?????????? OpenCTI ? KUMA

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

> Ссылка на репозиторий https://github.com/KUMA-Community/KumaOpenctiLookupProxy

В современных SOC-инфраструктурах эффективность анализа событий безопасности напрямую зависит от актуальных данных Threat Intelligence. Интеграция платформы OpenCTI с SIEM-системой Kaspersky Unified Monitoring and Analysis Platform (KUMA) позволяет использовать данные о киберугрозах для обогащения событий безопасности и ускорения расследования инцидентов.

Основная задача интеграции — автоматическая проверка индикаторов компрометации (IoC), обнаруженных в событиях KUMA, по базе Threat Intelligence OpenCTI. При нахождении совпадений события дополняются контекстной информацией: описанием индикатора, источником, уровнем доверия и связанными аналитическими отчётами.

Возможности интеграции:

- автоматическая проверка индикаторов компрометации из событий безопасности;
- обогащение событий данными Threat Intelligence;
- связывание обнаруженных индикаторов с отчётами и аналитикой OpenCTI;
- ускорение анализа и расследования инцидентов в SOC.

Таким образом, OpenCTI выступает централизованным источником данных о киберугрозах для KUMA, обеспечивая дополнительный уровень контекстного анализа и повышая качество обнаружения атак.

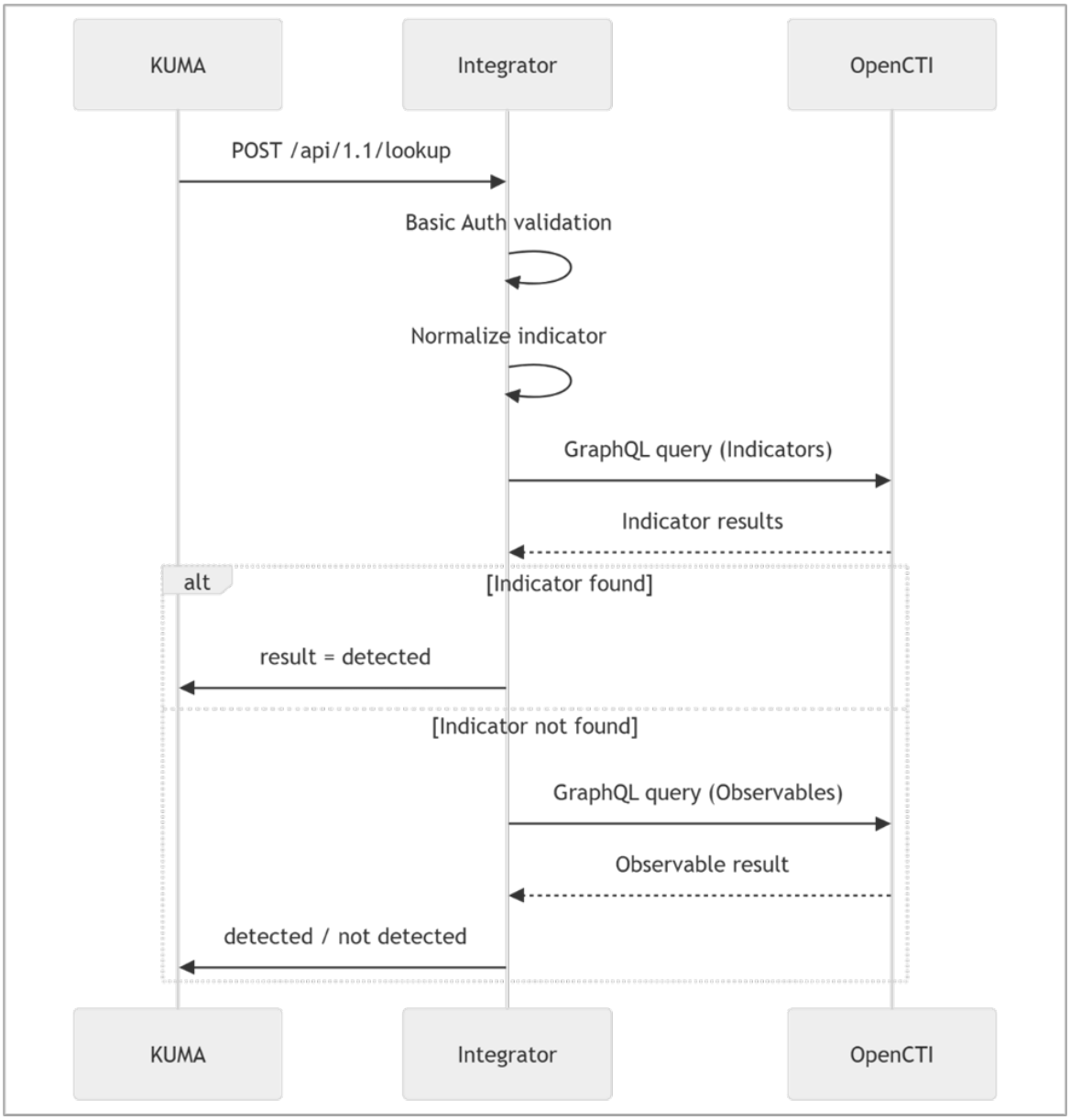
?????????? ??????????

