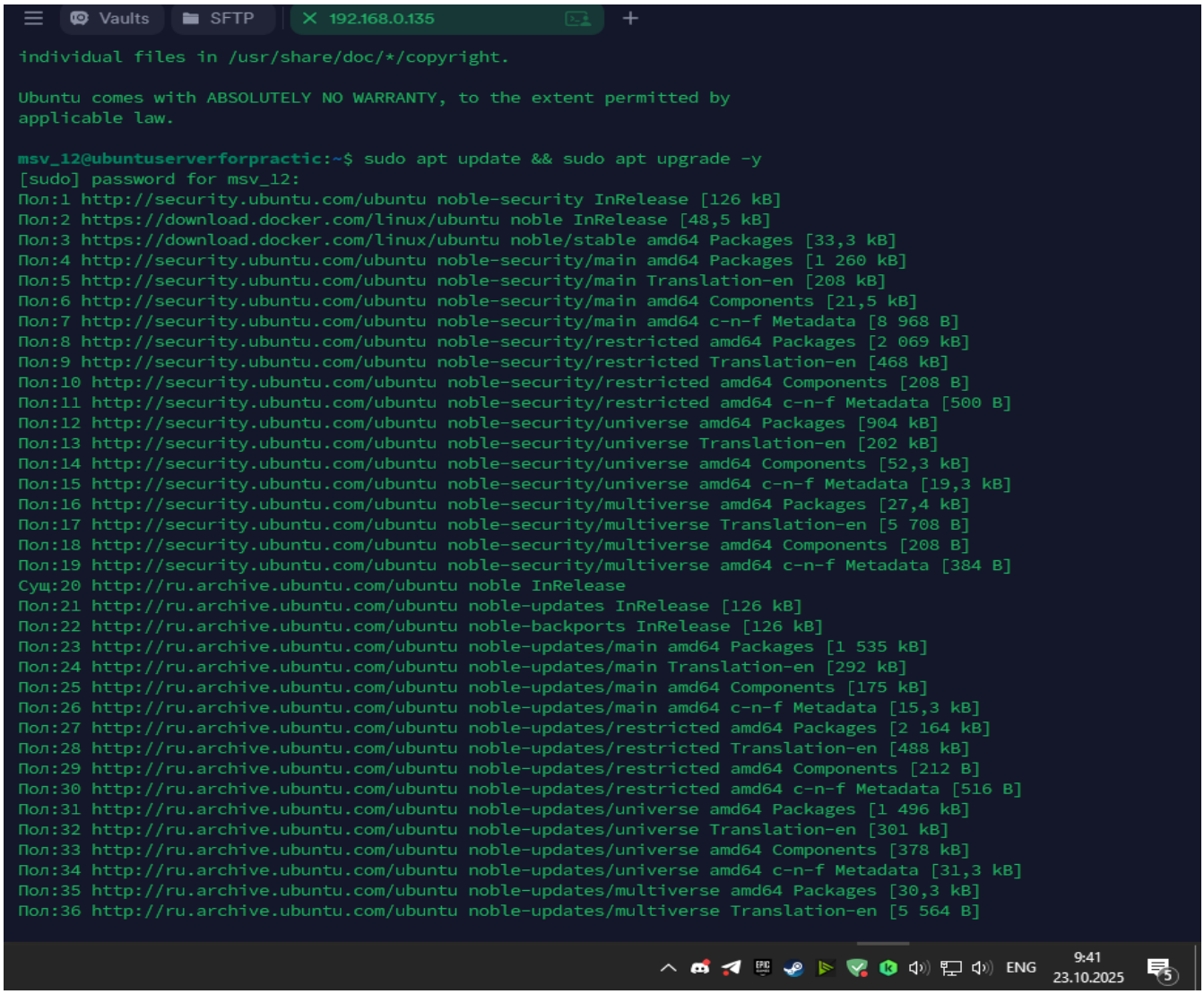
```
root@ubuntuserverforpractic:~# sudo adduser msv_12
info: Adding user `msv_12' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `msv_12' (1001) ...
info: Adding new user `msv_12' (1001) with group `msv_12 (1001)' ...
info: Creating home directory `/home/msv_12' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for msv_12
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `msv_12' to supplemental / extra groups `users' ...
info: Adding user `msv_12' to group `users' ...
```

```
root@ubuntuserverforpractic:~# sudo usermod -aG sudo msv_12
root@ubuntuserverforpractic:~#
```

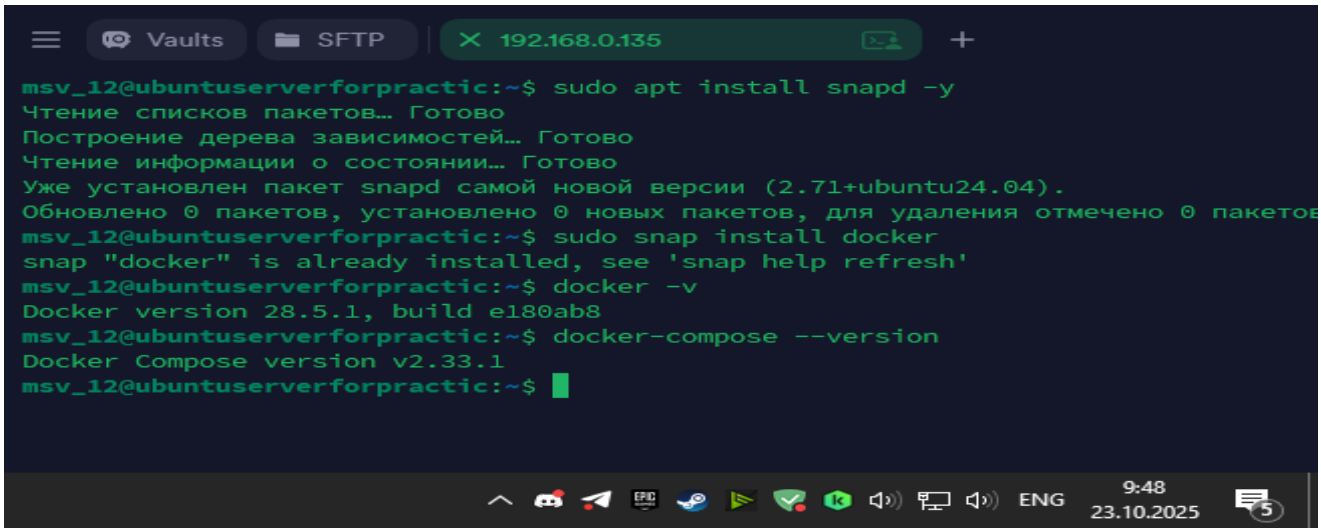
```
root@ubuntuserverforpractic:~# su msv_12
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

msv_12@ubuntuserverforpractic:/root$
```

3. Установка Docker и Docker Compose

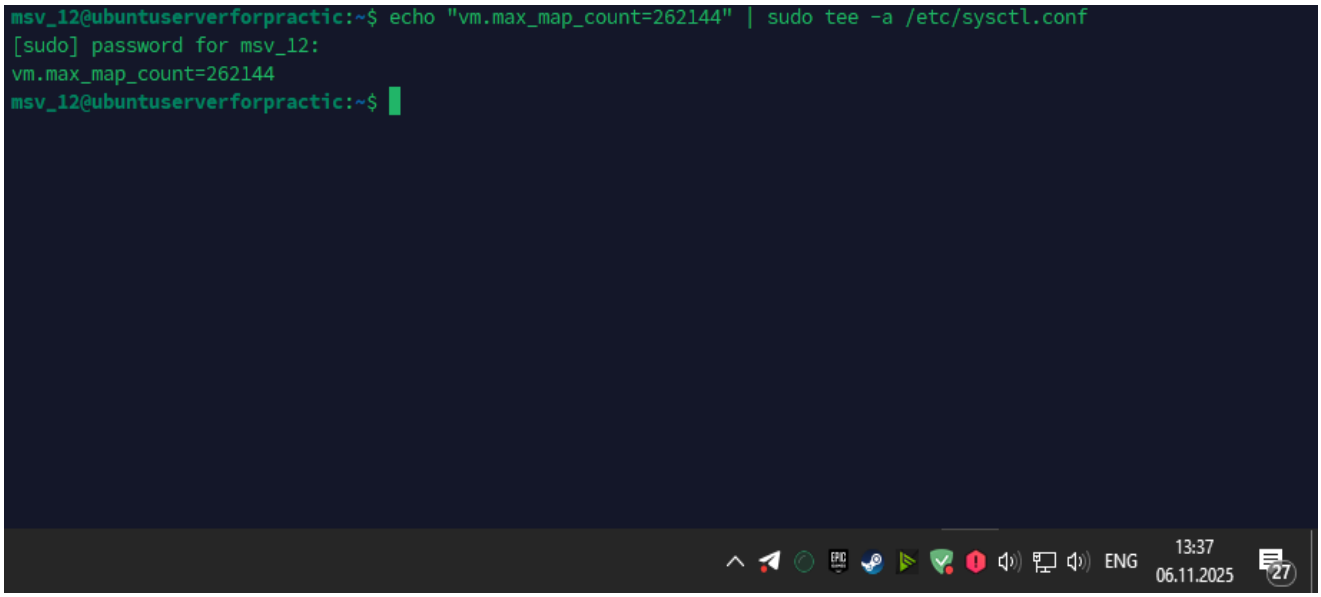
3.1. Обновление пакетов и установка зависимостей



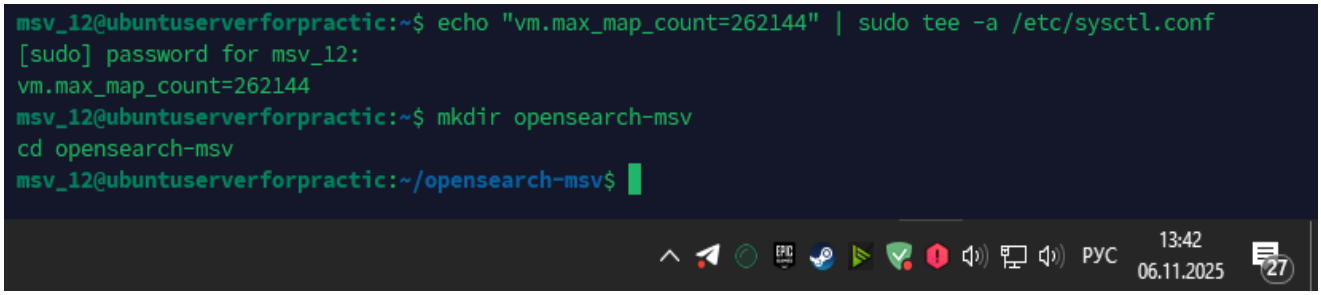
3.2. Установка Docker и Docker Compose через snap. Проверка установки.



