

Φ-APISEC

FAQ

SΦUAD

www.squadsec.ru

⊕ Область применения

⊕ Nginx

- Как развернуть Q-APISec, если используется Nginx?
- Как настроить проксирование в Nginx?
- Как настроить роуты в модуле интеграции envoy-alc при проксировании через Nginx?
- Как запустить Q-APISec с проксированием через Nginx?
- Как остановить Q-APISec с проксированием через Nginx?
- Как настроить TLS между клиентами и защищаемым сервисом при проксировании через Nginx?

⊕ Envoy

- Как развернуть Q-APISec, если используется Envoy?
- Как настроить проксирование в Envoy?
- Как настроить роуты в модуле интеграции envoy-alc при проксировании через Envoy?
- Как запустить Q-APISec с проксированием через Envoy?
- Как остановить Q-APISec с проксированием через Envoy?
- Как настроить TLS между клиентами и защищаемым сервисом при проксировании через Envoy?

⊕ Зеркалирование

- Как развернуть Q-APISec, если используется http-sniffer?
- Как запустить Q-APISec с зеркалированием через http-sniffer?
- Как остановить Q-APISec с зеркалированием через http-sniffer?
- Как настроить роуты в модуле интеграции envoy-alc при зеркалировании через http-snifferx?

⊕ Общие настройки

- Как клонировать репозиторий?
- Как обновить настройки роутов и анализаторов?
- Как указать набор анализаторов, которые влияют на блокировку?
- Как включить режим блокировки?
- Как включить прозрачный режим?
- Как включить/выключить анализатор?
- Как включить/выключить правило сигнатурного анализатора?
- Как добавить своё правило в сигнатурный анализатор?
- Как настроить правило сваггер анализатора?
- Как настроить правило статистического анализатора?
- Как настроить правило анализатора чувствительных данных?

⊕ Как настроить отправку алертов в формате CEF в syslog?

- Пример:
- Дополнение: при необходимости быстрой проверки можно воспользоваться тестовым образом Splunk (после запуска Q-APISEC):

Область применения

Этот FAQ создан для демо-дистрибутива, когда решение поставляется в контейнерах docker.

Nginx

Как развернуть Q-APISec, если используется Nginx?

1. Указать адреса и порты в [директивах Nginx](#).
2. [Запустить](#) Q-APISec.
3. [Клонировать](#) репозиторий.
4. [Указать](#) роуты, для которых будут действовать правила.
5. Настроить режим работы для роутов ([прозрачный](#) или [блокирующий](#)).
6. Настроить правила для анализаторов ([сваггер](#) анализатор, [статистический](#) анализатор, анализатор [чувствительных данных](#)).
7. Отправить запрос на защищаемый сервис через Nginx.
8. Найти запрос на странице События в панели оператора.

Как настроить проксирование в Nginx?

1. Открыть *дистрибутив\infra\nginx\routing.conf*
2. Указать адрес защищаемого сервиса в директиве [upstream](#). Пример:
upstream ecco_service {
server host.docker.internal:6065;
}
3. Указать в директиве [server](#) порт, который прослушивает nginx. Указать в директиве, куда перенаправлять запросы в параметре [proxy_pass](#). В примере показано проксирование в один и тот же сервис напрямую по порту и по пути в начале URL:

```
server {
    listen 10000;
    chunked_transfer_encoding off;
    location ^~ /direct/ {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_pass http://ecco_service;
    }
    location ^~ /ecco/ {
        on_request ecco host.docker.internal:9020 blocked;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_pass http://ecco_service;
    }
}
```

После `on_request` указано имя роута ecco. В конце этой же строки `blocked` - поддержка блокировок envoy-als. Для переключения в прозрачный режим нужно указать `nonblocked`, настройки блокировок в envoy-als при этом будут

игнорироваться.

4. В `дистрибутив\infra\nginx\docker-compose.yaml` убедиться, что порт, указанный в пункте 2, открыт. Пример:

```
ports:
  - 10000:10000
```

5. Выполнить **docker compose up -d** в `дистрибутив\infra\nginx\` (если Nginx был запущен до этого, то сначала выполнить **docker compose down**).

Как настроить роуты в модуле интеграции envoy-alc при проксировании через Nginx?

