

??????????? 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.

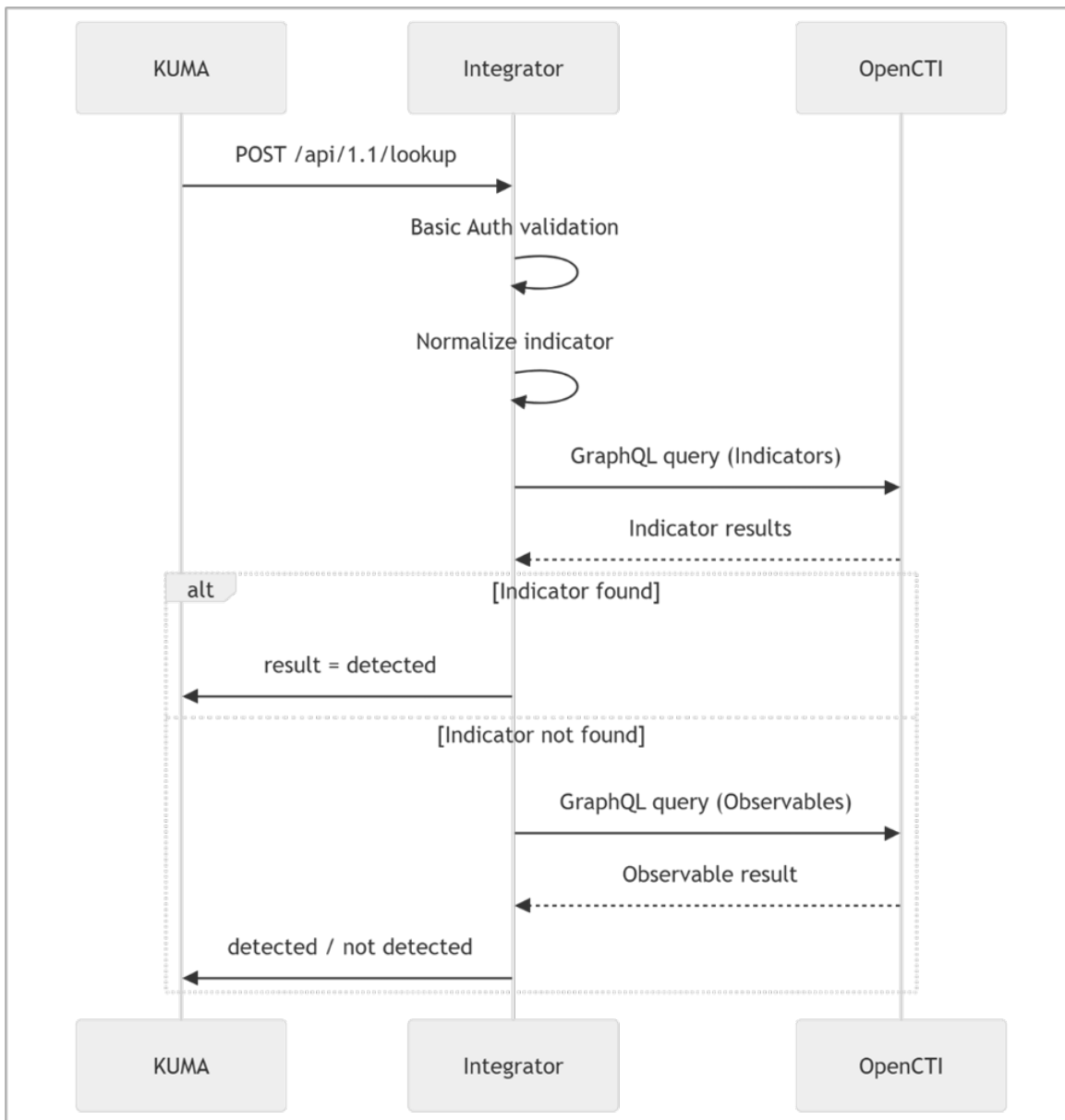
??????????????? ???????????????

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:

$$\text{workers} = 2 \times \text{CPU} + 1$$

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.