4. Настройка параметров ядра(Постоянное изменение)



5. Создание docker-compose.yml для OpenSearch

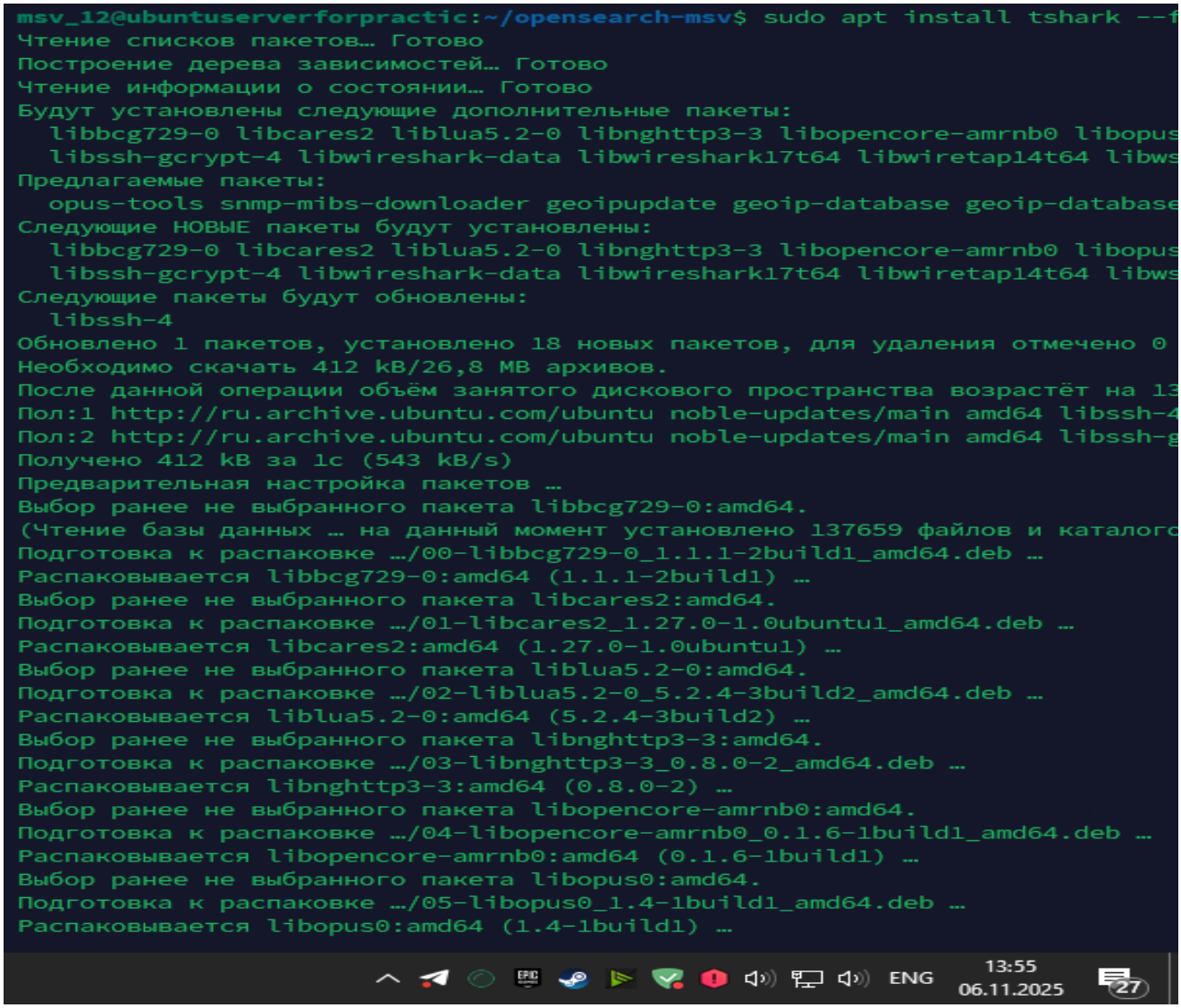


```yaml
services:
  opensearch:
    image: opensearchproject/opensearch:latest
    container_name: opensearch_msv
    environment:
      - cluster.name=opensearch-cluster-msv
      - node.name=opensearch-node-msv
      - discovery.type=single-node
      # УДАЛИТЬ ЭТУ СТРОКУ: - cluster.initial_cluster_manager_nodes=opensearch-node-msv
      - bootstrap.memory_lock=true
      - "OPENSEARCH_JAVA_OPTS=-Xms512m -Xmx512m"
      - "DISABLE_INSTALL_DEMO_CONFIG=true"
      - "DISABLE_SECURITY_PLUGIN=true"
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536
        hard: 65536
    volumes:
      - opensearch-data-msv:/usr/share/opensearch/data
    ports:
      - "9212:9200"
      - "9600:9600"
    networks:
      - opensearch-net-msv
    restart: unless-stopped

  dashboards:
    image: opensearchproject/opensearch-dashboards:latest
    container_name: dashboards_msv
    environment:
      - 'OPENSEARCH_HOSTS=["http://opensearch:9200"]'
      - "DISABLE_SECURITY_DASHBOARDS_PLUGIN=true"
    ports:
      - "5612:5601"
    networks:
      - opensearch-net-msv
    restart: unless-stopped
    depends_on:
      - opensearch

volumes:
  opensearch-data-msv:

networks:
  opensearch-net-msv:
```

6. Запуск OpenSearch

```
msv_12@ubuntuserverforpractic:~/opensearch-msv$ sudo docker-compose up -d
[+] Running 2/2
 ! dashboards       Interrupted
 ✗ opensearch Error Get "https://registry-1.docker.io/v2/": net/http: TLS handshake timeout
Error response from daemon: Get "https://registry-1.docker.io/v2/": net/http: TLS handshake timeout
msv_12@ubuntuserverforpractic:~/opensearch-msv$ sudo docker-compose up -d
[+] Running 10/14
 ⠙ dashboards [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿] 497.3MB / 497.6MB Pulling
   ✓ 6b8641a67e6c Pull complete
   ✓ 1ea1219adecd Pull complete
   ✓ 834cbf21ef4b Pull complete
   ✓ d7ef3b8e965e Pull complete
   ⠸ 758c91650d37 Extracting      [==================================>                ]  287.4MB/417MB
   ✓ 4f4fb700ef54 Download complete
 ⠙ opensearch [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿] 1.048GB / 1.048GB Pulling
   ✓ a6a2e926a3ac Pull complete
   ✓ 4324580a1ad5 Pull complete
   ✓ 7bb4c0e3d646 Pull complete
   ⠙ ade63c612292 Downloading     [================================================>]  989.9MB/989.9MB
   ✓ d58586574fdd Download complete
   ✓ 1f4cdd2bc9bb Download complete
write /var/snap/docker/common/var-lib-docker/tmp/GetImageBlob703835994: no space left on device
msv_12@ubuntuserverforpractic:~/opensearch-msv$
```

7. Установка TShark



8. Скачивание образцов трафика

В процессе поиска файлов на сайте https://www.malware-traffic-analysis.net были найдены три наиболее подходящих по условию задания:

1. https://www.malware-traffic-analysis.net/2023/02/13/2023-02-13-IcedID-traffic-carved-and-sanitized.pcap.zip
2. https://www.malware-traffic-analysis.net/2023/02/23/2023-02-23-example-of-IcedID-from-URL-file-with-WebDAV-traffic.pcap.zip
3. https://www.malware-traffic-analysis.net/2023/02/27/2023-02-27-Qakbot-infection-traffic.pcap.zip

Скачиваем архивы используя wget:

```
msv_12@ubuntuserverforpractic:~/opensearch-msv$ wget https://www.malware-traffic-analysis.net/2023/02/13/2023-02-13-IcedID-traffic-carved-and-saniti
zed.pcap.zip
--2025-11-06 11:22:31--  https://www.malware-traffic-analysis.net/2023/02/13/2023-02-13-IcedID-traffic-carved-and-sanitized.pcap.zip
Resolving www.malware-traffic-analysis.net (www.malware-traffic-analysis.net)... 199.201.110.204
Connecting to www.malware-traffic-analysis.net (www.malware-traffic-analysis.net)|199.201.110.204|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4838817 (4,6M) [application/zip]
Saving to: '2023-02-13-IcedID-traffic-carved-and-sanitized.pcap.zip'

2023-02-13-IcedID-traffic-carved-and 100%[=================================================================>]   4,61M  2,82MB/s    in 1,6s

2025-11-06 11:22:33 (2,82 MB/s) - '2023-02-13-IcedID-traffic-carved-and-sanitized.pcap.zip' saved [4838817/4838817]

msv_12@ubuntuserverforpractic:~/opensearch-msv$ wget https://www.malware-traffic-analysis.net/2023/02/23/2023-02-23-example-of-IcedID-from-URL-file-
with-WebDAV-traffic.pcap.zip
--2025-11-06 11:22:55--  https://www.malware-traffic-analysis.net/2023/02/23/2023-02-23-example-of-IcedID-from-URL-file-with-WebDAV-traffic.pcap.zip
Resolving www.malware-traffic-analysis.net (www.malware-traffic-analysis.net)... 199.201.110.204
Connecting to www.malware-traffic-analysis.net (www.malware-traffic-analysis.net)|199.201.110.204|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2698164 (2,6M) [application/zip]
Saving to: '2023-02-23-example-of-IcedID-from-URL-file-with-WebDAV-traffic.pcap.zip'

2023-02-23-example-of-IcedID-from-UR 100%[=================================================================>]   2,57M  1,83MB/s    in 1,4s

2025-11-06 11:22:57 (1,83 MB/s) - '2023-02-23-example-of-IcedID-from-URL-file-with-WebDAV-traffic.pcap.zip' saved [2698164/2698164]

msv_12@ubuntuserverforpractic:~/opensearch-msv$ wget https://www.malware-traffic-analysis.net/2023/02/27/2023-02-27-Qakbot-infection-traffic.pcap.zi
p
--2025-11-06 11:23:13--  https://www.malware-traffic-analysis.net/2023/02/27/2023-02-27-Qakbot-infection-traffic.pcap.zip
Resolving www.malware-traffic-analysis.net (www.malware-traffic-analysis.net)... 199.201.110.204
Connecting to www.malware-traffic-analysis.net (www.malware-traffic-analysis.net)|199.201.110.204|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17129439 (16M) [application/zip]
Saving to: '2023-02-27-Qakbot-infection-traffic.pcap.zip'

2023-02-27-Qakbot-infection-traffic. 100%[=================================================================>]  16,33M  7,35MB/s    in 2,2s

2025-11-06 11:23:16 (7,35 MB/s) - '2023-02-27-Qakbot-infection-traffic.pcap.zip' saved [17129439/17129439]

msv_12@ubuntuserverforpractic:~/opensearch-msv$
```