1. [Клонировать](#) репозиторий
2. Открыть `директория_куда_клонирован_репозиторий\rules-config\envoy-alc-testcontainers.yml`.
3. Добавить роут в `application.routes.id`. Пример:
 - `id: "ессо"` # имя должно совпадать с именем роута в директиве `server`. В директиве имя роута указано в начале строки `on_request` `имя_роута`
4. [Обновить](#) настройки.

Как запустить Q-APISec с проксированием через Nginx?

Выполнить в корне дистрибутива

```
docker compose -f ./infra/nginx/docker-compose.yaml up -d && docker compose -f
./infra/docker-compose.yml up -d && docker compose -f ./services/docker-compose.yml up -d
&& docker ps
```

Как остановить Q-APISec с проксированием через Nginx?

Выполнить в корне дистрибутива

```
docker compose -f ./infra/nginx/docker-compose.yaml down && docker compose -f
./services/docker-compose.yml down && docker compose -f ./infra/docker-compose.yml down
&& docker ps
```

Как настроить TLS между клиентами и защищаемым сервисом при проксировании через Nginx?

1. Открыть файл `дистрибутив\infra\nginx\docker-compose.yaml`.
2. Добавить в блок `services.nginx.volumes` ключи и сертификаты в формате "откуда:куда". Пример:
 - `./keystore/test-rest-service-ssl.crt:/etc/ssl/certs/test-rest-service-ssl.crt`
 - `./keystore/rootCA.crt:/etc/ssl/certs/rootCA.crt`
 - `./keystore/test-rest-ssl.key:/etc/ssl/certs/test-rest-ssl.key`
3. Добавить в блок `services.nginx.ports` порт, на котором Nginx будет принимать защищённые соединения.

Пример:

```
ports:
  - 10003:10003
```

4. Открыть файл `дистрибутив\infra\nginx\routing.conf`.
5. В директиве `upstream` для защищаемого сервиса указать порт, который используется для HTTPS/ Пример:

```
upstream test-rest-service-ssl {
  server host.docker.internal:6443 ;
}
```
6. В директиве `server` для защищаемого сервиса указать использование TLS.
Пример:

```
server {
  listen 10003 ssl; # Установка HTTPS для пары пользователь – Nginx
  server_name test-rest-service-ssl; # Доменное имя для доступа к приложению
  # Ниже указаны сертификат приложения и закрытый ключ для него
  ssl_certificate /etc/ssl/certs/test-rest-service-ssl.crt;
  ssl_certificate_key /etc/ssl/certs/test-rest-ssl.key;

  chunked_transfer_encoding off;
  location ^~ /test-rest-service-ssl/ {
    on_request test-rest-service-ssl host.docker.internal:9020 blocked;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header Host $http_host;
    proxy_redirect off;
    proxy_pass https://ecco_service; # Установка HTTPS для пары Nginx –
защищаемый сервис
    proxy_ssl_trusted_certificate /etc/ssl/certs/rootCA.crt; # Путь к корневому
сертификату, который удостоверяет сертификат приложения
    proxy_ssl_verify on;

    # Если требуется клиентский сертификат
    proxy_ssl_certificate /etc/ssl/certs/test-rest-ssl.crt.pem; # Путь к сертификату
    proxy_ssl_certificate_key /etc/ssl/certs/test-rest-ssl.key.pem; # Путь к ключу
  }
}
```

7. Выполнить **`docker compose up -d`** в `дистрибутив\infra\nginx\` (если Nginx был запущен до этого, то сначала выполнить **`docker compose down`**).

Envoy

Как развернуть Q-APISec, если используется Envoy?

1. Указать адреса и порты в директивах [Envoy](#).
2. [Запустить](#) Q-APISec.
3. [Клонировать](#) репозиторий.
4. [Указать](#) роуты, для которых будут действовать правила.
5. Настроить режим работы для роутов ([прозрачный](#) или [блокирующий](#)).
6. Настроить правила для анализаторов ([сваггер](#) анализатор, [статистический](#) анализатор, анализатор [чувствительных данных](#)).
7. Отправить запрос на защищаемый сервис через Envoy.

8. Найти запрос на странице События в панели оператора.

Как настроить проксирование в Envoy?

