

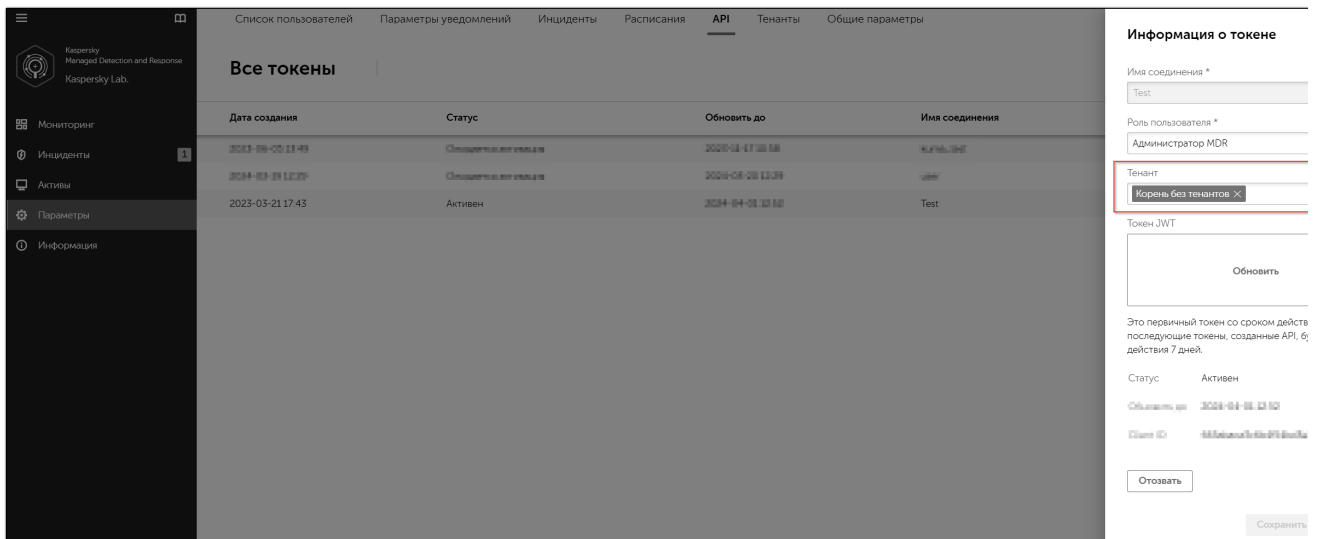
Интеграция с Kaspersky MDR

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

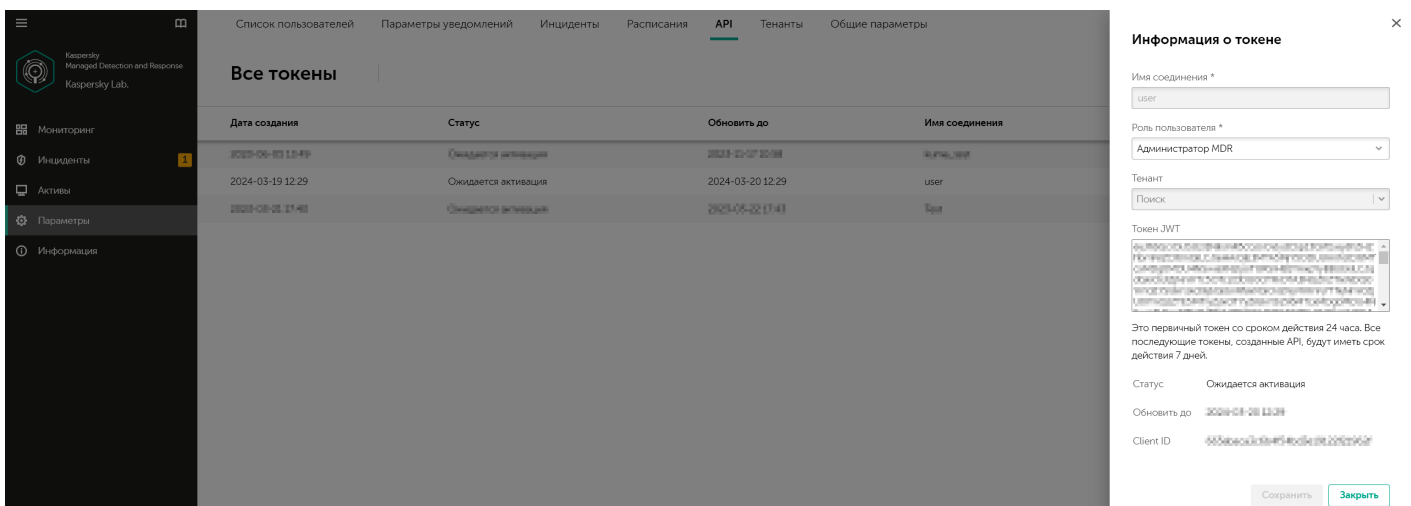
Предварительные требования

- На пограничном МСЭ создайте правило доступа к **mdr.kaspersky.com** по порту **TCP/443** (<https://mdr.kaspersky.com>)
- В KUMA добавьте пользователя с ролью Администратор и доступом к API (права **POST /events**)
- В консоли MDR сгенерируйте токен для доступа к API:
 - Перейдите в **Settings -> API**
 - Нажмите **Add** и укажите **Connection Name** (данное имя будет использоваться, как имя пользователя при создании инцидентов/комментариев/вложений и т.д., так как токен доступа не привязан к конкретному пользователю)
 - Укажите **Role**, чтобы определить права доступа для токена
 - Укажите **Tenant** при необходимости

Если для создаваемых в консоли MDR инцидентов в поле "Тенант" отсутствует значение, добавьте значение "**Корень без тенанта**", чтобы скрипт при подключении обнаружил данные инциденты



- Нажмите **Generate**
- После завершения процесса генерации будут получены:
 - **JWT Token** - он же **refresh_token**, который требуется активировать, чтобы получить новую пару **refresh_token** и **access_token**
 - **ClientID** - ID-клиента для подключения к API (требуется указывать при каждом запросе к API MDR)



- Загрузите архив **kuma_mdr_integration.tar.gz** со скриптом отсюда - <https://box.kaspersky.com/f/11a58e42f63e4cef8741/>
- Загрузите актуальную цепочку сертификатов для консоли mdr.kaspersky.com в формате pem. После загрузки файлов выполните объединение сертификатов в один файл **mdr.pem** (см. скриншот ниже).