Теперь распакуем архивы:

```
msv_12@ubuntuserverforpractic:~/opensearch-msv$ unzip -P infected_20230213 2023-02-13-IcedID-traffic-carved-and-sanitized.pcap.zip
unzip -P infected_20230223 2023-02-23-example-of-IcedID-from-URL-file-with-WebDAV-traffic.pcap.zip
unzip -P infected_20230227 2023-02-27-Qakbot-infection-traffic.pcap.zip
Archive:  2023-02-13-IcedID-traffic-carved-and-sanitized.pcap.zip
  inflating: 2023-02-13-IcedID-traffic-carved-and-sanitized.pcap
Archive:  2023-02-23-example-of-IcedID-from-URL-file-with-WebDAV-traffic.pcap.zip
  inflating: 2023-02-23-example-of-IcedID-from-URL-file-with-WebDAV-traffic.pcap
Archive:  2023-02-27-Qakbot-infection-traffic.pcap.zip
  inflating: 2023-02-27-Qakbot-infection-traffic.pcap
msv_12@ubuntuserverforpractic:~/opensearch-msv$ ls -la *.pcap
-rw-r--r-- 1 msv_12 msv_12  5112540 фев 13  2023 2023-02-13-IcedID-traffic-carved-and-sanitized.pcap
-rw-r--r-- 1 msv_12 msv_12  2947526 фев 23  2023 2023-02-23-example-of-IcedID-from-URL-file-with-WebDAV-traffic.pcap
-rw-r--r-- 1 msv_12 msv_12 19141869 фев 28  2023 2023-02-27-Qakbot-infection-traffic.pcap
msv_12@ubuntuserverforpractic:~/opensearch-msv$
```

9. Обработка PCAP-файлов с помощью TShark

9.1. Первый файл

```
msv_12@ubuntuserverforpractic:~/opensearch-msv$ sudo tshark -r 2023-02-13-IcedID-traffic-carved-and-sanitized.pcap \
  -Y "http.host or tls.handshake.extensions_server_name or dns.qry.name" \
  -T fields \
  -e frame.time_epoch \
  -e ip.src \
  -e ip.dst \
  -e tcp.srcport \
  -e tcp.dstport \
  -e udp.srcport \
  -e udp.dstport \
  -e http.host \
  -e tls.handshake.extensions_server_name \
  -e frame.protocols \
  -e frame.len \
  -e http.request.method \
  -e http.response.code \
  -e dns.qry.name \
  -e tcp.flags \
  -E header=y -E separator=\; -E aggregator=, > result_msv.csv
Running as user "root" and group "root". This could be dangerous.
msv_12@ubuntuserverforpractic:~/opensearch-msv$
```

9.2. Второй файл

```
msv_12@ubuntuserverforpractic:~/opensearch-msv$ sudo tshark -r 2023-02-23-example-of-IcedID-from-URL-file-with-WebDAV-traffic.pcap \
  -Y "http.host or tls.handshake.extensions_server_name or dns.qry.name" \
  -T fields \
  -e frame.time_epoch \
  -e ip.src \
  -e ip.dst \
  -e tcp.srcport \
  -e tcp.dstport \
  -e udp.srcport \
  -e udp.dstport \
  -e http.host \
  -e tls.handshake.extensions_server_name \
  -e frame.protocols \
  -e frame.len \
  -e http.request.method \
  -e http.response.code \
  -e dns.qry.name \
  -e tcp.flags \
  -E header=n -E separator=\; -E aggregator=, >> result_msv.csv
Running as user "root" and group "root". This could be dangerous.
msv_12@ubuntuserverforpractic:~/opensearch-msv$
```

9.3. Третий файл

```
msv_12@ubuntuserverforpractic:~/opensearch-msv$ sudo tshark -r 2023-02-27-Qakbot-infection-traffic.pcap \
  -Y "http.host or tls.handshake.extensions_server_name or dns.qry.name" \
  -T fields \
  -e frame.time_epoch \
  -e ip.src \
  -e ip.dst \
  -e tcp.srcport \
  -e tcp.dstport \
  -e udp.srcport \
  -e udp.dstport \
  -e http.host \
  -e tls.handshake.extensions_server_name \
  -e frame.protocols \
  -e frame.len \
  -e http.request.method \
  -e http.response.code \
  -e dns.qry.name \
  -e tcp.flags \
  -E header=n -E separator=\; -E aggregator=, >> result_msv.csv
Running as user "root" and group "root". This could be dangerous.
 ** (tshark:10924) 11:38:45.606694 [Epan WARNING] -- Dissector bug, protocol CLDAP, in packet 2289: ./epan/dissectors/packet-ldap.c:2180: failed ass
ertion "recursion_depth <= 100"
 ** (tshark:10924) 11:38:45.607356 [Epan WARNING] -- Dissector bug, protocol CLDAP, in packet 2293: ./epan/dissectors/packet-ldap.c:2180: failed ass
ertion "recursion_depth <= 100"
 ** (tshark:10924) 11:38:45.624502 [Epan WARNING] -- Dissector bug, protocol CLDAP, in packet 2865: ./epan/dissectors/packet-ldap.c:2180: failed ass
```

Обоснование по каждому дополнительно выбранному полю:

1. http.request.method:
   Позволяет определить тип выполняемого HTTP-запроса (GET, POST, PROPFIND и др.). Помогает выявить подозрительную активность, например, использование методов WebDAV (PROPFIND, PUT) или массовые автоматизированные GET-запросы, характерные для сканирования и C2-трафика.

2. http.response.code:
   Отражает результат обработки запроса сервером (200, 404, 500 и др.). Позволяет оценить успешность выполнения операций и обнаружить ошибки, связанные с атаками или сканированием. Анализ сочетания с методом запроса помогает понять контекст действий (например, POST + 200 — возможная передача данных).

3. dns.qry.name:
   Показывает доменные имена, к которым выполняются обращения. Используется для выявления вредоносных доменов, DGA-активности и аномальных запросов. Помогает сопоставить IP-адреса с доменами и анализировать сетевое взаимодействие на уровне имён.

4. ssh.protocol:
   Содержит информацию о версии и типе SSH-протокола. Позволяет отслеживать активность удалённого доступа, выявлять несанкционированные подключения или попытки подбора паролей, а также использование устаревших версий SSH.

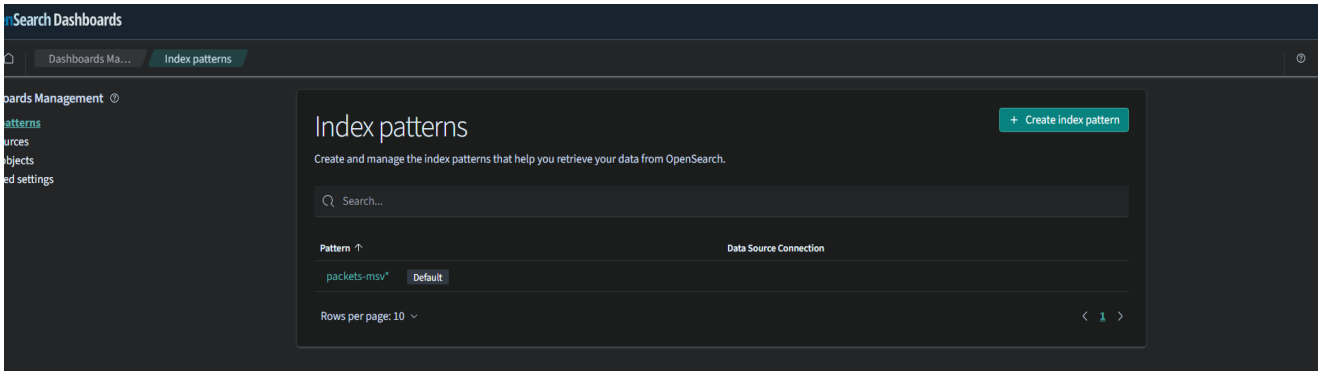
5. ftp.request.command:
   Отражает команды FTP (USER, PASS, STOR, RETR и др.). Полезно для анализа передачи файлов, обнаружения попыток эксфильтрации данных или брутфорса. Позволяет отследить загрузку или выгрузку подозрительных файлов.