1. Открыть *дистрибутив\infra\envoy\envoy.yaml*
2. Указать интерфейс и порт в static_resources.listeners.address.socket_address (откуда перенаправлять запросы?). Пример:
 socket_address:
 address: 0.0.0.0 # Здесь порт прослушивается на всех интерфейсах
 port_value: 10000 # Envoy слушает этот порт
3. В static_resources.listeners.filter_chains.filters.config.route_config.virtual_hosts.routes.route.cluster (куда отправлять запросы?) указать имя кластера.
 Пример:
 cluster: backend_service
4. Указать имя кластера из пункта 3 в:
 1. static_resources.clusters.name
 2. static_resources.clusters.load_assignment.cluster_name
5. В static_resources.clusters.load_assignment.endpoints.lb_endpoints.endpoint.address.socket_address указать адрес защищаемого сервиса и порт, через который он доступен. Пример:
 socket_address:
 address: host.docker.internal
 port_value: 6060
6. В *дистрибутив\infra\envoy\docker-compose.yaml* убедиться, что порт, указанный в пункте 2, открыт. Пример:
 ports:
 - 10000:10000
7. Выполнить **docker compose up -d** в *дистрибутив\infra\envoy* (если envoy был запущен до этого, то сначала выполнить **docker compose down**).

Как настроить роуты в модуле интеграции envoy-alc при проксировании через Envoy?

1. [Клонировать](#) репозиторий
2. Открыть *директория_куда_клонирован_репозиторий\rules-config\envoy-alc-testcontainers.yml*.
3. Добавить роут в application.routes. Пример:
 - id: "ессо" # Произвольное имя, какое удобно
 envoy-config:
 host: "0.0.0.0" #Значение
 static_resources.listeners.address.socket_address.address из файла
 дистрибутив\infra\envoy\envoy.yaml *дистрибутив\infra\envoy\envoy.yaml*
 port: 10001 #Значение static_resources.listeners.address.socket_address.port из
 файла *дистрибутив\infra\envoy\envoy.yaml*
4. [Обновить](#) настройки.

Как запустить Q-APISec с проксированием через Envoy?

Выполнить в корне дистрибутива

```
docker compose -f ./infra/envoy/docker-compose.yaml up -d && docker compose -f
./infra/docker-compose.yml up -d && docker compose -f ./services/docker-compose.yml up -d
```

&& docker ps

Как остановить Q-APISec с проксированием через Envoy?

Выполнить в корне дистрибутива

```
docker compose -f ./infra/envoy/docker-compose.yaml down && docker compose -f
./services/docker-compose.yml down && docker compose -f ./infra/docker-compose.yml down
&& docker ps
```

Как настроить TLS между клиентами и защищаемым сервисом при проксировании через Envoy?

1. Открыть файл *дистрибутив\infra\envoy\docker-compose.yaml*.
2. Добавить в блок `services.envoy.volumes` ключи и сертификаты в формате "откуда:куда". Пример:
 - ./keystore/test-rest-service-ssl.crt:/etc/ssl/certs/test-rest-service-ssl.crt
 - ./keystore/test-rest-ssl.crt.pem:/etc/ssl/certs/test-rest-ssl.crt.pem
 - ./keystore/test-rest-ssl.key.pem:/etc/ssl/certs/test-rest-ssl.key.pem
3. Открыть файл *дистрибутив\infra\envoy\envoy.yaml*.
4. **Добавить** в раздел `static_resources.listeners.filter_chains.transport_socket` для защищаемого сервиса блок, отвечающий за TLS. В блоке указать путь к сертификату и ключу (часть "куда" из пункта 2). Пример:


```
transport_socket:
  name: envoy.transport_sockets.tls
  typed_config:
    "@type":
      type.googleapis.com/envoy.extensions.transport_sockets.tls.v3.DownstreamTlsContext
    common_tls_context:
      tls_certificates:
        - certificate_chain:
            filename: /etc/ssl/certs/test-rest-ssl.crt.pem # Путь к сертификату
          private_key:
            filename: /etc/ssl/certs/test-rest-ssl.key.pem # Путь к ключу
```
5. Добавить в раздел `static_resources.clusters.transport_socket` защищаемого сервиса блок, отвечающий за TLS. В блоке указать путь к сертификату (часть "куда" из пункта 2). Пример:


```
transport_socket:
  name: envoy.transport_sockets.tls
  typed_config:
    "@type":
      type.googleapis.com/envoy.extensions.transport_sockets.tls.v3.UpstreamTlsContext
    allow_renegotiation: false
    common_tls_context:
      validation_context:
        trusted_ca:
          filename: /etc/ssl/certs/test-rest-service-ssl.crt # Путь к сертификату
```
6. Выполнить **docker compose up -d** в *дистрибутив\infra\envoy* (если Envoy был запущен до этого, то сначала выполнить **docker compose down**).