Интеграция реализована через промежуточный сервис **OpenCTI-KUMA Lookup Proxy** — HTTP-сервис, принимающий запросы от KUMA и преобразующий их в запросы к API OpenCTI.

Функции сервиса:

- приём lookup-запросов от KUMA;
- поиск индикаторов в OpenCTI среди объектов Indicators и Observables;

- формирование ответа в формате CyberTrace HTTP, используемом KUMA для обогащения событий.



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

Сервис реализован на **Python** с использованием фреймворка **FastAPI** и развёрнут в Docker-контейнере на сервере OpenCTI в рамках той же Docker-сети.

Для запуска используется **Gunicorn** с Uvicorn workers, что обеспечивает эффективную обработку параллельных запросов. Количество worker-процессов рассчитывается по формуле:

```
workers = 2 × CPU + 1
```

Каждый worker — это отдельный процесс, обслуживающий множество асинхронных HTTP-запросов.

Для доступа к сервису извне используется **Nginx** в роли reverse-proxy: он публикует порт `8000` и обеспечивает доступ по HTTPS. Сост

```
opencti-kuma-proxy:
  image: opencti-kuma-proxy:1.1
  expose:
    - "8000"
  command: >
    gunicorn app.main:app -k uvicorn.workers.UvicornWorker --workers 18 --bind 0.0.0.0:8000 --log-level info
  volumes:
  - /app/opencti/kuma-opencti-proxy:/app
  environment:
    - OPENCTI_URL_INTEGRATION=${OPENCTI_URL_INTEGRATION}
    - OPENCTI_TOKEN=${OPENCTI_TOKEN}
    - LOOKUP_BASIC_USER=${LOOKUP_BASIC_USER}
    - LOOKUP_BASIC_PASSWORD=${LOOKUP_BASIC_PASSWORD}
    - HTTP_TIMEOUT_SECONDS=60
    - HTTP_RETRIES=2
  restart: always
  depends_on:
    opencti:
      condition: service_healthy
  networks:
    - openctinet
```

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

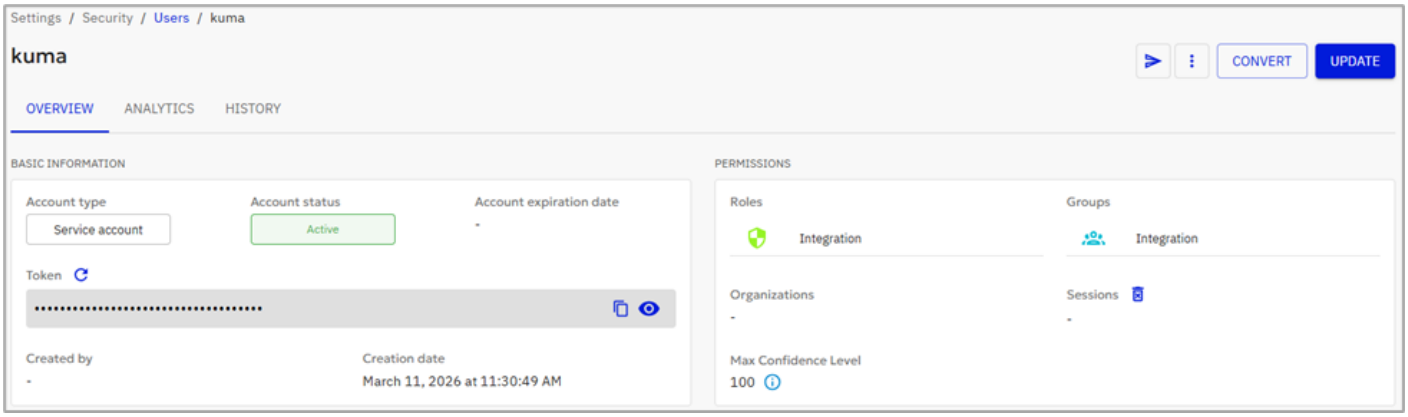
Переменная	Описание
`LOOKUP_BASIC_USER`	Имя пользователя для подключения KUMA к сервису
`LOOKUP_BASIC_PASSWORD`	Пароль для подключения KUMA к сервису
`OPENCTI_TOKEN`	Bearer-токен сервисного пользователя OpenCTI

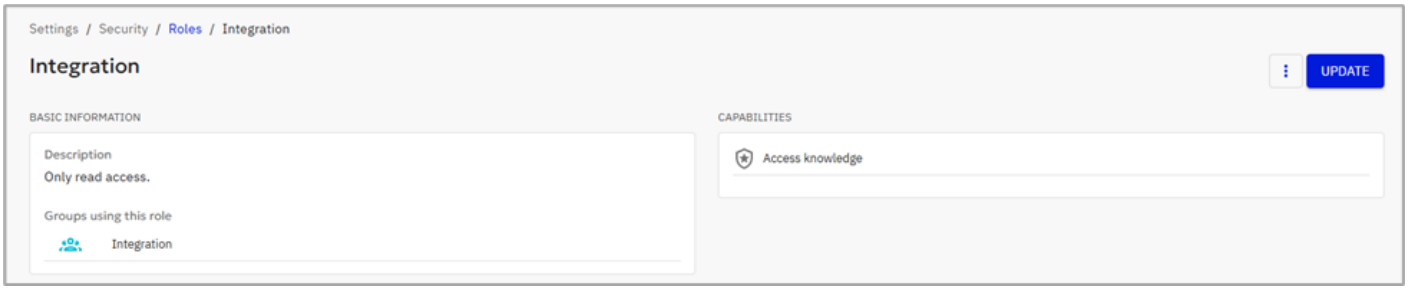
KUMA обращается к сервису с использованием **Basic Authentication**, а сам сервис выполняет запросы к OpenCTI с **Bearer-токеном** сервисного пользователя.