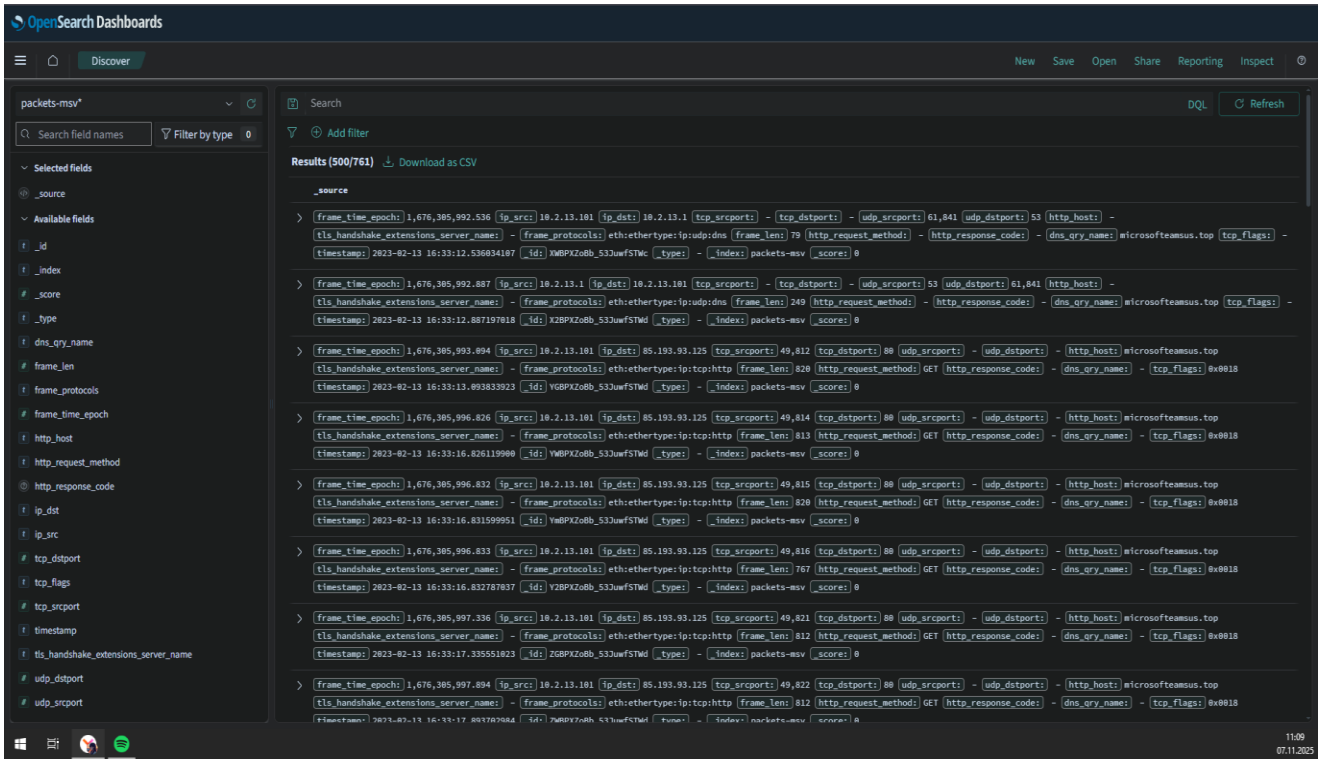
10. Загрузка данных используя Python скрипт.

Используемый Python скрипт:
https://github.com/l1ratch/WC_BISO/blob/main/5_sem/ТХвСК/practic/pr_5/uto.py

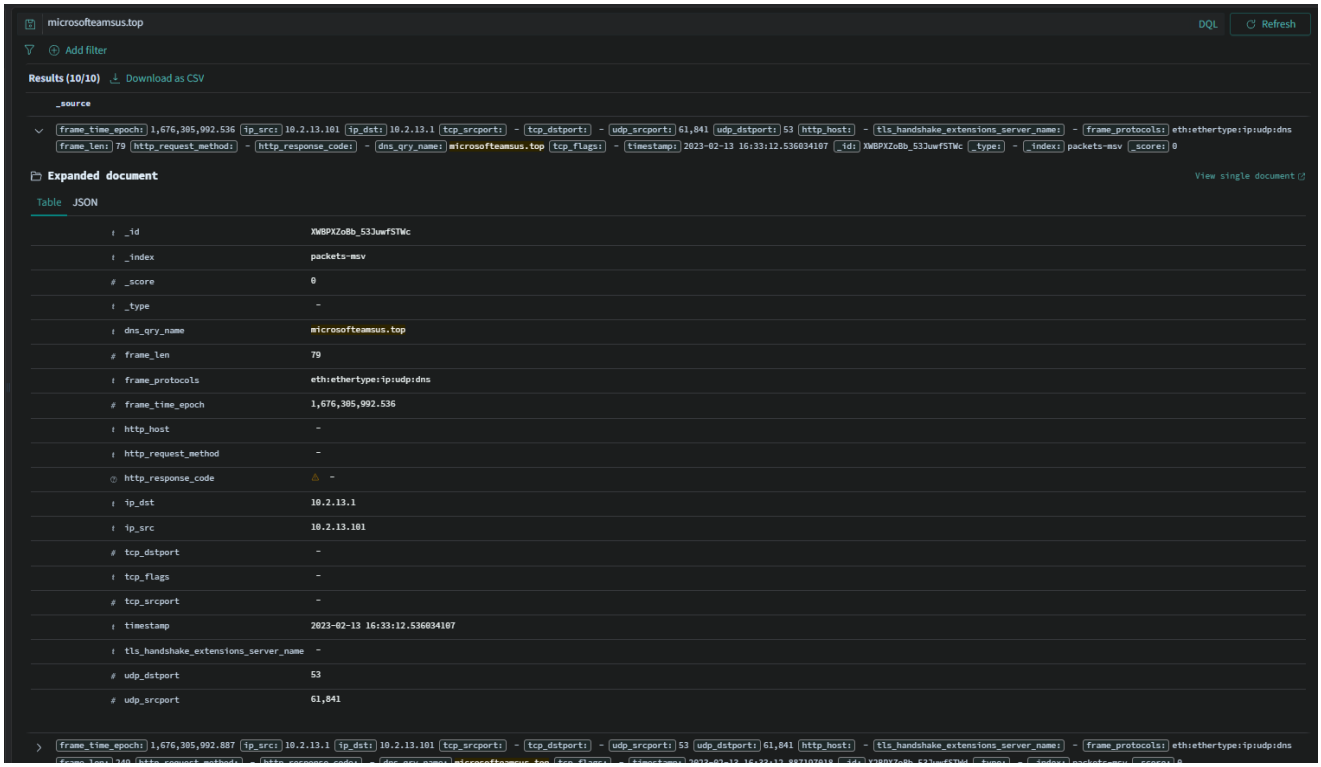
```
(venv) msv_12@ubuntuserverforpractic:~/opensearch-msv$ python uto.py
2025-11-07 07:54:28,003 - INFO - HEAD http://localhost:9212/ [status:200 request:0.076s]
2025-11-07 07:54:28,003 - INFO - Успешное подключение к OpenSearch
2025-11-07 07:54:28,356 - INFO - PUT http://localhost:9212/_template/packets_msv [status:200 request:0.353s]
2025-11-07 07:54:28,357 - INFO - Шаблон индекса 'packets_msv' успешно создан
2025-11-07 07:54:28,357 - INFO - Чтение файла result_msv.csv
2025-11-07 07:54:29,340 - INFO - POST http://localhost:9212/_bulk?refresh=true [status:200 request:0.942s]
2025-11-07 07:54:29,343 - INFO - Всего загружено 761 документов в индекс packets-msv
2025-11-07 07:54:29,343 - INFO - Данные успешно загружены в OpenSearch
2025-11-07 07:54:29,383 - INFO - POST http://localhost:9212/packets-msv/_count [status:200 request:0.038s]
2025-11-07 07:54:29,384 - INFO - В индексе packets-msv теперь 761 документов
(venv) msv_12@ubuntuserverforpractic:~/opensearch-msv$
```

11. Запуск панели, подготовка OpenSearchDashboards.