Зеркалирование

Как развернуть Q-APISec, если используется http-sniffer?

1. Указать порты в *дистрибутив/infra/http-sniffer/config.properties*.
2. Настроить интерфейс, на который подается копия трафика:
дистрибутив/httpsnf_install.sh <имя_интерфейса>
3. [Запустить](#) Q-APISec включая http-sniffer.
4. [Клонировать](#) репозиторий.
5. [Указать](#) роуты, для которых будут действовать правила.
6. Настроить правила для анализаторов ([сваггер](#) анализатор, [статистический](#) анализатор, анализатор [чувствительных данных](#)).
7. Отправить запрос на защищаемый сервис, трафик которого зеркалируется.
8. Найти запрос на странице События в панели оператора.

Как запустить Q-APISec с зеркалированием через http-sniffer?

Запустить сниффер как сервис:
systemctl start httpsnf

Выполнить в корне дистрибутива
docker compose -f ./infra/docker-compose.yml up -d && docker compose -f ./services/docker-compose.yml up -d && docker ps

Как остановить Q-APISec с зеркалированием через http-sniffer?

Остановить сниффер как сервис:
systemctl stop httpsnf

Выполнить в корне дистрибутива
docker compose -f ./services/docker-compose.yml down && docker compose -f ./infra/docker-compose.yml down && docker ps

Как настроить роуты в модуле интеграции envoy-alc при зеркалировании через http-snifferx?

1. [Клонировать](#) репозиторий
2. Открыть *директория_куда_клонирован_репозиторий/rules-config/envoy-alc-testcontainers.yml*.
3. Добавить роут в *application.routes.id* и роут с URL паттерном в *application.kafka-routes.id*.
Пример:
routes:
- id: "juice-shop"
kafka-routes:
- id: "juice-shop"

- path-pattern: "^/juice/"
4. [Обновить](#) настройки.

Общие настройки

Как клонировать репозиторий?

В любом месте выполнить

```
git clone ssh://git@localhost:2222/srv/git/rules-config.git
```

Пароль 12345. В месте выполнения не должно быть директории *rules-config*.

Как обновить настройки роутов и анализаторов?

1. Выполнить в *rules-config*

```
git add . && git commit -m updated && git push
```

 Пароль 12345. В описании коммита (после -m можно указать краткое описание изменений).
2. Дождаться применения настроек. Для изменений в *envoy-alc-testcontainers.yml* настройки применяются в течение времени, указанного в *application.config-refresh-rate-sec* этого же файла. В случае добавления сигнатуры нужно перезапустить *envoy-alc* с помощью **docker restart envoy-alc**
 Для применения изменений в *http-request-logger-testcontainers.yml* нужно перезапустить модуль *srv-request-logger* с помощью **docker restart srv-request-logger**.

Если используется GitHub, GitLab или подобный продукт, то настройки можно обновить, запусив изменения в репозиторий через IDE или web-интерфейс.

Как указать набор анализаторов, которые влияют на блокировку?

1. Открыть *директория_куда_клонирован_репозиторий/rules-config/envoy-alc-testcontainers.yml*.
2. Добавить в *application.analyzers.id* анализаторы. Пример:

```
analyzers:
  - id: swagger-analyzer
  - id: signature-analyzer
  - id: statistic-analyzer
  - id: class-analyzer
```

 Запрос не будет заблокирован, если анализатор выдал FAIL для запроса, но анализатора нет в списке.
3. [Обновить](#) настройки.

Как включить режим блокировки?

1. Открыть *директория_куда_клонирован_репозиторий/rules-config/envoy-alc-testcontainers.yml*.

