?????????? ? Kaspersky MDR

> Информация, приведённая на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

?????????? ? Kaspersky MDR

???????? ??????????

Интеграция с Kaspersky Managed Detection and Response позволяет автоматически импортировать инциденты из консоли MDR в KUMA. Скрипт периодически опрашивает API MDR, получает новые инциденты и создаёт соответствующие инциденты в KUMA через API.

???????????????? ??????????

1. ??????? ??????

На пограничном межсетевом экране создайте правило, разрешающее доступ к **mdr.kaspersky.com** по порту **TCP/443**.

2. ????????? ????????????? ? ?????? ? KUMA

В KUMA добавьте пользователя с ролью **Младший аналитик**, сгенерируйте токен для взаимодействия по API и разрешите доступ к следующим методам API:

- `POST /events`
- `POST /incidents/create`
- `POST /incidents/comment`
- `POST /incidents/close`

3. ????????? ?????? ? ??????? MDR

1. Перейдите в **Параметры → API**.

2. Нажмите **Добавить** и укажите **Имя соединения**.

> Это имя будет отображаться как имя пользователя при создании инцидентов, комментариев и вложений, поскольку токен доступа не привязан к конкретному пользователю.

4. В поле **Роль пользователя** выберите роль **Администратор MDR**.
5. В поле **Тенант** выберите тенант, инциденты которого необходимо импортировать в KUMA.

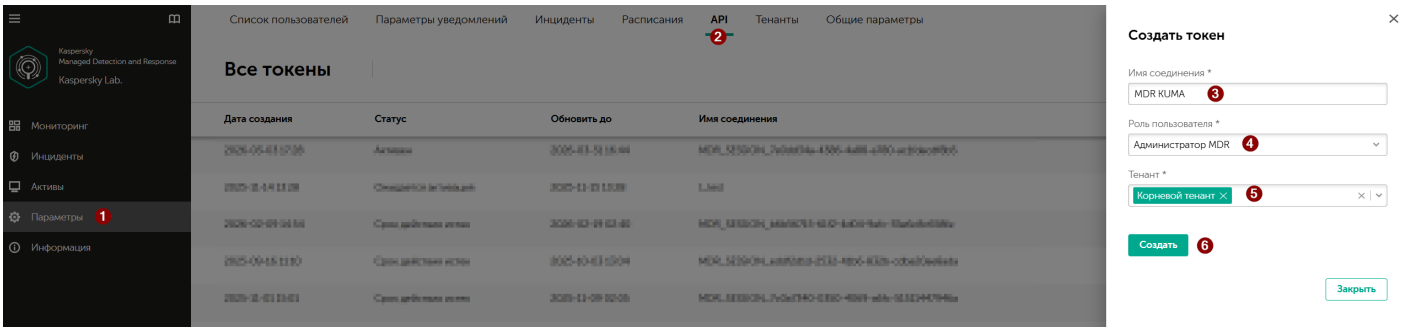
> Если для инцидентов, создаваемых в консоли MDR, в поле Тенант отсутствует значение — укажите в поле Тенант значение «Корень без тенанта», иначе скрипт не обнаружит эти инциденты.

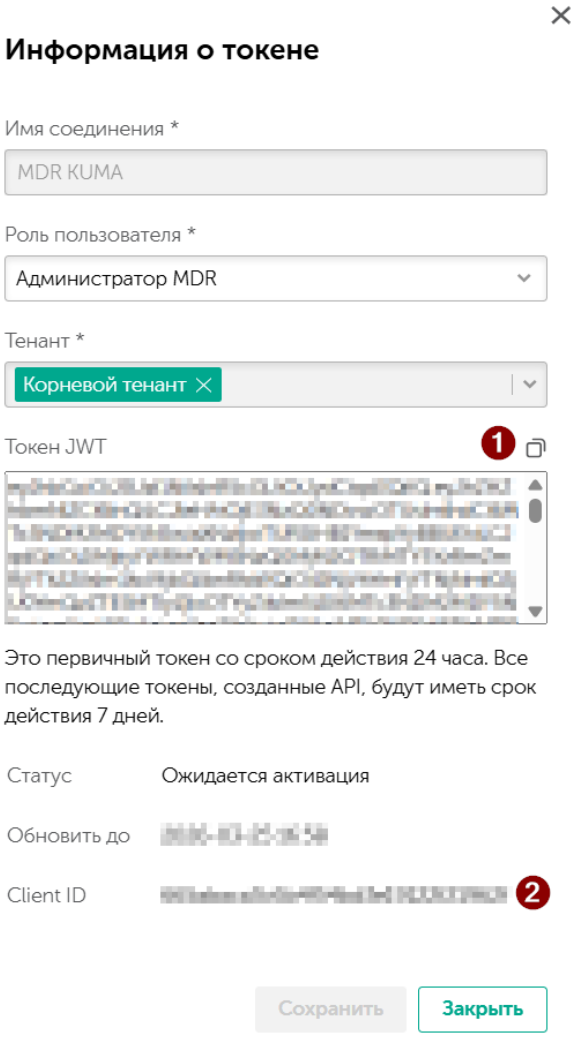
6. Нажмите **Создать**.