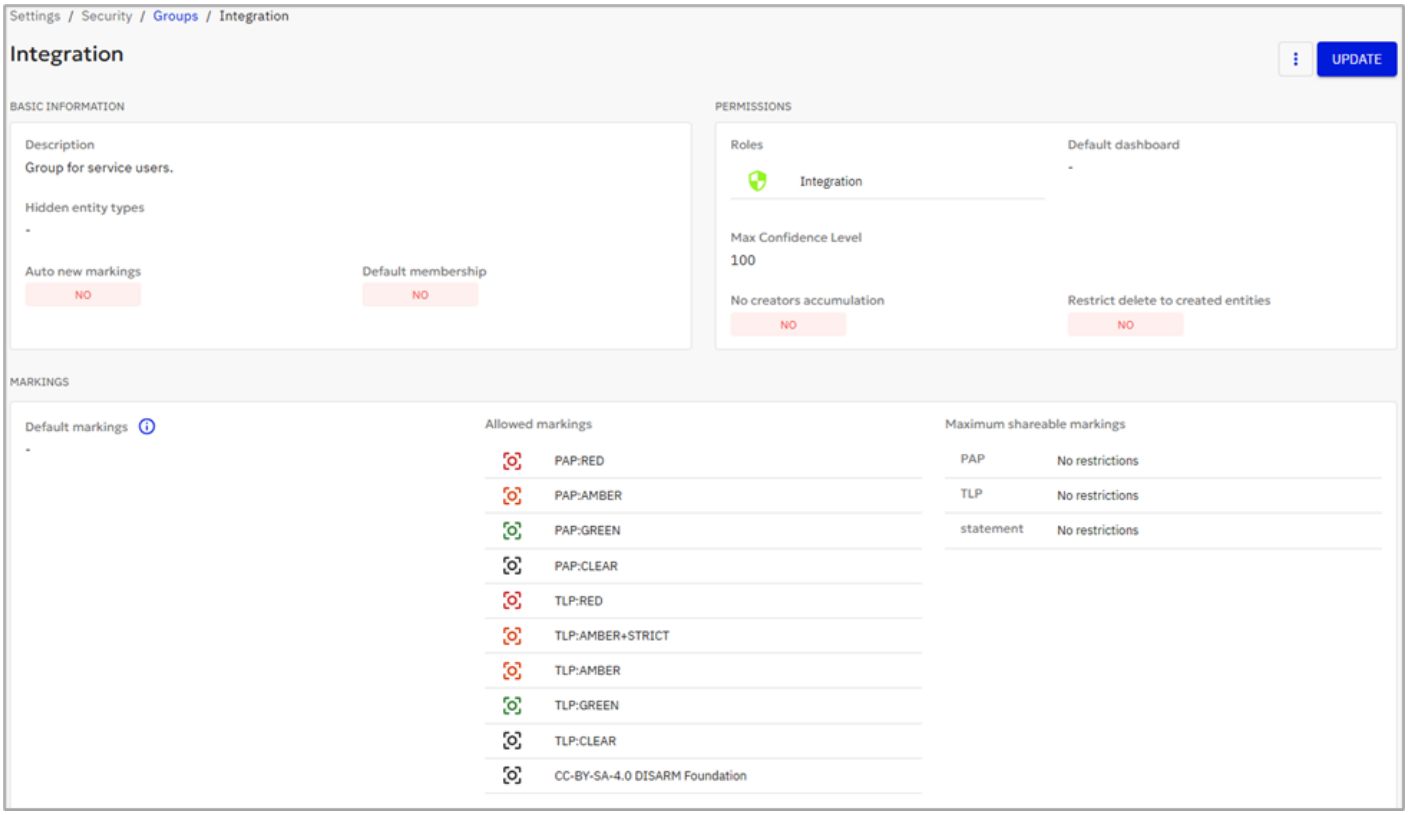
????????? ????????????? ? OpenCTI

Для работы интеграции необходимо создать отдельного сервисного пользователя в OpenCTI.

1. Перейдите в **Settings → Security → Users** и нажмите **Create User**.
2. В окне **New User** укажите **Name** — имя пользователя.
3. Добавьте пользователя в заранее созданную группу **Integration**.
4. Отметьте пользователя как **Service Account**.
5. Откройте профиль пользователя и скопируйте **API Token** — он будет использоваться сервисом для обращения к GraphQL API OpenCTI.







> **Примечание:** группа Integration должна иметь роль с правами **только на чтение** в соответствии с принципом минимально необходимых привилегий.