General **Details**

Certificate Hierarchy

▼ DigiCert Global Root G2 3

▼ DigiCert Global G2 TLS RSA SHA256 2020 CA1 2

*.mdr.kaspersky.com 1

Certificate Fields

▼ *.mdr.kaspersky.com

▼ Certificate

Version

Serial Number

Certificate Signature Algorithm

Issuer

▼ Validity

Not Before

Field Value



Export...

```
1 -----BEGIN CERTIFICATE-----
2 MIIHujCCBqKgAwIBAgIQCfd78crUhJ1lnoXdeUPT9TANBgkqhkiG9w0BAQsFADBP
3 MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMSkwJWYyDVQQDEyBE
4 aWdpQ2VydCBUTFMgU1NBIFNIQTl1NiAyMDIwIENBMTAeFw0yMTA3MDUwMDAwMDBa
5 Fw0yMjA3MjAyMzU5NTlaMFcxZzAJBgNVBAYTA1JVMQ8wDQYDVQQHEwZnb3Njb3cx
6 GTAXBgNVBAoTEEFPIEthc3BlcnNreSBMYWlxdAAaBgNVBAMMEyoubWRyLmthc3Bl
7 cnNreS5jb20wggiMA0GCSqGSIb3DQEBAAQUAA4ICDwAwggIKAoICAQDcunj7B+v7
8 Ywv8eQeSl10Yr4fR8PDr8cmW1hBuB/BjtLITcaO6DnB4qucZueEchF88cZSKexV
9 21liWitOfU/6hzm0+RdxqePFX+/B1j9mlrmNVAtXWwPIUa+bC5fJ2uYmqWDkQK6
10 5R4yEM6sg+671/cePUJqLQBAT/5QvVj2uCLvUushhEkxWpqETXLswVDdYltcqSnP
11 n19lnbQk4TivWPONDI5zPICFHfr4mUFysMCETSzc7i7a2ZcHV0qLQLQa9m9c+OzN
12 Yax8PFTOn2ltvYxhVXYam6amU9hYzitJg/ws9nI+FgkEDvn/wHmNeB9/OnZzWNZx
13 IVPaz1fwPk1cQAti0vd93+Vw8jCBV6wX7L1MPF1EGB+UTjgX4bnzo6kzrvr3IMZe
14 Y98UK48CS8SHT/gt8gIduf+RcxJaPFJ0rNQ7Z0fzWukGs19/BxIf06THM9jfwC/D
15 +q/SpYUtGtdD3DUw6kGVM8tsVP5aIvI5Yt/yIhstHxLIKbfYM45fj5hLHeN8AtVj
16 RGV0iCXqB+kCrpetl3uMS0su/Psg4JNZNdZgLCa7N3XCBto3iLA3gSxep05wRVA1
17 63wmadIPTx7QgvDstIToZVfxchleIbifCJ/0LuXE3edhc7e/ANvnsPyFo6dYEUeV
18 4yULT2zuosuCRgJjCaCN06PsV4Ntt2pcSQIDAQABo4IDiDCCA4QwHwYDVR0jBBgw
19 FoAUT2ui6qiqhIx56rTaD5iyxZV2ufQwHQYDVR0OBBYEFN/FZS5dTdcvvoSwwCS1
20 ZoDZQLXnMDEGA1UdEQQqMCiCEyoubWRyLmthc3BlcnNreS5jb22CEWlkc5rYXNW
21 ZXJza3kuY29tMA4GA1UdDwEB/wQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYI
22 KwYBBQUHAIWgY8GA1UdHwSBhZCBhDBAoD6gPIY6aHR0cDovL2NybdMuZGlnaWNl
23 cnQuY29tL0RpZ21lDZXJ0VExTU1NB0hBMjU2MjAyMENBMS0xLmNybdBAoD6gPIY6
24 aHR0cDovL2NybdQuZGlnaWNlcnQuY29tL0RpZ21lDZXJ0VExTU1NB0hBMjU2MjAy
25 MENBMS0xLmNybdBA+BgNVHSAENzAlMDMGBmeBDAECAjApMCcGCCsGAQUFBwIBFhto
26 dHRwOi8vd3d3LmRpZ21lZXJ0LmNvbS9DUFMwfwYIKwYBBQUHAQEecBxMCQGCCCg
27 AQUFBzABhhhodHRwOi8vb2Nzc5kaWdpY2VydC5jb20wSQYIKwYBBQUHMAKGPWh0