2. Установить значение "BLOCKED" для:

1. application.mode, чтобы все роуты работали в режиме блокировки.
Отдельный роут может работать в прозрачном режиме, если для него явно указан режим.
2. application.routes.mode, чтобы отдельный роут работал в режиме блокировки. Пример:

```
routes:
  - id: "demo-service"
    mode: "BLOCKED"
```

 Роут будет игнорировать общую настройку режима работы, если она отличается от его режима.
3. application.routes.analyzers.mode, чтобы выбранный анализатор работал в режиме блокировки. Пример:

```
routes:
  - id: "demo-service"
    mode: "TRANSPARENT"
  analyzers:
    - id: signature-analyzer
      mode: "BLOCKED"
```

 Анализатор будет игнорировать общую настройку режима работы и режим работы роута, если они отличаются от его режима.
4. [Обновить](#) настройки.

Как включить прозрачный режим?

1. Открыть *директория_куда_клонирован_репозиторий\rules-config\envoy-alc-testcontainers.yml*.
2. Установить значение "TRANSPARENT" для:
 1. application.mode, чтобы все роуты работали в режиме блокировки.
Отдельный роут может работать в прозрачном режиме, если для него явно указан режим.
 2. application.routes.mode, чтобы отдельный роут работал в режиме блокировки. Пример:

```
routes:
  - id: "demo-service"
    mode: "TRANSPARENT"
```

 Роут будет игнорировать общую настройку режима работы, если она отличается от его режима.
 3. application.routes.analyzers.mode, чтобы выбранный анализатор работал в режиме блокировки. Пример:

```
routes:
  - id: "demo-service"
    mode: "BLOCKED"
  analyzers:
    - id: signature-analyzer
      mode: "TRANSPARENT"
```

 Анализатор будет игнорировать общую настройку режима работы и режим работы роута, если они отличаются от его режима.
 4. [Обновить](#) настройки.

Как включить/выключить анализатор?

1. Открыть `директория_куда_клонирован_репозиторий\rules-config\envoy-alc-testcontainers.yml`.
2. Установить значение true/false в:
 1. `application.ml-gateway.analyzer.class.enabled` для анализатора чувствительных данных.
 2. `application.signature-analyzer.enabled` для сигнатурного анализатора.
 3. `application.swagger-analyzer.enabled` для сваггер анализатора.
 4. `application.statistic-analyzer.enabled` для статистического анализатора.
3. [Обновить](#) настройки.

Как включить/выключить правило сигнатурного анализатора?

1. Найти имя файла с правилом. Например, оно указано в деталях сигнатурного анализатора для события.
2. Открыть `директория_куда_клонирован_репозиторий\rules-config\rules\config\_rule_status_list.json`.
3. Найти имя файла с правилом в списке
4. Установить значение true/false для правила. Пример:


```
{
  "id": "ldap-injection",
  "enabled": false
}
```
5. [Обновить](#) настройки.

Как добавить своё правило в сигнатурный анализатор?

1. Создать `rule_name.json` в `директория_куда_клонирован_репозиторий\rules-config\rules\`
2. Указать проверяемые методы, части запроса и регулярное выражение. Пример:


```
{
  "order": 1,
  "description": "SSTI payloads exploitation",
  "method": [
    "POST",
    "PUT",
    "PATCH",
    "DELETE",
    "GET"
  ],
  "scope": "INOUT",
  "join": "AND",
  "rules": [
    {
      "id": 1,
      "target": [
        "BODY",
```

```

    "URI"
  ],
  "signature": "regex"
}
]
}

```

"\" в поле сигнатуре должен быть экранирован: \ -> \\\.

3. Добавить имя правила и режим работы в *директория_куда_клонирован_репозиторий\rules-config\rules\config_rule_status_list.json*. Пример:

```

{
  "id": "rule_name",
  "enabled": true
}

```

4. [Обновить](#) настройки.

Как настроить правило сваггер анализатора?

1. Скопировать файл схемы в *директория_куда_клонирован_репозиторий\rules-config\schemas*.
2. Открыть *директория_куда_клонирован_репозиторий\rules-config\envoy-alc-testcontainers.yml*.
3. Указать роут в [application.swagger-analyzer.schema.routes.id](#).
4. Указать имя файла из пункта 1 в [application.swagger-analyzer.schema.routes.file](#).
5. [Обновить](#) настройки.