В результате вы получите:

- **JWT Token** (он же `refresh_token`) — это первичный токен со сроком действия 24 часа. Данный токен требуется активировать, чтобы получить новую пару `refresh_token` / `access_token`.
- **ClientID** — идентификатор клиента, который необходимо указывать при каждом запросе к API MDR.

Сохраните оба значения — они потребуются на этапе настройки.





4. ???????? ???????????

Скачайте содержимое репозитория MDR-integrations. В качестве альтернативы можно:

- Загрузить только папку **mdr_integration** через сервис https://download-directory.github.io/
- Клонировать репозиторий напрямую на сервер KUMA:

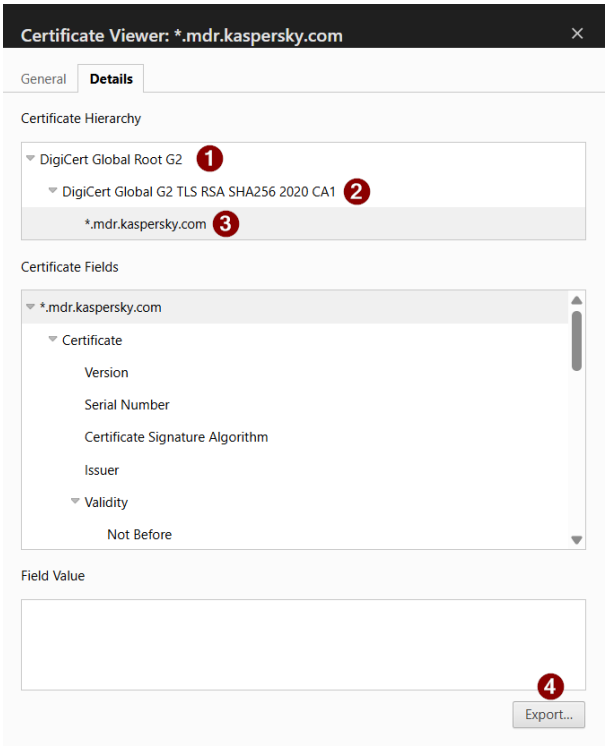
```
git clone https://github.com/KUMA-Community/MDR-integrations.git
```

5. ????????? ??????? ???????????

Загрузите актуальную цепочку сертификатов для **mdr.kaspersky.com** в формате PEM. Для этого:

1. Откройте https://mdr.kaspersky.com в браузере.

2. Просмотрите информацию о сертификате и экспортируйте все сертификаты цепочки (как правило, 2–3 файла).



4. Объедините их в один файл командой:

```
cat "DigiCert Global G2 TLS RSA SHA256 2020 CA1.crt" "DigiCert Global Root G2.crt"
"_.mdr.kaspersky.crt" > mdr.pem
```

Полученный файл `mdr.pem` потребуется на этапе настройки.

?????????

??? 1. ?????????? ?????? ?? ???????

Скопируйте архив **MDR-integrations-main.zip** на сервер (при распределённой инсталляции — на сервер Core) и распакуйте его в папку `/opt`:

```
unzip MDR-integrations-main.zip -d /opt/
```

Уберите лишний уровень вложенности:

```
mv /opt/MDR-integrations-main/mdr_integration /opt/
rm -rf /opt/MDR-integrations-main
```

??? 2. ???????? ????? ???????????