API ??????? OpenCTI-KUMA Lookup Proxy

Сервис предоставляет два HTTP endpoint:

Метод	Endpoint	Назначение
`POST`	`/api/1.1/lookup`	Проверка индикаторов (используется KUMA)
`GET`	`/health`	Мониторинг состояния сервиса

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

KUMA отправляет список индикаторов в формате JSON. Каждый элемент массива содержит значение индикатора для проверки в OpenCTI.

```
[
  {
    "object": "malicious-"ip
  },
  {
    "object": "malicious-domain.com"
  },
  {
    "object": "https://www.malicious-domain.com""
  },
  {
    "object": "malicious-hash"
  }
]
```

Сервис обрабатывает запрос в три этапа:

1. **Аутентификация** — проверка заголовка `Authorization: Basic base64(user:password)`. Значения сравниваются с `LOOKUP_BASIC_USER` и `LOOKUP_BASIC_PASSWORD`. При несоответствии запрос отклоняется.
2. **Нормализация индикаторов** — например, из URL-адреса дополнительно извлекаются доменное имя и IP-адрес для расширения области поиска.
3. **Поиск в OpenCTI через GraphQL API** — сначала среди объектов **Indicators**, затем, при отсутствии совпадений, среди **Observables**. Для авторизации используется API-токен сервисного пользователя.

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

При обнаружении индикатора в OpenCTI событие KUMA обогащается следующими данными:

Поле	Описание
`description` / `x_opencti_description`	Описание индикатора
`pattern`	STIX-паттерн индикатора
`created_at`	Дата создания
`updated_at`	Дата последнего обновления
`valid_from`	Дата начала актуальности индикатора
`valid_until`	Дата окончания актуальности индикатора
`x_opencti_score`	Степень доверия от 0 до 100
`objectLabel`	Теги индикатора / наблюдаемого объекта
`createdBy`	Источник информации
`reports`	Связанные аналитические отчёты (при наличии)

```
_GQL_indicator = """
query Search($filterGroup: FilterGroup!, $first: Int) {
  indicators(filters: $filterGroup, first: $first) {
    edges {
      node {
        id
        name
        description
        pattern
        created_at
        updated_at
        valid_from
        valid_until
        x_opencti_score
        objectLabel {
          value
        }
        createdBy {
          ... on Identity {
            name
          }
        }
        reports(first: 10) {
          edges {
            node {
              id
            }
          }
        }
      }
    }
  }
}
"""
```

```
_GQL_observable = """
query Search($search: String!, $first: Int) {
  stixCyberObservables(search: $search, first: $first) {
    edges {
      node {
        id
        observable_value
        ... on StixCyberObservable {
          observable_value
          x_opencti_description
          x_opencti_score
        }
        created_at
        updated_at
        objectLabel {
          value
        }
        createdBy {
          ... on Identity {
            name
          }
        }
        reports(first: 10) {
          edges {
            node {
              id
            }
          }
        }
      }
    }
  }
}
"""
```

?????????? ?? ??????? KUMA

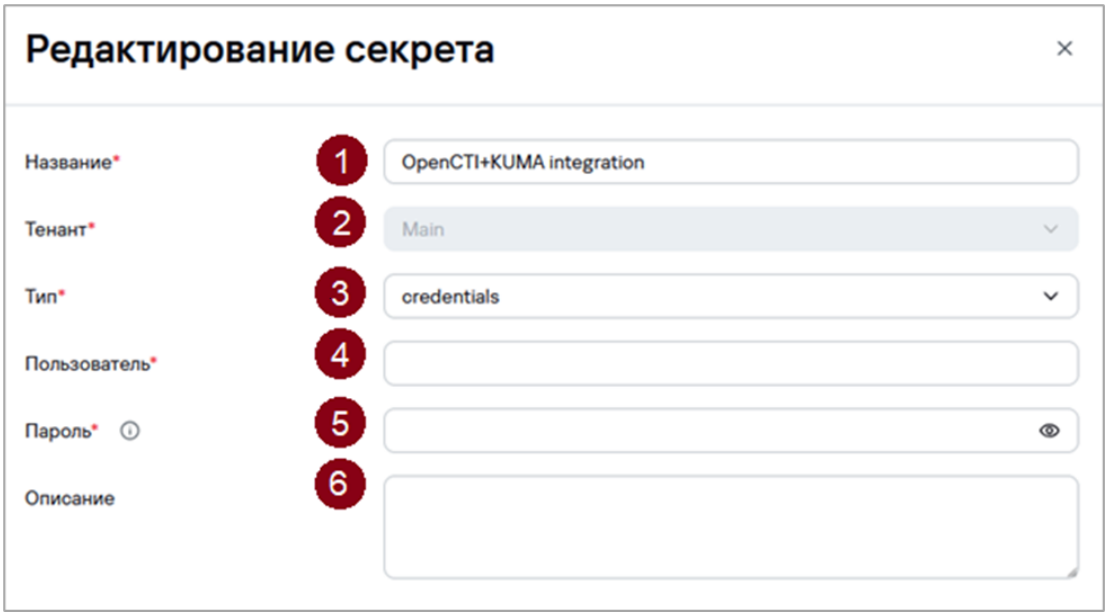
Для интеграции в KUMA используется механизм **cybertrace-http**, обеспечивающий потоковое обогащение событий.