Пример:

```

swagger-analyzer:
  enabled: true
  schema:
    reload-period-sec: 10
  routes:
    - id: demo-service-gw
      file: TestSchemas.json

```

Как настроить правило статистического анализатора?

1. Открыть *директория_куда_клонирован_репозиторий\rules-config\envoy-alc-testcontainers.yml*.
2. Указать имя правила в [application.statistic-analyzer.rules.id](#).
3. Указать роут в [application.statistic-analyzer.rules.route](#).
4. Указать предельное количество запросов за период в [application.statistic-analyzer.rules.count](#).
5. Указать период, за который считается количество запросов, в [application.statistic-analyzer.rules.duration](#).
6. Указать режим работы анализатора в [application.statistic-analyzer.rules.group-mode](#).
7. [Обновить](#) настройки.

Пример:

```

statistic-analyzer:
  enabled: true
  rules:
    - id: test1
      route: ecco
      count: 2
      duration: PT5S
      group-mode: ALL

```

Как настроить правило анализатора чувствительных данных?

1. Открыть *директория_куда_клонирован_репозиторий*\rules-config\envoy-alc-testcontainers.yml.
2. Указать роут в application.routes.analyzer.class.routes.id.
3. Указать разрешённые к передаче категории в application.routes.analyzer.class.routes.data-class. Если категория отсутствует в списке, то она запрещена к передаче. Если нужно запретить все категории, то значение ключа оставить пустым
4. [Обновить](#) настройки.

Пример:

```

analyzer:
  max-connections: 4
  class:
    enabled: true
    url: "http://srv-ml-class-analyzer:3000"
    routes:
      - id: "ecco"
        data-class: LOCATION, RU_BANK_ACCOUNT_NUMBER, CREDIT_CARD, EMAIL_ADDRESS,
        FIO, INN, PASSPORT, PHONE_NUMBER, SNILS
        # data-class: # Запросы с любой категорией чувствительных данных будут
        блокироваться

```

Как настроить отправку алертов в формате CEF в syslog?

1. Открыть *директория_куда_клонирован_репозиторий*\rules-config\http-request-logger-testcontainers.yml.
2. Указать в inventory.alerts.analyzers список анализаторов, которые будут инициировать отправку алерта.
3. Указать в inventory.alerts.targets.type значение cef.
4. Указать в inventory.alerts.targets.properties.format значение full.
5. Указать в inventory.alerts.targets.properties.transport протокол передачи (udp или tcp).
6. Указать в inventory.alerts.targets.properties.hostname адрес принимающего алерт сервиса.
7. Указать в inventory.alerts.targets.properties.port порт принимающего алерт сервиса.
8. Указать в inventory.alerts.rules.route роут, для которого должны инициироваться

- алерты.
9. Указать в `inventory.alerts.rules.status` при каких результатах проверки запроса будет отправляться алерт.
 10. Указать в `inventory.alerts.rules.status_condition` условие отправки алерта. OR, чтобы отправлять алерт при сработке хотя бы одного анализатора из пункта 2. AND, чтобы отправлять алерт при сработке только всех анализаторов из пункта 2
 11. Указать в `inventory.alerts.rules.target` значение из шага 3.
 12. [Обновить](#) настройки.

Пример:

```
alerts:
analyzers:
- signature-analyzer
- swagger-analyzer
- statistic-analyzer
waitRequestCheck: 5000
targets:
- type: cef
  properties:
    format: full
    transport: udp
    hostname: splunk
    port: 514
rules:
- route: ecco
  status: FAIL
  status_condition: OR
  target: cef
```

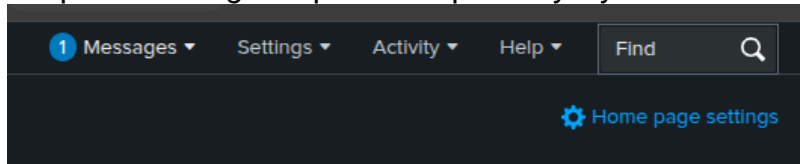
Дополнение: при необходимости быстрой проверки можно воспользоваться тестовым образом Splunk (после запуска Q-APISEC):

```
services:
splunk:
  image: splunk/splunk:latest
  container_name: splunk
  restart: always
  environment:
    - SPLUNK_PASSWORD=YourSplunkPassword # Замените на надёжный пароль
    - SPLUNK_GENERAL_TERMS=--accept-sgt-current-at-splunk-com
    - SPLUNK_START_ARGS=--accept-license
    - SPLUNK_ACCEPT_LICENSE=yes
  ports:
    - "8000:8000" # Веб-интерфейс Splunk
    - "514:514/udp" # Прием syslog-сообщений
  cap_add:
    - NET_ADMIN # Необходим для привязки порта 514/udp
  tmpfs:
    - /opt/splunk/var/run/splunk:size=6g # Монтируем tmpfs с размером 6 ГБ
  networks:
    - backend
```