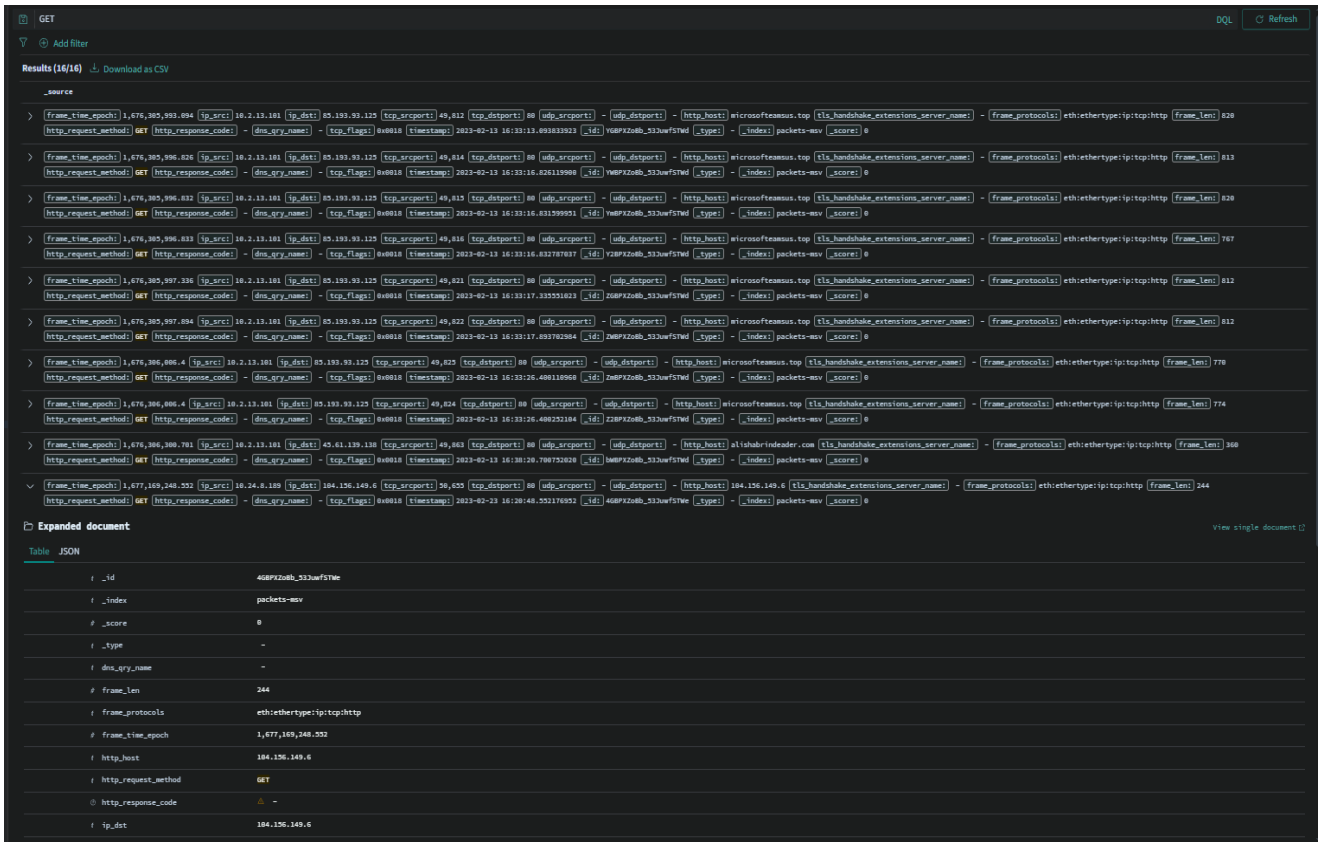


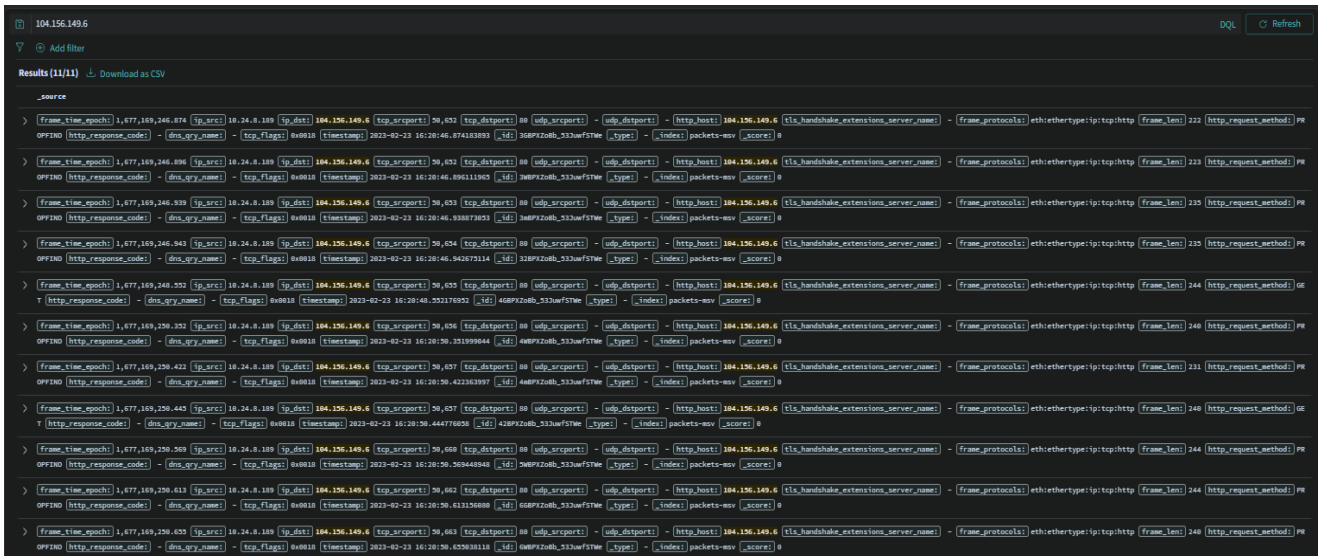


12. Выполним поиск в разделе Discover.

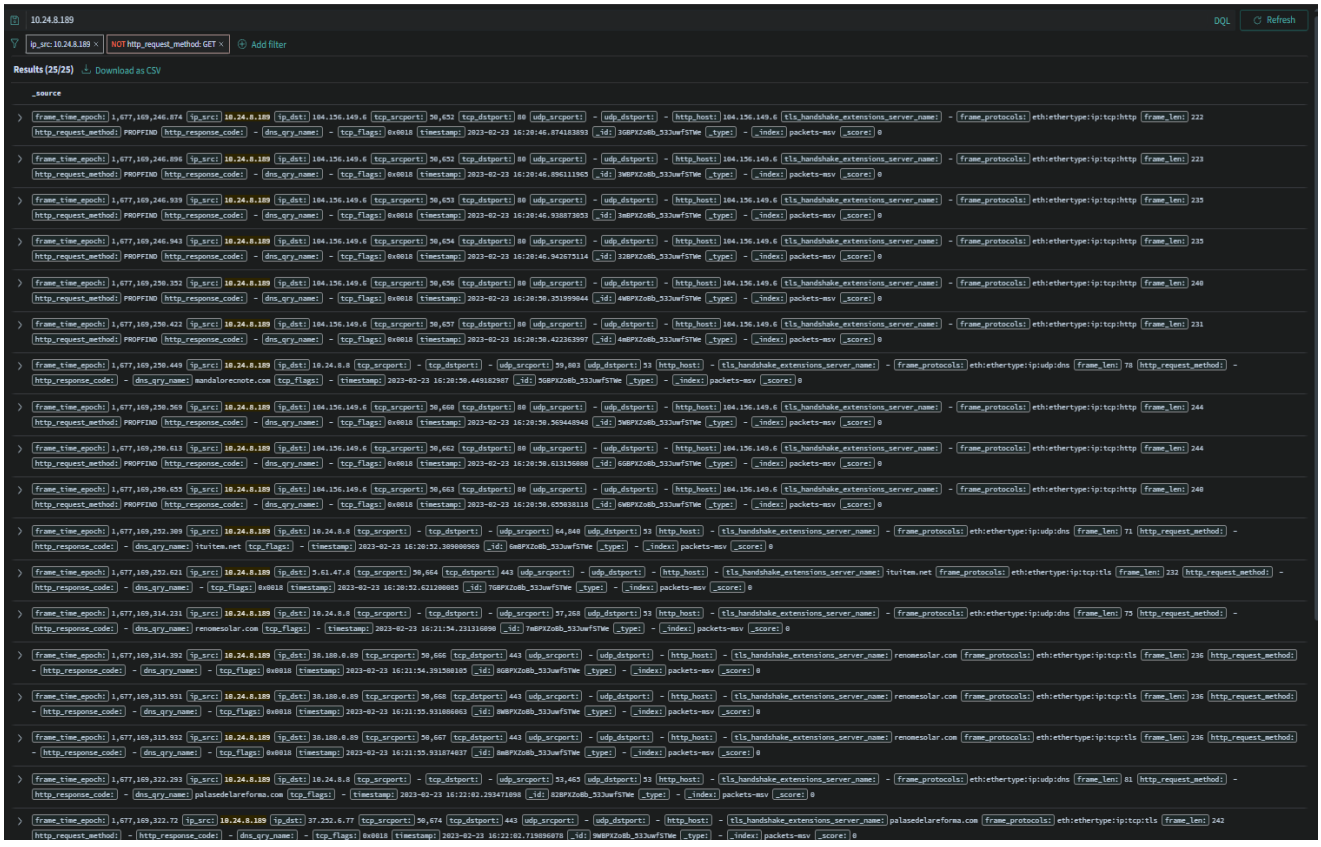


Попробовал функцию поиска, выполнил поисковый запрос по одному из найденных доменов microsoftteamsus.top. Запросы используют метод GET.





Решил изучить все GET запросы. Нашел запросы от http хоста 104.156.149.6. Изучив вывод поиска по хосту заметил, что все выполненные им запросы практически идентичны и используют метод GET, предполагаю, что это скрипт.
Выполню поиск по ip-источнику этих запросов(10.24.8.189) исключив GET запросы:

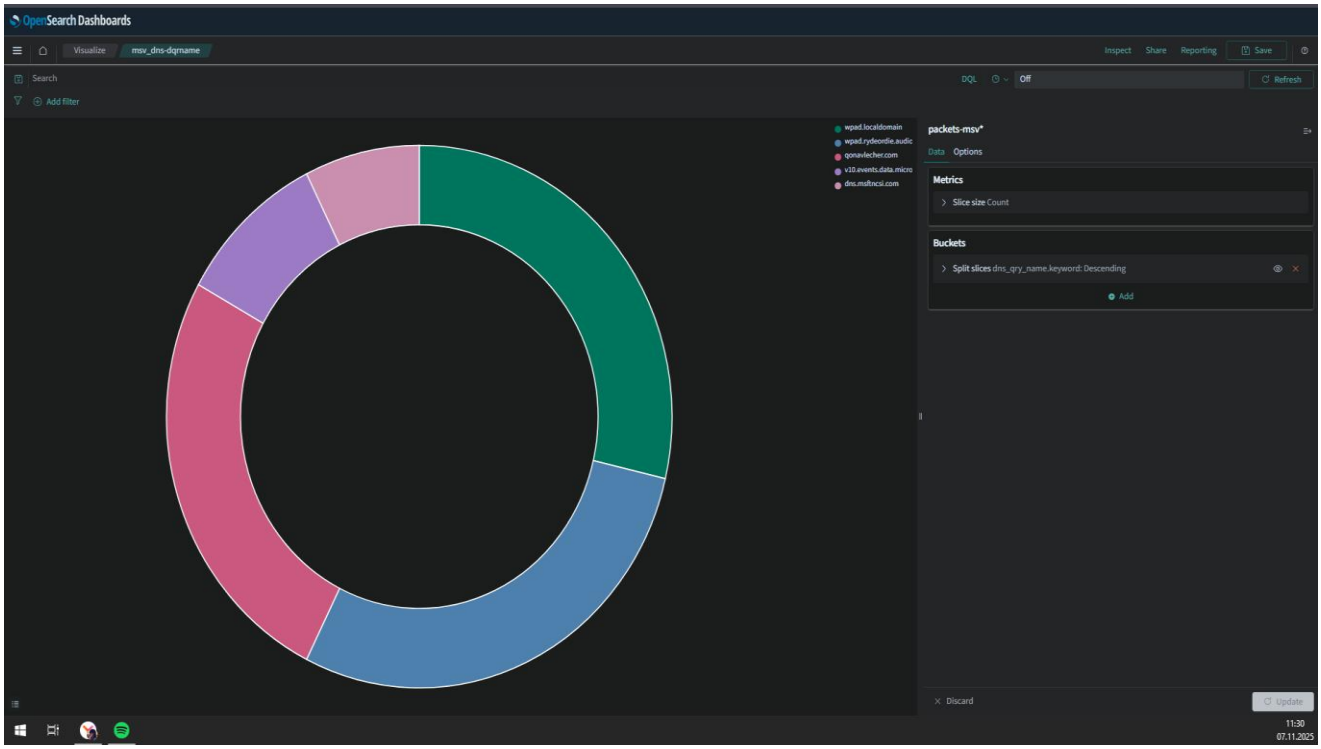


Здесь в основном запросы на порты 443 и 53. Предполагаю, что это хаб или другое устройство организующее устройства в сети, возможно устрйоства на котором расположен dns сервер.

Нашел запросы к 38.180.0.89 по 443, связанные с renomesolar.com. Первые два запроса являются dns запросами. Сначала 10.24.8.189 отправил запрос 10.24.8.8 на 53 порт, затем получил ответ от 10.24.8.8 и начал отправлять запросы 38.180.0.89 на 443 порт. Хороший пример того как работает dns сервер. Можем сделать вывод, что 10.24.8.189 является одним из устройств в сети, а 10.24.8.8 является локальным dns сервером этой сети.