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

Перейдите в **Ресурсы → Секреты → Добавить** и заполните параметры:

Параметр	Значение
Название	`<название секрета>`
Тенант	`<название тенанта, например, Main>`
Тип	`credentials`

Параметр	Значение
Пользователь	`<LOOKUP_BASIC_USER>`
Пароль	`<LOOKUP_BASIC_PASSWORD>`
Описание	*(опционально)*



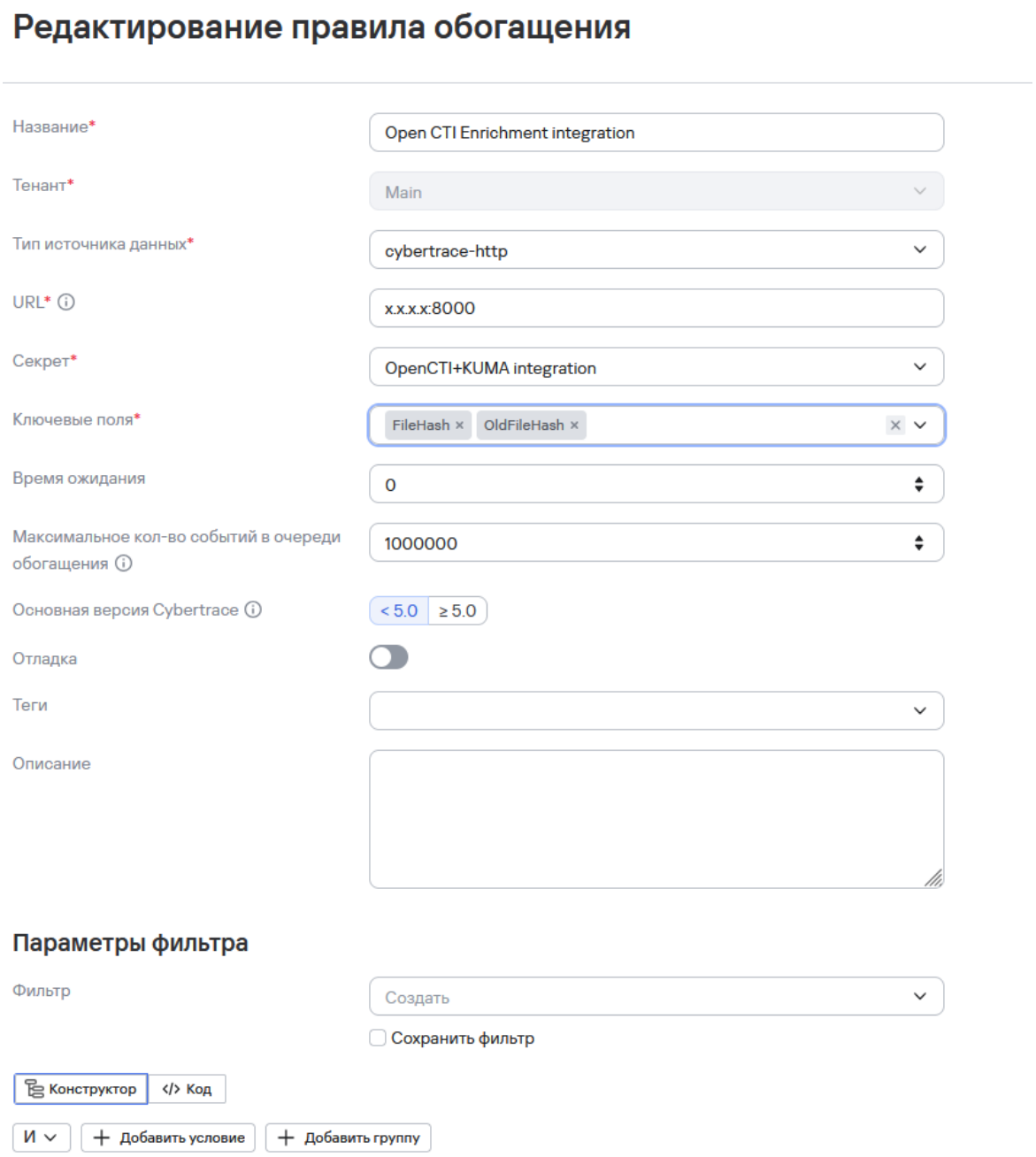
2. ???????? ??????? ???????????

Перейдите в **Ресурсы → Правила обогащения → Добавить** и заполните параметры:

Параметр	Значение
Название	`<название правила обогащения>`
Тенант	`<название тенанта, например, Main>`
Исходный тип	`cybertrace-http`
URL	`<IP-адрес/FQDN сервера OpenCTI:порт>`
Секрет	`<секрет, созданный на предыдущем шаге>`
Ключевые поля	`<поля, значения которых передаются в OpenCTI на анализ>`
Время ожидания	`0`
Макс. кол-во событий в очереди	`1 000 000`
Описание	*(опционально)*
Параметры фильтра	`<условия срабатывания правила, например: внутренние IP → внешние IP>`

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

После создания добавьте правило обогащения на нужный коррелятор.