28 dHA6Ly9jYWNlcnRzLmRpZ21lZXJ0LmNvbS9EaWdpQ2VydFRMU1JTQVNIQTl1NjIw
29 MjBDQTEtMS5jcnQwDAYDVR0TAQH/BAIwADCCAX0GCisGAQQBlbkCBAIEggFtBIIB
30 aQFnAHUARqVV63X6kSAwtaKJafTzfRESQXS+/Um4havy/HD+bUcAAAF6dhWgCGAA
31 BAMARjBEAiBvkP0ksqgsQtTkjrtT7zf+VjltRLqvb5/k+prilZ21rAIGKlO/CWIV
32 kSxZ1YS40qJFYt7f/cVzochPtAspDUXxz/AADQBRo7D1/QF5nFZtuDd4jwykeswb
33 J8v3nohCmg3+1IsF5QAAAXp2HAaoAAAEAwBGMEQCIFf5B07TdaJtpj63WtS2nzF4
34 Y6JCKHX55Wzfgb+UAuHHAiBSMJ+Uu5i5RGe28BJ8pbRjMynRfe6317f+Ozhg1LjH
35 eQB3AEHIYrHfIkZKEMahOglCh15OMysbA+vrs8do8JBilgb2AAABenYcBoEAAQD
36 AEgWRgIhAMjnJKTPEq/oQ6RGz7NldgVfHmbUXOoPD3BdtizdspSXAiEA7H4lySGM
37 XuVdw38QJjBkk/rScAmT3nCyURxh3/ao6YwDQYJKoZIhvcNAQELBQADggEBAGyi
38 zDK8GnkfWVriNouzGZK2t9zei2N8acBoUDDIFWsdW3/D+nOA+VC+FYdLmf3fTUR1
39 50d8xTbHw9Vh0bSFfozk/zurCUFs+IFxU3kDgUWZEJgvrRSsLbGh/ORIaj+TZDB
40 YWoQmG219QyowUngSDGZrmmPj0X0F1CIU/idH4SHPKotOTbx3s6eijbzaReHLrie
41 jKE+hWZlQRlnRyugWR4++p4qwxXATx24lW9uf8zZV8RkT2gms9Qj4kPS/sLWtv3
42 oBJN6vOUCT7j3BdLFEoXaMdyQxJZLjmcJrDqTorkjMSWlAxckT3XzkggXTlXG3uC
43 Ovev/C3aBoxtuA6Oevk=
44 -----END CERTIFICATE-----
45 -----BEGIN CERTIFICATE-----
46 MIIEvjCCA6agAwIBAgIQBtjZBNVYQ0b2ii+nVCJ+xDANBgkqhkiG9w0BAQsFADBh
47 MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYyDVQQLExB3
48 d3cuZGlnaWNlcnQuY29tMSAwHgYyDVQQDExEaWdpQ2VydCBHbG9iYWwU9vdCBD
49 QTAEFw0yMTA0MTQwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
50 MRUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
51 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
52 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
53 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
54 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
55 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
56 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
57 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
58 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
59 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
60 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
61 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
62 MUwEwYyDVQQKEwEaWdpQ2VydCBHbG9iBmMxKTANBgNVBAMTIERpZ21lDZXJ0IFR
```