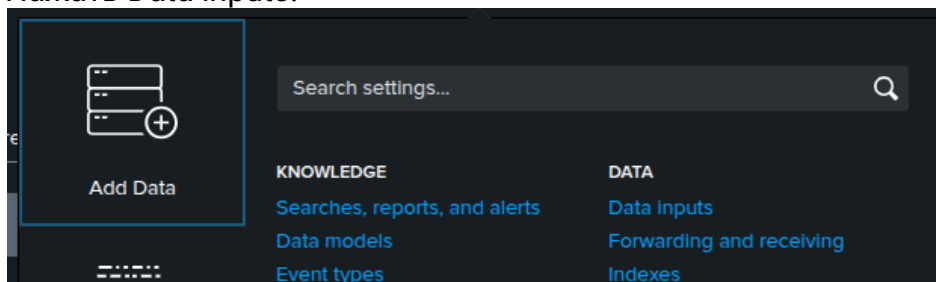
networks:

backend:
external: true

1. Скопировать компоуз файл выше в отдельный файл.
2. В папке, где хранится файл, выполнить **docker compose up -d**.
3. Для настройки Splunk на получение алертов открыть localhost:8000.
4. Открыть Settings в правом верхнем углу.



5. Нажать Data inputs.



6. Нажать + Add new в строке для UDP.

Local inputs		
Type	Inputs	Actions
Files & Directories Index a local file or monitor an entire directory.	19	+ Add new
HTTP Event Collector Receive data over HTTP or HTTPS.	0	+ Add new
TCP Listen on a TCP port for incoming data, e.g. syslog.	0	+ Add new
UDP Listen on a UDP port for incoming data, e.g. syslog.	0	+ Add new

7. Указать порт в поле Port и протокол передачи (должен совпадать с выбранным в настройках srv-request-logger), нажать кнопку Next.

Source

Input Settings

Review

Done

< Back

Next >

Configure this instance to listen on any TCP or UDP port to capture data sent over the network (such as syslog). [Learn More](#)

TCP

UDP

Port ? 514
Example: 514

Source name override ? optional
host:port

Only accept connection from ? optional
example: 10.1.2.3, lbadhost.splunk.com, *.splunk.com

FAQ

8. В выпадающем списке Select Source Type выбрать syslog (в точности, можно воспользоваться поиском по тексту) и нажать кнопку Review.

Add Data | Select Source | **Input Settings** | Review | Done | < Back | Review >

Input Settings

Optionally set additional input parameters for this data input as follows:

Source type

The source type is one of the default fields that the Splunk platform assigns to all incoming data. It tells the Splunk platform what kind of data you've got, so that the Splunk platform can format the data intelligently during indexing. And it's a way to categorize your data, so that you can search it easily.

App context

Application contexts are folders within a Splunk platform instance that contain configurations for a specific use case or domain of data. App contexts improve manageability of input and source type definitions. The Splunk platform loads all app contexts based on precedence rules. [Learn More](#)