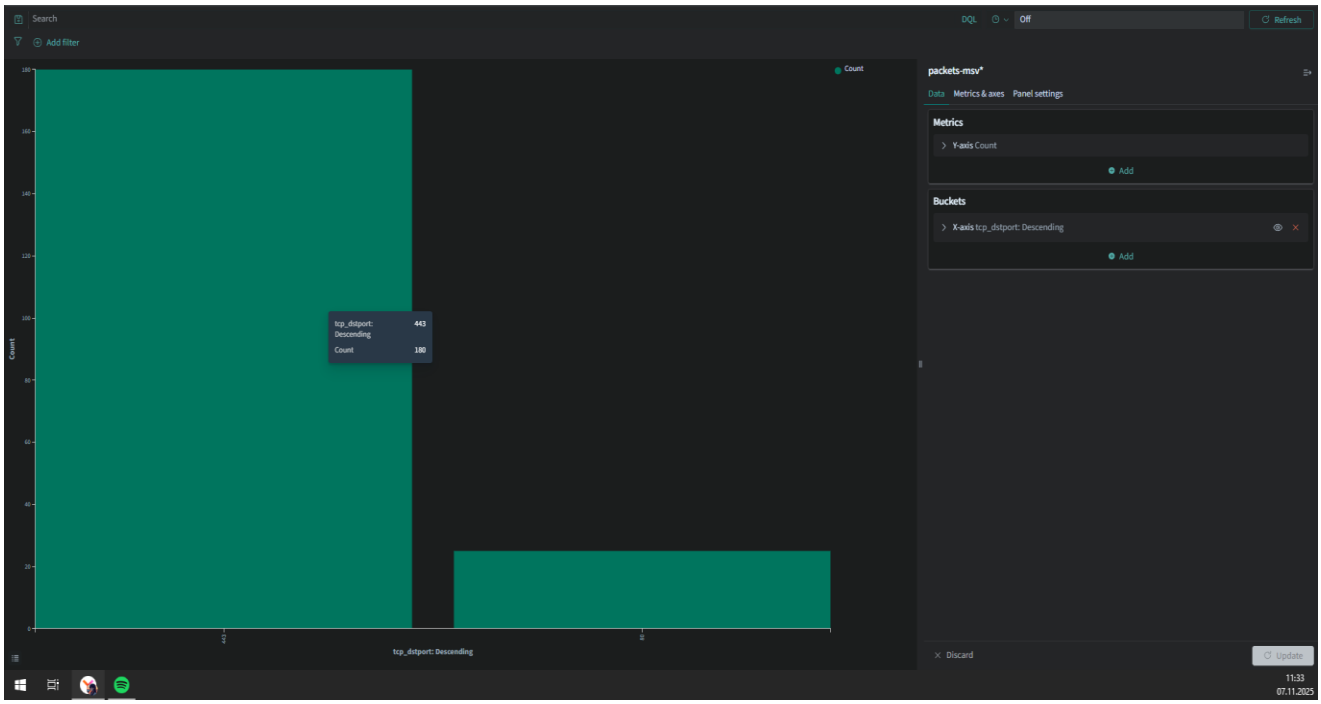
13. Создадим несколько визуализаций.

13.1. По полю dns_qry_name.keyword



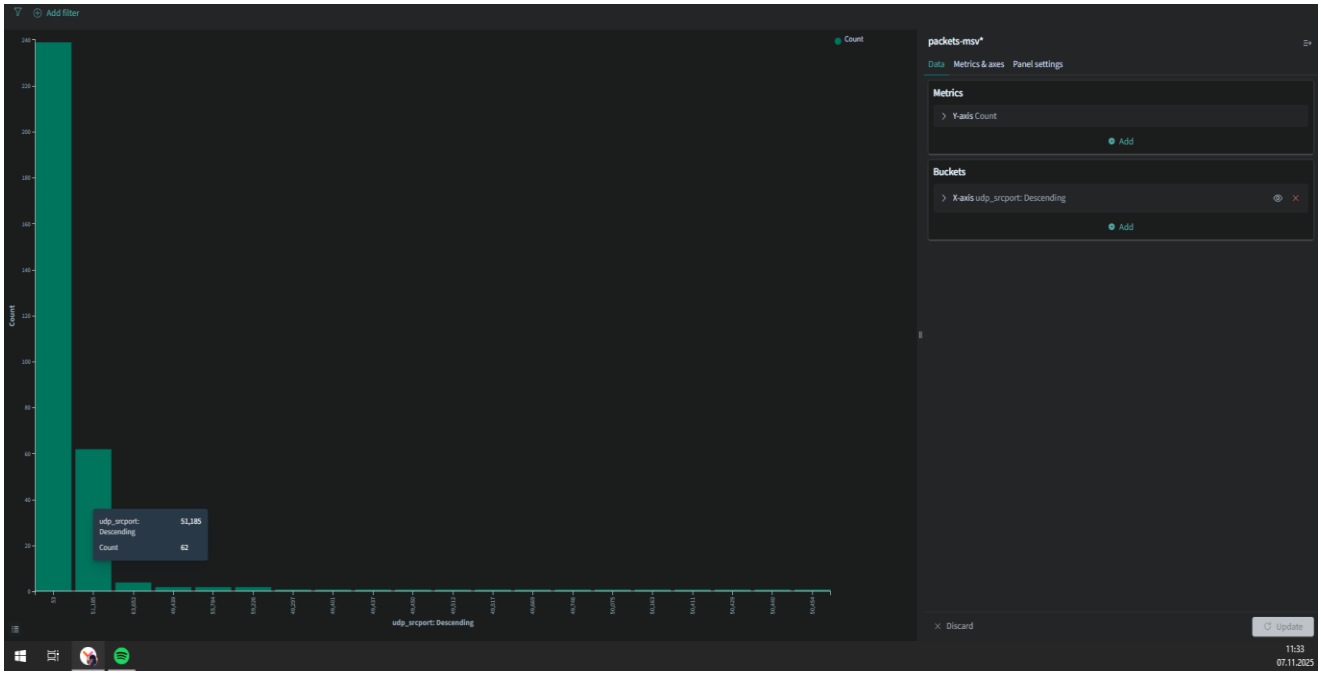
Чаще всего запросы шли к dns wpad.localdomain и wpab.rydeordie.audio – по 78 запросов. Чуть чаще к qonavlecher.com – по 70, и совсем редко к двум от компании Microsoft.

13.2. По полю tcp_dstport



На данной диаграмме видно, что наиболее часто используемым dst портом является 443(180 запросов) из чего можно сделать вывод что в основном все запросы были по защищенному протоколу, лишь 25 запросов относятся к 80 порту.