Описание интеграции

Описанная в данной статье интеграция с Kaspersky Managed Detection and Response позволяет автоматически импортировать инциденты из консоли MDR в KUMA.

Настройка

- Скопируйте архив **kuma_mdr_integration.tar.gz** на сервер (в случае распределенной инсталляции на сервер Core) и распакуйте его в папку **/opt** с помощью следующей команды:

```
sudo tar -xf kuma_mdr_integration.tar.gz -C /opt
```

- Перейдите в папку **/opt/mdr/conf** и отредактируйте файл **config.yml**:
 - В секции **General settings** укажите **актуальный путь до папок (по умолчанию, указан /opt/mdr/*)**, а также **client_id**
 - В секции **Modules settings** -> **kuma** укажите:
 - **api_url** (FQDN/IP:порт API-интерфейса KUMA)
 - **username** (ранее созданный пользователь с доступом к API)
 - **password** (пароль ранее созданного пользователя с доступом к API)
 - **tenantId** (тенант, в котором будут создаваться инциденты)

TenantID можно получить из события аудита KUMA

- В секции **Modules settings** -> **logging** также укажите **актуальный путь до папки со скриптом (по умолчанию, указан /opt/mdr/log)**
- Перейдите в папку **/opt/mdr/conf** и добавьте в файл **.refresh_token** ранее сгенерированный токен для доступа к API MDR

После добавления токена в файл **.refresh_token** проверьте, что в конце файла отсутствует символ новой строки **\n**, из-за которого попытка аутентификации будет неуспешной. См. команды ниже:

```
# проверяем наличие символа новой строки
wc -l /opt/mdr/conf/.refresh_token

# если вывод "1 .refresh_token", то удаляем символ
perl -p -i -e 'chomp if eof' /opt/mdr/conf/.refresh_token

# проверяем, что символ новой строки успешно удален
wc -l /opt/mdr/conf/.refresh_token
```

```
# должен быть вывод "0 .refresh_token"
```

- Перейдите в папку **/opt/mdr/conf** и в файле **.last_check** укажите время, начиная с которого необходимо начать собирать инциденты. Для теста можно указать время до появления последнего инцидента. Формат в миллисекундах, то есть должно быть 13 цифр (пример, 1672520400000)
- Замените существующий файл **/opt/mdr/conf/mdr.pem** на актуальный (см. этап "Предварительные требования")
- Запустите скрипт **main.py** с помощью команды:

```
python3 ./main.py
```


Если при запуске скрипта появляются сообщения об отсутствии необходимых пакетов - выполните их установку

- Если после запуска скрипта в консоли отсутствуют ошибки (кроме предупреждений о невалидном сертификате), значит интеграция работает корректно.

Лог работы скрипта пишется в **/opt/mdr/log/app.log**

- Убедитесь, что выполнен импорт инцидентов, созданных в консоли MDR начиная с момента времени, указанного в файле **.last_check**

Incidents

<input type="checkbox"/>	Name	Created ↓	Severity	Tenant
<input type="checkbox"/>	W10-MDR-KES.evilcorp.local - проверка зашифрованного архива	2023-09-13 20:31:46	 Low	Main

- Остановите выполнение скрипта **main.py**
- Запустите скрипт **main.py** в фоновом режиме с помощью команды:

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

- Настройте автоматический запуск скрипта после перезагрузки сервера:

```
sudo crontab -e
@reboot sleep 300 && python3 /opt/mdr/main.py & # sleep в 5 минут добавлен, чтобы сервис kuma-core успел стартовать
```

Revision #20
Created 26 February 2024 14:04:55 by Dmitry Borisov
Updated 3 September 2024 08:48:10 by Koala