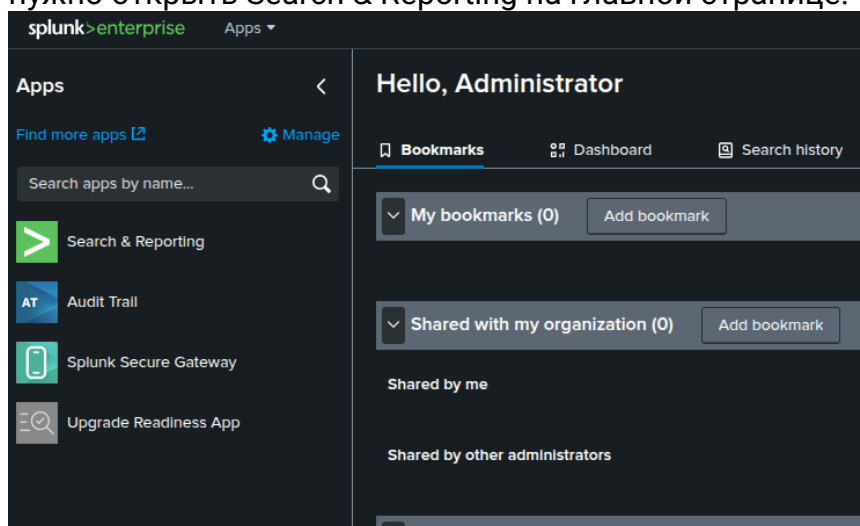
Host

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More](#)

Source type dropdown:

- postfix_syslog
Output produced by the Postfix email server
- sendmail_syslog
Output produced by the Sendmail email server
- syslog**
Output produced by many syslog daemons, as described in RFC3164 by the IETF
- windows_snare_syslog
Output produced by the Snare syslog server on Windows

9. Нажать кнопку Submit.
10. Нажать кнопку Start Searching. Новые алерты будут появляться при обновлении данных.
11. Чтобы снова открыть поиск алертов (если страница была закрыта) в Splunk нужно открыть Search & Reporting на главной странице.



12. Ввести source="udp:514" sourcetype="syslog" (номер порта соответствует примеру) в поле поиска и выполнить поиск.

New Search | Save As | Create Table View | Close

enter search here... | Last 24 hours | Search

Пример сообщения:

```
Jun 06 09:05:40 4efe3122f66d CEF:0|SQUAD|Q-APISec|1|3|request
fail|High|msg=[{"analyzer_name":"statistic-
analyzer","check_status":"OK","check_date":"2025.06.06
09:05:18"},{"analyzer_name":"swagger-
analyzer","check_status":"FAIL","check_date":"2025.06.06
```

```

09:05:18"},"{"analyzer_name":"signature-
analyzer","check_status":"FAIL","check_date":"2025.06.06
09:05:18"},"{"analyzer_name":"class-
analyzer","check_status":"OK","check_date":"2025.06.06 09:05:18"}] src=1.1.1.1 spt=0
dst=172.17.0.1 dpt=6065 destinationServiceName=ecco
requestUrl=/hi?name\\=ade87c26-fe17-4615-a960-bfd527d9e1db&qwe\\=12
requestMethod=GET externalId=c7103c25-da14-456d-a76d-6fbc8236378a outcome=null
cs1={"x-request-id":"c7103c25-da14-456d-a76d-6fbc8236378a","content-length":"18","x-
forwarded-proto":"http","postman-token":"537a366a-893d-4650-9ee9-d93cecd85b51","x-
forwarded-for":"1.1.1.1,hop,
192.165.1.3","accept":"*//*","method":"GET","scheme":"http","content-
type":"text/plain","path":"/hi?name\\=ade87c26-fe17-4615-a960-
bfd527d9e1db&qwe\\=12","accept-encoding":"gzip, deflate,
br","authority":"localhost:10001","user-agent":"PostmanRuntime/7.37.3"}
cs1Label=requestHeaders cs2=bash sudo rm -rf / cs2Label=requestBody
cs3={"analyzer_name":"statistic-analyzer","check_status":"OK","check_date":"2025.06.06
09:05:18"},"{"analyzer_name":"swagger-
analyzer","check_status":"FAIL","check_date":"2025.06.06
09:05:18","meta":{"type":"swagger","schemaName":"Swagger Petstore - OpenAPI
3.0","schemaFile":"service.yml","messages":["ERROR - No API path found that matches
request '/hi': []]}},"{"analyzer_name":"signature-
analyzer","check_status":"FAIL","check_date":"2025.06.06
09:05:18","meta":{"type":"signature","signatureDescription":"Remote Command Execution:
Direct Unix Command Execution","signatureFilename":"rce-unix-shell-
3.json","signatureDetect":[{"signatureText":"bash
","signatureTarget":"BODY"}]}},{"analyzer_name":"class-
analyzer","check_status":"OK","check_date":"2025.06.06
09:05:18","meta":{"type":"ml_class","result":[]}}] cs3Label=request_analyzers

```