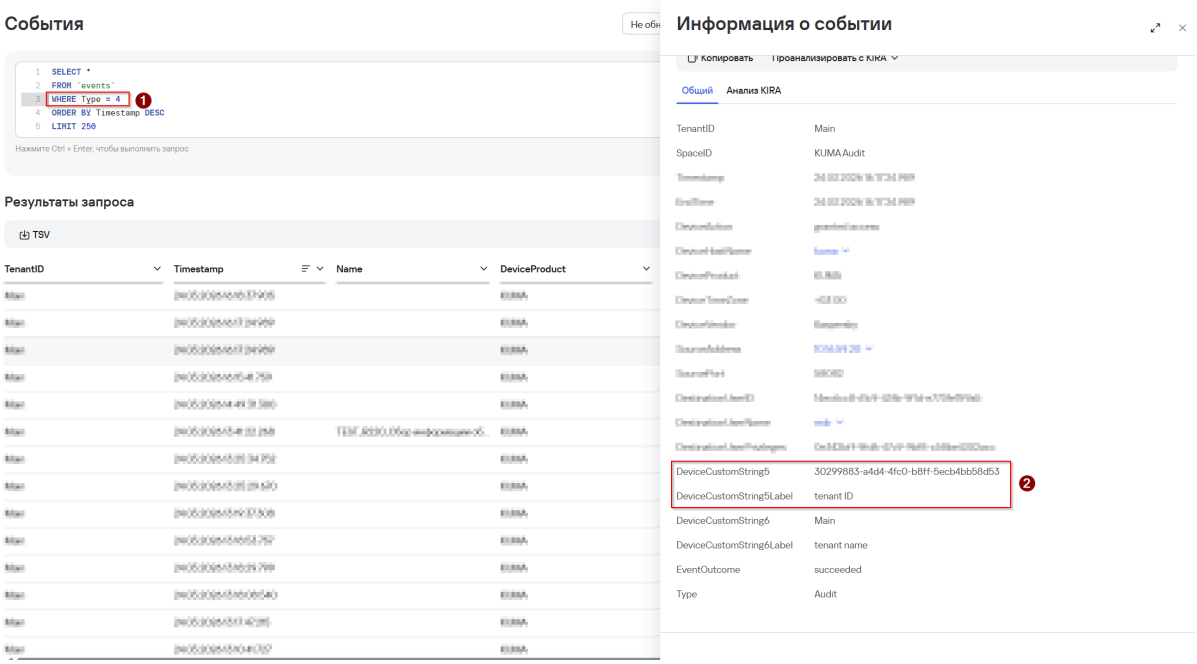
Перейдите в папку с конфигурацией и создайте файл `config.yml` из шаблона:

```
cd /opt/mdr_integration/conf
cp sample_config.yml config.yml
```

Отредактируйте файл `config.yml`:

- В секции **General settings** укажите `client_id` (значение **ClientID** из консоли MDR).
- В секции **Modules settings → kuma** укажите:
  - `api_url` — FQDN или IP-адрес с портом API KUMA (по умолчанию порт `7223`).
  - `api_token` — токен, сгенерированный при создании учётной записи пользователя KUMA.
  - `tenant_id` — идентификатор тенанта KUMA, в котором будут создаваться инциденты.

> **tenant_id можно получить из события аудита KUMA.



??? 3. ????????? ???????

Скрипт использует два файла токенов:

- `.refresh_token` — заполняется вручную один раз. Это **JWT Token**, полученный в консоли MDR на этапе предварительных требований.
- `.access_token` — заполнять вручную **не нужно**. При запуске скрипт автоматически обменивает `refresh_token` на пару `access_token` + обновлённый `refresh_token` через

MDR API и записывает их в файлы. В дальнейшем фоновый процесс `TokenUpdater` проверяет и обновляет токены каждые 10 минут.

Создайте оба файла (`.access_token` должен быть пустым):

```
touch /opt/mdr_integration/conf/.refresh_token
touch /opt/mdr_integration/conf/.access_token
```

Запишите JWT Token из консоли MDR в файл `.refresh_token`:

```
echo -n 'ВСТАВЬТЕ_ТОКЕН_ЗДЕСЬ' > /opt/mdr_integration/conf/.refresh_token
```

> **Важно:** В конце файла не должно быть символа новой строки (`\n`) — иначе аутентификация завершится ошибкой. Флаг `-n` в команде `echo` предотвращает его добавление. Если вы вставили токен вручную, проверьте и при необходимости удалите символ новой строки:
>
> ```
> # Проверяем наличие символа новой строки (результат "1" означает, что он
> есть)
> wc -l /opt/mdr_integration/conf/.refresh_token
>
> # Если вывод "1 .refresh_token" — удаляем символ
> perl -p -i -e 'chomp if eof' /opt/mdr_integration/conf/.refresh_token
>
> # Проверяем повторно (должен быть вывод "0 .refresh_token")
> wc -l /opt/mdr_integration/conf/.refresh_token
> ```

??? 4. ???????? ????????? ????? ????? ??????????

В файле `.last_check` укажите момент времени, начиная с которого нужно собирать инциденты. Значение задаётся в миллисекундах (Unix timestamp × 1000, 13 цифр).

Для получения нужного значения используйте команду:

```
# Пример: получить timestamp для 1 января 2024 00:00:00
date -d "2024-01-01 00:00:00" +%s000
```

Запишите полученное значение в файл:

```
echo -n '1704060000000' > /opt/mdr_integration/conf/.last_check
```

> Для тестирования укажите время, незадолго до появления последнего инцидента в MDR.

??? 5. ?????????? ???????????

Замените файл сертификата по умолчанию на актуальный, подготовленный в разделе **Предварительные требования**:

```
cp /path/to/mdr.pem /opt/mdr_integration/conf/mdr.pem
```

??? 6. ???????? ????? data

Создайте папку **data**:

```
mkdir /opt/mdr_integration/data
```

В папку **data** в отдельный JSON-файл записывается каждое обновление (новый инцидент, комментарий, вложение) на MDR-портале.