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

Количество запросов к сервису OpenCTI-KUMA Lookup Proxy не ограничено жёстко в коде, однако зависит от:

- производительности сервера;
- параметров Gunicorn;
- производительности базы данных OpenCTI;
- сложности GraphQL-запросов.

OpenCTI хранит данные в **Elasticsearch** в виде графовой модели STIX:

- **SDO** (STIX Domain Objects);
- **SRO** (STIX Relationship Objects).

Сложные графовые запросы могут существенно снижать производительность, поэтому **не рекомендуется** использовать интеграцию для обогащения всего потока событий.

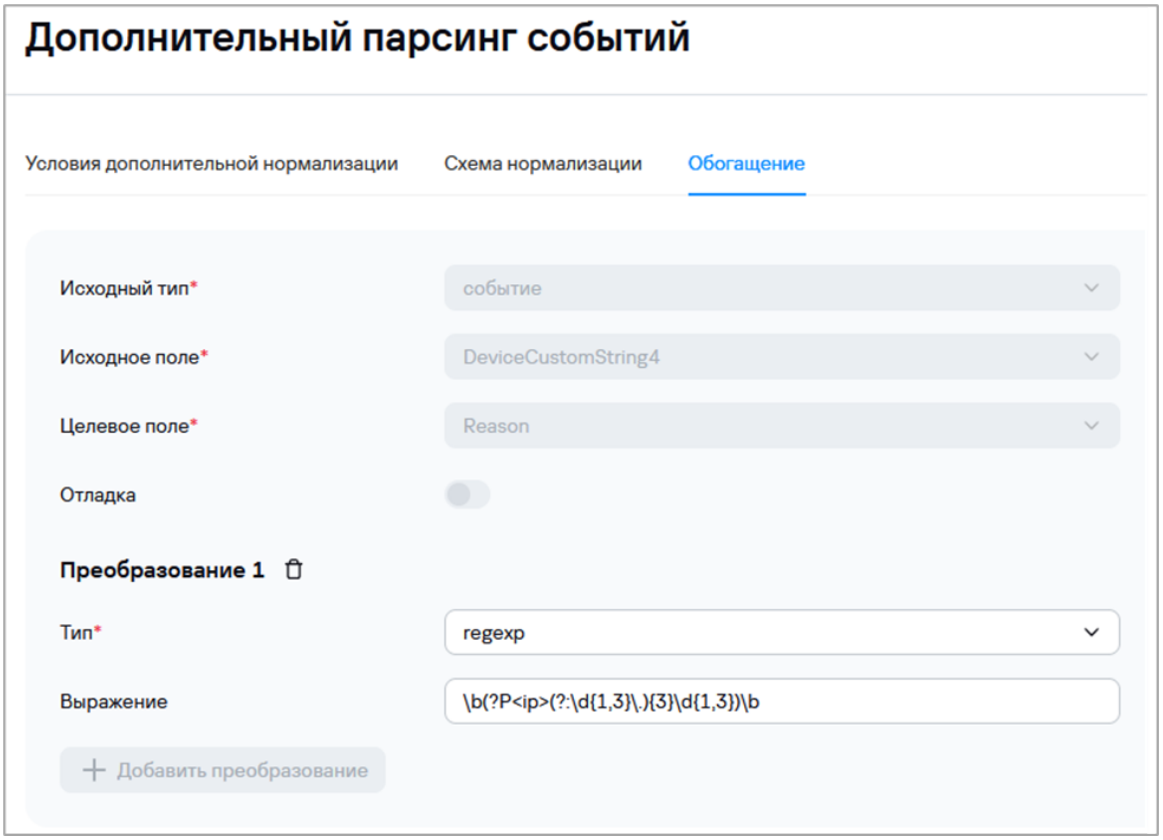
Рекомендуется применять интеграцию для:

- корреляционных событий;
- событий, отфильтрованных по узким условиям.