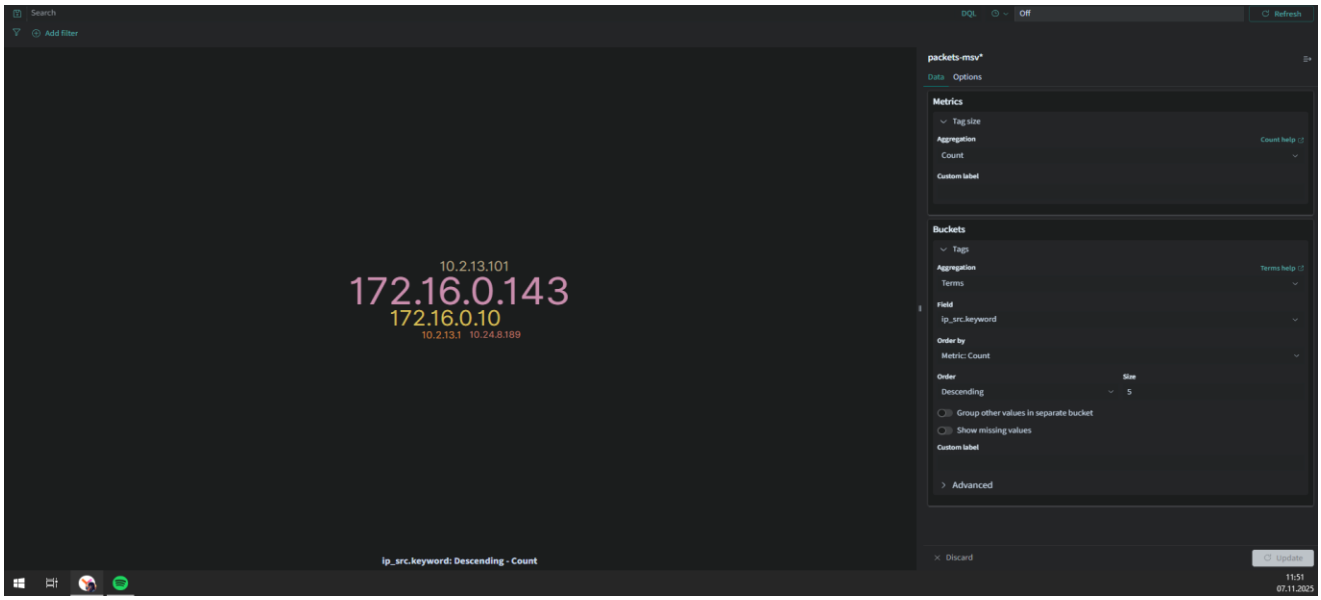
13.3. По полю udp_srcport



На данной диаграмме можно заметить, что самым часто используемым портом является

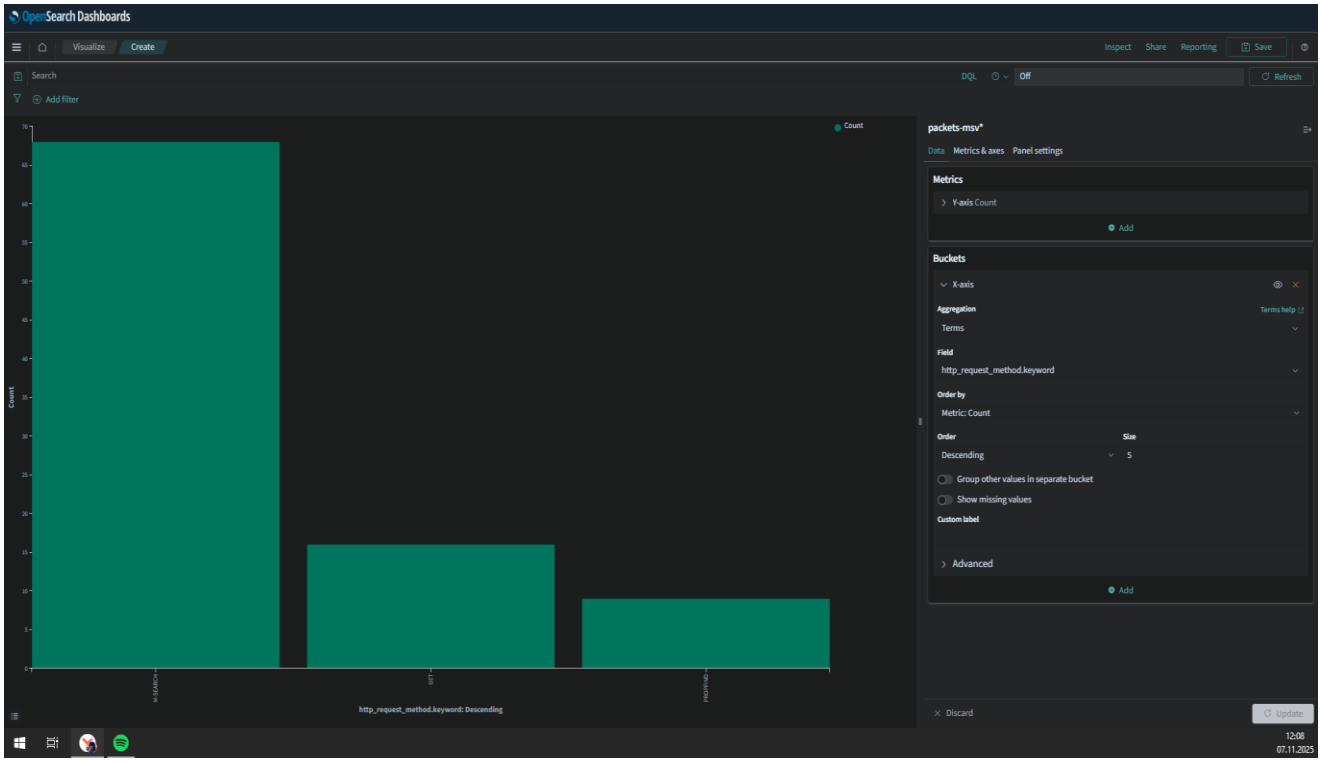
53 - порт который чаще всего используется для dns серверов(239 запросов), на втором месте 51185 порт у которого всего 62 запроса. Следом уже идут другие по 4, 2 и 1 запросам.
(По исследованным диаграммам можно сделать предположение, что на устройстве или в локальной сети устройства был запущен локальный DNS-сервер. Вероятно для отслеживания запросов.)

13.4. По полю ip_src



На это диаграмме мы видим вывод с сортировкой по частоте использования ip адреса источников запросов – локальные адреса с которых выполняются запросы. Наиболее часто используемым является 172.16.0.143 – который вероятно является одним из основных адресов устройства в его локальной сети.

13.5. По полю http_rewuest_method

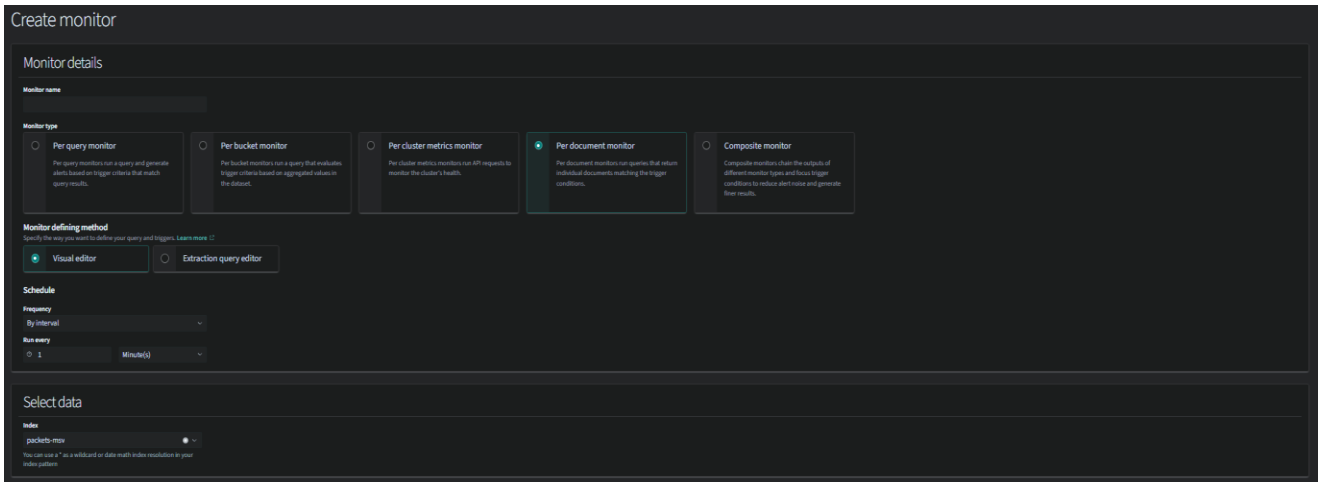


Здесь мы видим диаграмму по используемым методам http запросов. Наиболее часто используемым является метод M-SEARCH , который используется устройствами (например, Smart TV, медиаплеерами) для поиска других сервисов по сети – 68 запросов, Значительно меньшее количество стандартных запросов GET и совсем немного запросов PROPFIND (метод для работы с веб-ресурсами, как WebDAV) говорят о том, что обычный веб-трафик и управление файлами являются вторичной активностью в данном срезе данных. Следовательно можно сделать вывод, что в основном сетевая активность связана с автоматическим поиском и объявлением сетевых устройств.

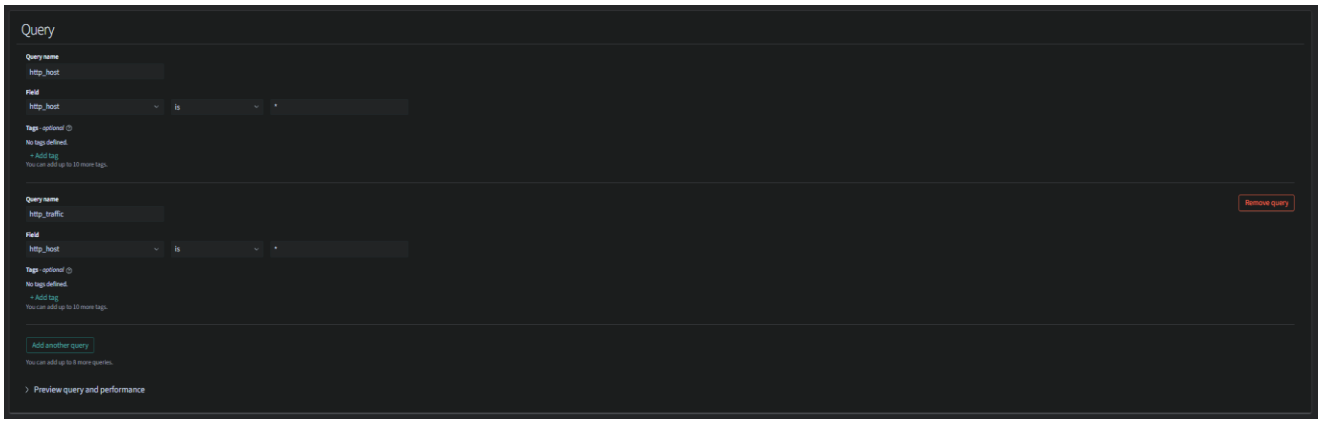
14. Изучим OpenSearch Alerting.

Из-за некоторых ограничений нашей версии контейнера, я попытался настроить простой монитор который будет срабатывать если есть ЛЮБОЕ количество документов

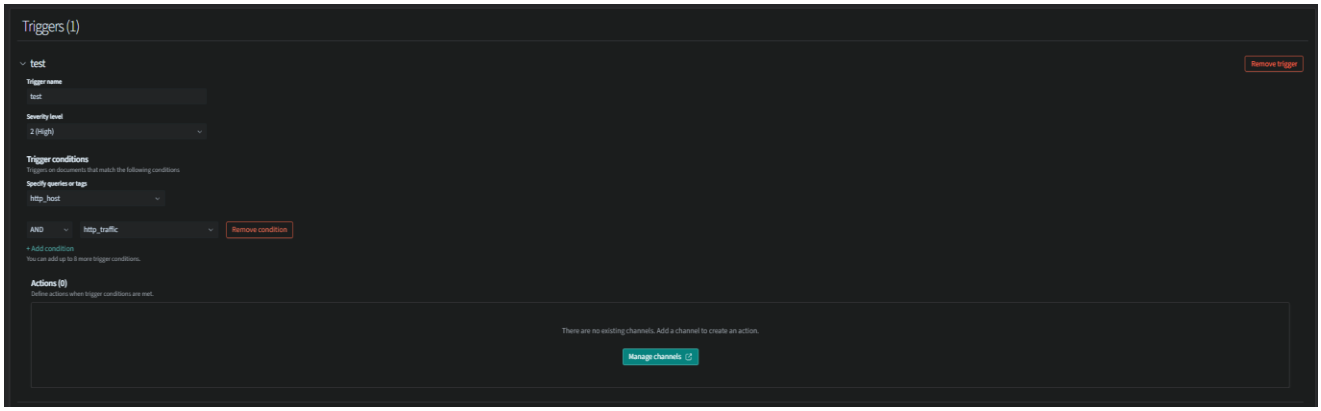
Начальная настройка:



Создаем Query. Из-за ограничений платформы для нашего сценария создадим два Query с одинаковыми условиями:



Создаем тригер:
AND http_traffic = условие "ЕСЛИ есть документы, соответствующие запросу http_traffic", что эквивалентно "Сработать, если обнаружен HTTP трафик".



Таким образом наш тригер будет срабатывать когда какой-либо http трафик появится. Однако из-за ограничений api в нашей реализации контейнера большинство функционала недоступно, и по большей части из-за выключенной безопасности.