??? 7. ???????? ???????????????

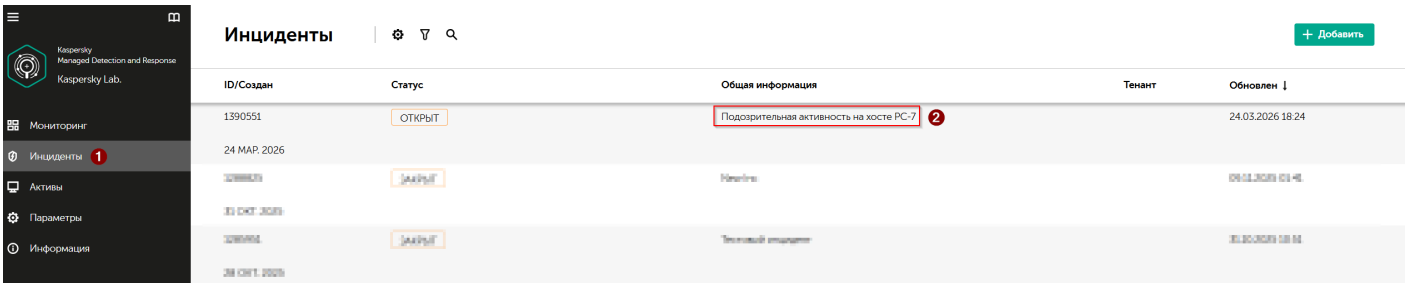
Перейдите в папку со скриптом и запустите его:

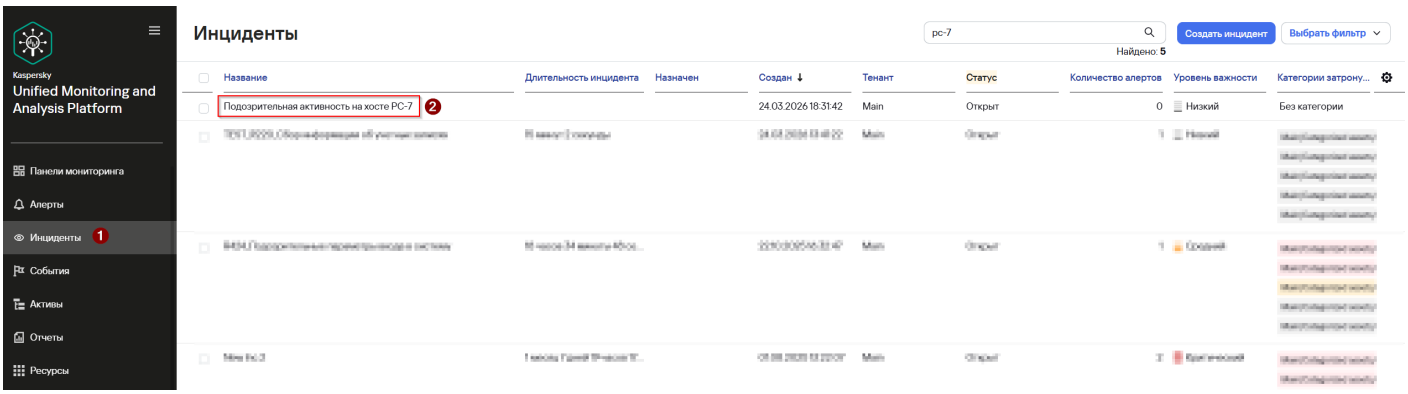
```
cd /opt/mdr_integration
python3 ./main.py
```

Если при запуске скрипта в консоли отсутствую ошибки (кроме предупреждений о невалидном сертификате), значит интеграция работает корректно. Лог работы скрипта записывается в:

```
/opt/mdr_integration/log/app.log
```

Убедитесь, что инциденты из консоли MDR, созданные начиная с момента, указанного в `.last_check`, появились в KUMA.





После проверки остановите скрипт (`Ctrl+C`).

??? 7. ?????? ? ??????? ??????

Запустите скрипт в фоне:

```
nohup python3 /opt/mdr_integration/main.py &
```

??? 8. ?????????? ????? ???????????

Настройте автоматический запуск скрипта через cron:

```
sudo crontab -e
```

Добавьте строку:

```
@reboot sleep 300 && python3 /opt/mdr_integration/main.py &
```

> Задержка sleep 300 (5 минут) необходима, чтобы сервис kuma-core успел запуститься до начала работы скрипта.

Revision #7
Created 2026-03-24 13:00:37 UTC by Dmitry Borisov
Updated 2026-03-24 15:56:20 UTC by Dmitry Borisov