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

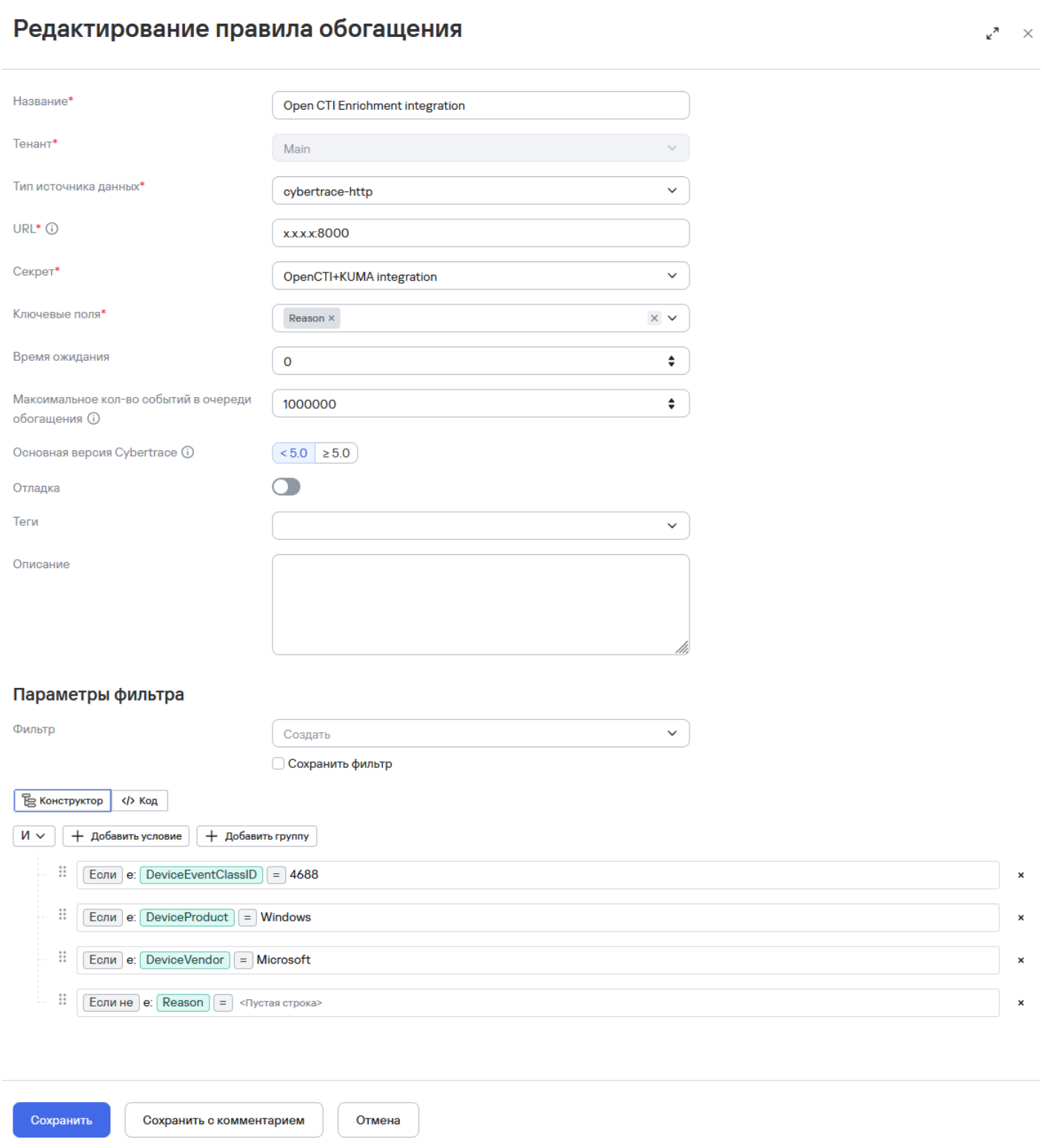
В качестве примера рассмотрим детектирование обращений к вредоносным IP-адресам из командной строки. Для этого:

1. В нормализатор событий Windows Event ID 4688 добавьте извлечение IP-адресов из командной строки (*названия полей могут отличаться с учетом Вашей схемы нормализации).

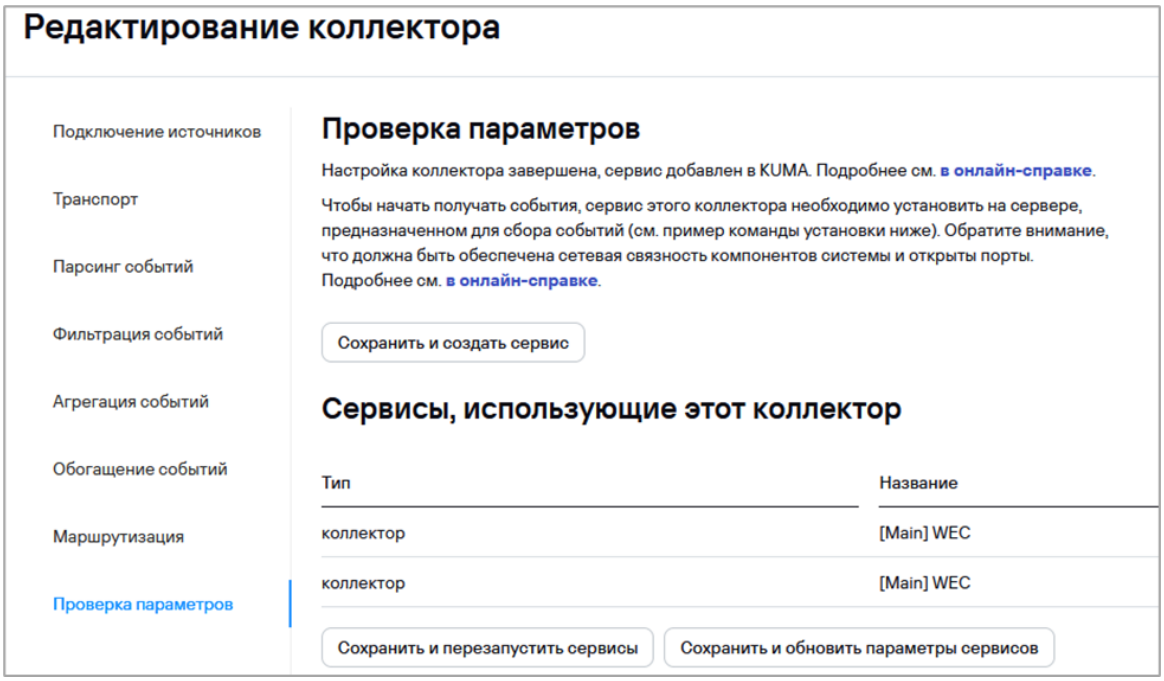


2. Отредактируйте правило обогащения (названия полей могут отличаться с учетом Вашей схемы нормализации). В разделе «Ключевые поля» укажите поле, в которое записывается IP адрес при его обнаружении в командной строке, тем самым «сужая» фильтр поиска событий