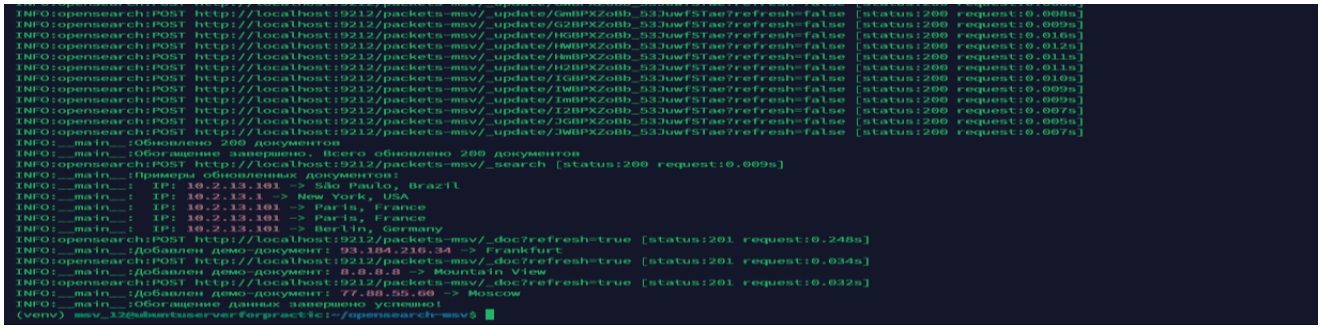
15. Изучим возможности OpenSearch Dashboards по анализу гео-данных

В наших данных нет геометрической информации, поэтому для работы с гео-данными потребуется обогащение исходных данных. Для этого обычно используются сервисы по типу MaxMind GeoIP2 и IP2Location. Однако в целях изучения мы обогатим наши исходные данные вручную “выдуманными” данными, для чего будем использовать Python скрипт:
https://github.com/l1ratch/WC_BISO/blob/main/5_sem/ТХвСК/practic/pr_5/ewgd.py

Для обогащения данных был разработан Python-скрипт, который выполняет следующие операции:

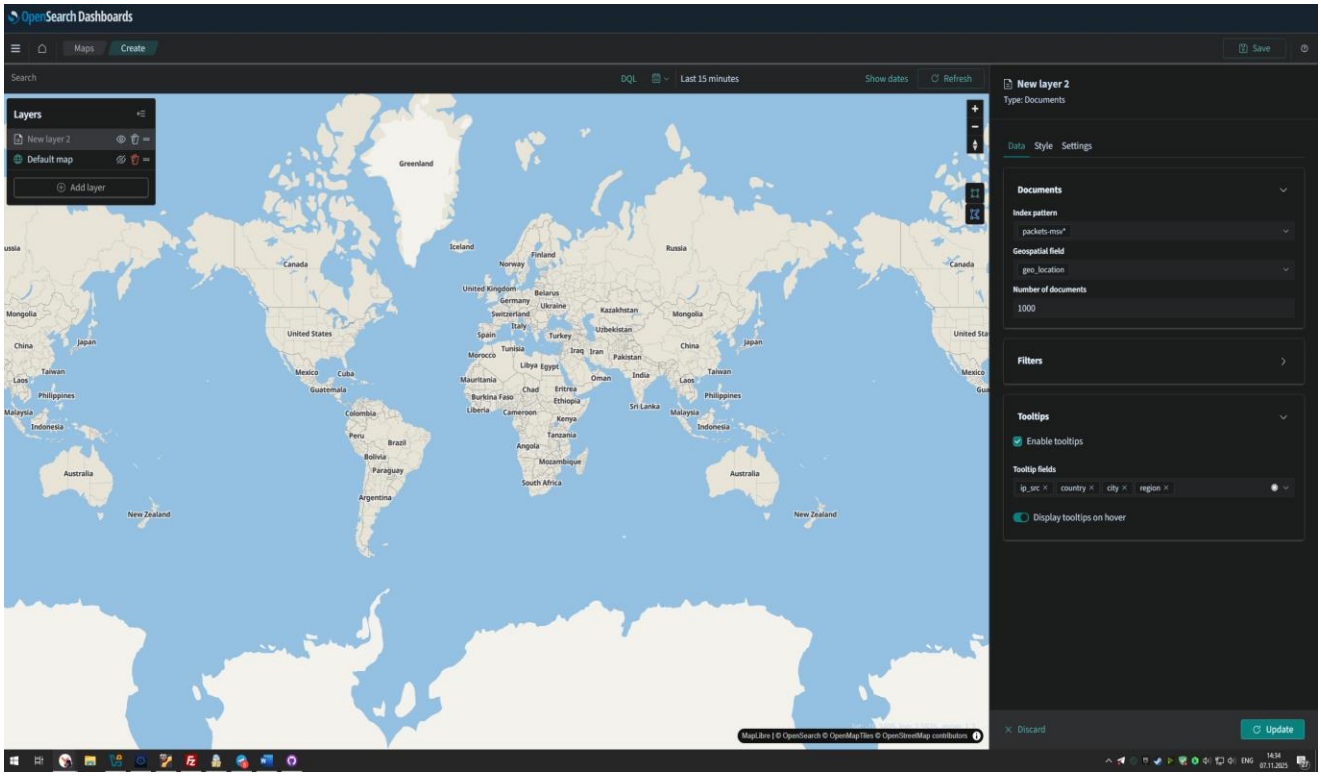
1. Добавление полей в маппинг индекса OpenSearch для хранения геоданных:
   geo_location (тип geo_point) - координаты широты/долготы;
   country - страна расположения;
   city – город;
   region - географический регион.
2. Генерация демонстрационных данных на основе предопределенного списка городов мира, включая:
   Столицы и крупные города (Нью-Йорк, Лондон, Токио и др.);
   Различные географические регионы (Северная Америка, Европа, Азия и т.д.);
   Реалистичные координаты для каждого местоположения.
3. Обновление документов в индексе packets-msv с присвоением случайных географических меток существующим записям сетевого трафика.

Приступим к обогащению:

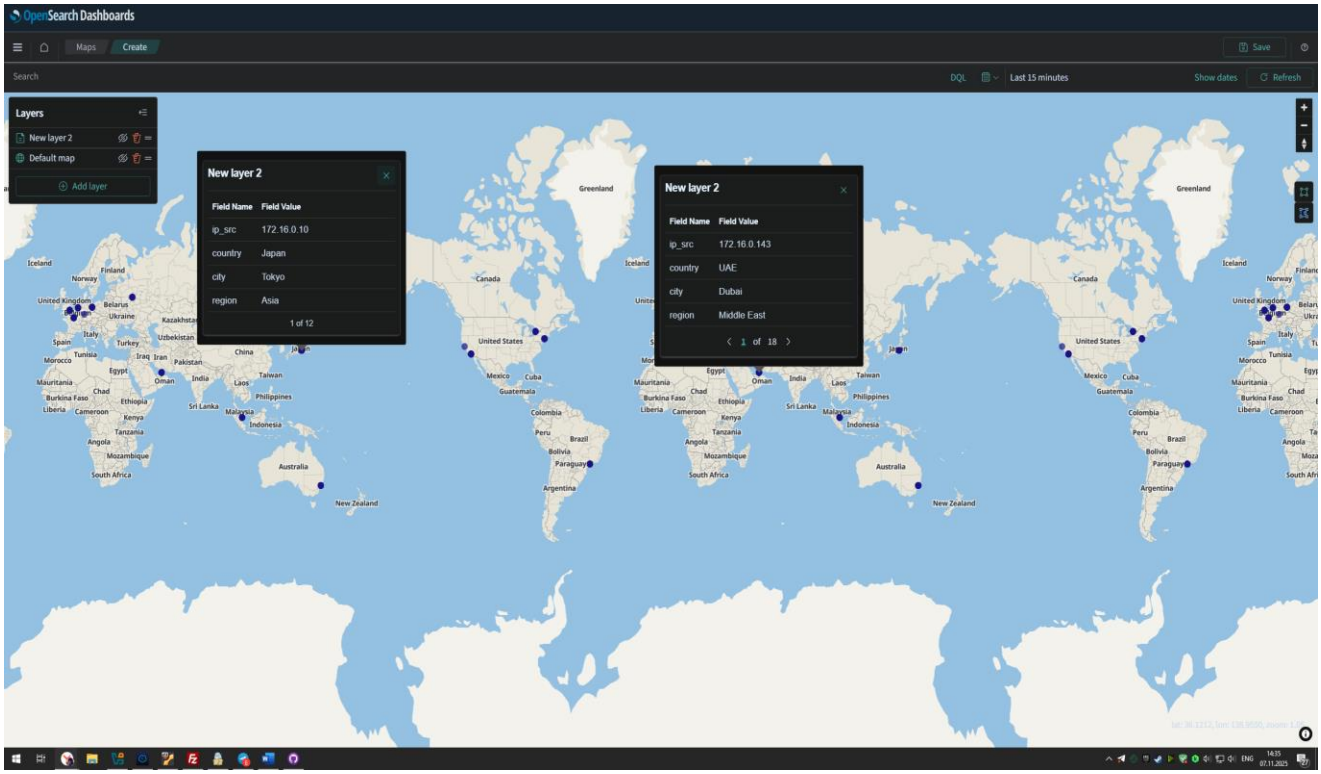


Проверим добавление данных:

Открываем в визуализациях карту и добавляем новый слой:



Получаем карту с точками отображающие географические локации добавленные нами и связанные с данными из исходных файлов:



В этом пункте мы обновили документы сетевого трафика путем добавления географических меток. Для обогащения данных использовался Python-скрипт, который присвоил случайные географические локации из предопределенного списка, включающего 14 городов из различных регионов мира.

16. Заключение

В ходе выполнения практической работы были получены навыки развертывания и администрирования контейнеризированных приложений с использованием Docker и Docker Compose. Успешно развернута система OpenSearch и панель OpenSearch Dashboards, произведена загрузка и анализ сетевого трафика, а также визуализация результатов.

С помощью инструмента TShark выполнена предварительная обработка PCAP-файлов, после чего данные были загружены в OpenSearch для дальнейшего анализа. Исследование логов показало характерные особенности сетевой активности, включая использование протоколов HTTP, DNS и защищённых соединений HTTPS. Были выявлены возможные признаки автоматизированных запросов и работы локального DNS-сервера.

Кроме того, реализовано обогащение данных географическими метками, что позволило расширить возможности аналитики и визуализации. Также изучены базовые принципы настройки мониторинга и оповещений (Alerting) в OpenSearch, что является важным элементом систем информационной безопасности.

Таким образом, в результате проделанной работы достигнута цель — освоены инструменты для работы с частично структурированными данными и получения аналитической информации на основе сетевого трафика с использованием современных технологий контейнеризации и анализа данных.