для обогащения. Пример настройки правила обогащения ниже:

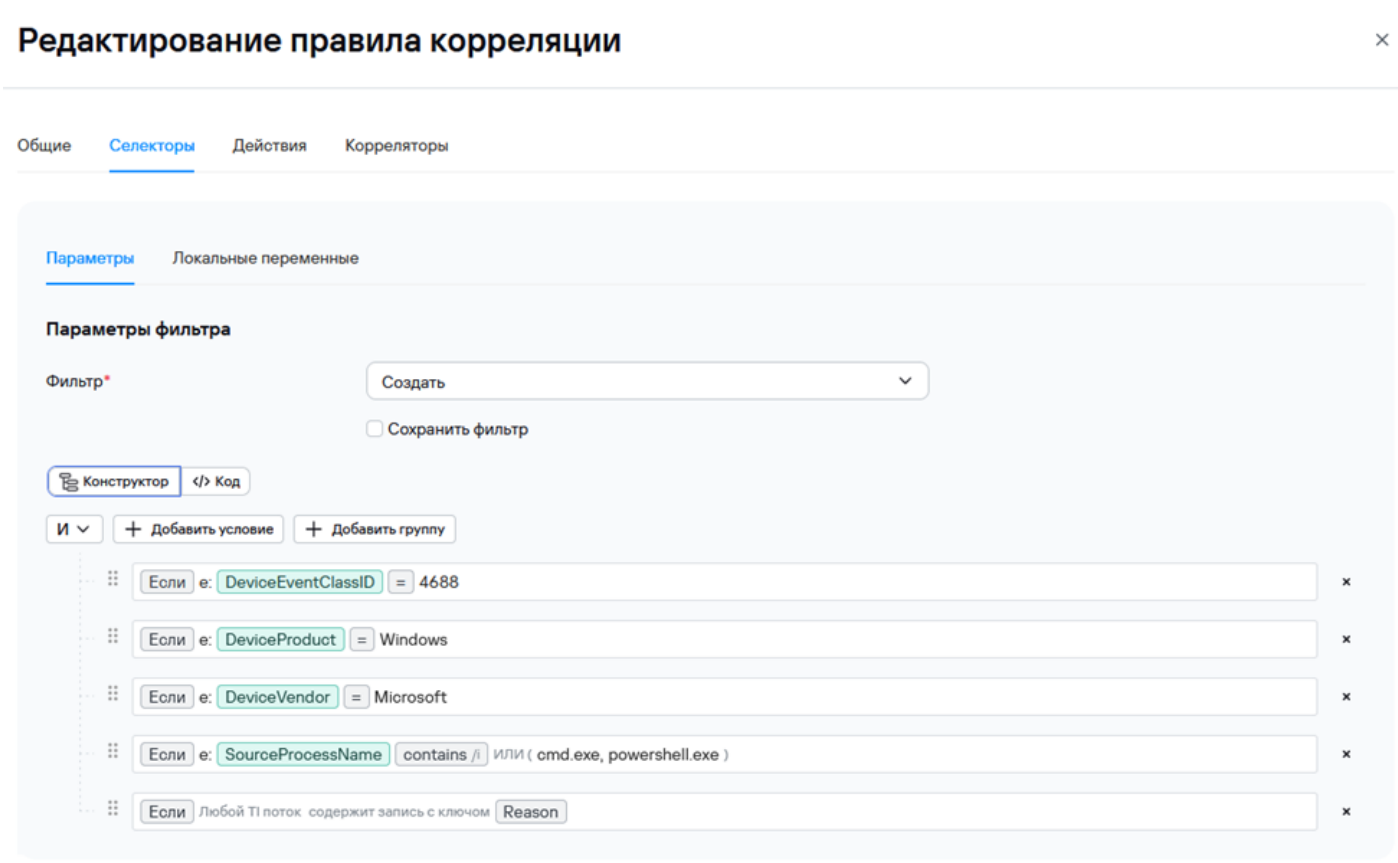


3. Сохраните и обновите параметры коллектора.



4. Создайте правило корреляции с фильтром:

```
DeviceEventClassID = 4688
AND DeviceProduct = 'Windows'
AND DeviceVendor = 'Microsoft'
AND SourceProcessName icontains [
  'cmd.exe',
  'powershell.exe'
]
AND tidetect('', Reason)
```



5. Привяжите правило к коррелятору. После выполнения настроек необходимо сохранить и обновить (можно и перезапустить) параметры коррелятора.

Таким образом, KUMA может обнаруживать обращения к вредоносным IP-адресам, выполняемым из командной строки, и автоматически обогащать такие события данными Threat Intelligence.

Revision #7
Created 2026-04-07 07:41:01 UTC by Boris Rzr
Updated 2026-04-07 10:25:44 UTC by Boris Rzr