Settings / Security / Users / kuma

kuma

▶

⋮

CONVERT

UPDATE

OVERVIEW

ANALYTICS

HISTORY

BASIC INFORMATION

Account type

Service account

Account status

Active

Account expiration date

-

Token

⌂

.....

📄

👁

Created by

-

Creation date

March 11, 2026 at 11:30:49 AM

PERMISSIONS

Roles

Integration

Groups

Integration

Organizations

-

Sessions

🔑

Max Confidence Level

100

🔍

Settings / Security / Roles / Integration

Integration

⋮

UPDATE

BASIC INFORMATION

Description

Only read access.

Groups using this role

Integration

CAPABILITIES

Access knowledge

Settings / Security / Groups / Integration

Integration

⋮

UPDATE

BASIC INFORMATION

Description

Group for service users.

Hidden entity types

-

Auto new markings

NO

Default membership

NO

PERMISSIONS

Roles

Integration

Default dashboard

-

Max Confidence Level

100

No creators accumulation

NO

Restrict delete to created entities

NO

MARKINGS

Default markings

🔍

-

Allowed markings

🔴

PAP:RED

🔴

PAP:AMBER

🟢

PAP:GREEN

🔴

PAP:CLEAR

🔴

TLP:RED

🔴

TLP:AMBER+STRICT

🔴

TLP:AMBER

🟢

TLP:GREEN

🔴

TLP:CLEAR

🔴

CC-BY-SA-4.0 DISARM Foundation

Maximum shareable markings

PAP

No restrictions

TLP

No restrictions

statement

No restrictions

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

Пример запроса

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:

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

Пример ответа

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)

Редактирование секрета

Название*

1

OpenCTI+KUMA integration

Тенант*

2

Main

Тип*

3

credentials

Пользователь*

4

Пароль* ⓘ

5

Описание

6

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.

Редактирование правила обогащения

Название*

Open CTI Enrichment integration

Тенант*

Main

Тип источника данных*

cybertrace-http

URL* ⓘ

x.x.x.x:8000

Секрет*

OpenCTI+KUMA integration

Ключевые поля*

FileHash x OldFileHash x

Время ожидания

0

Максимальное кол-во событий в очереди обогащения ⓘ

1000000

Основная версия Cybertrace ⓘ

< 5.0 ≥ 5.0

Отладка

☐

Теги

Описание

Параметры фильтра

Фильтр

Создать

☐ Сохранить фильтр

Конструктор

</> Код

И ▾

+ Добавить условие

+ Добавить группу

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

Количество запросов к сервису OpenCTI-KUMA Lookup Proxu 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).

Дополнительный парсинг событий

Условия дополнительной нормализации Схема нормализации Обогащение

Исходный тип*

событие

Исходное поле*

DeviceCustomString4

Целевое поле*

Reason

Отладка

☐

Преобразование 1

Тип*

regex

Выражение

`\b(?:P<ip>(?:\d{1,3}\.){3}\d{1,3})\b`

+ Добавить преобразование

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:

Редктирование правила обогащения

Название*

Open CTI Enrichment integration

Тенант*

Main

Тип источника данных*

cybertrace-http

URL ⓘ

x.x.x.x:8000

Секрет*

OpenCTI+KUMA integration

Ключевые поля*

Reason x

Время ожидания

0

Максимальное кол-во событий в очереди обогащения ⓘ

1000000

Основная версия Cybertrace ⓘ

< 5.0 ≥ 5.0

Отладка

☐

Теги

Описание

Параметры фильтра

Фильтр

Создать

☐ Сохранить фильтр

Конструктор

</> Код

И

Добавить условие

Добавить группу

Если e: DeviceEventClassID = 4688

Если e: DeviceProduct = Windows

Если e: DeviceVendor = Microsoft

Если не e: Reason = <Пустая строка>

Сохранить

Сохранить с комментарием

Отмена

3. Сохраните и обновите параметры коллектора.

Редактирование коллектора

Подключение источников

Транспорт

Парсинг событий

Фильтрация событий

Агрегация событий

Обогащение событий

Маршрутизация

Проверка параметров

Проверка параметров

Настройка коллектора завершена, сервис добавлен в KUMA. Подробнее см. [в онлайн-справке](#).

Чтобы начать получать события, сервис этого коллектора необходимо установить на сервере, предназначенном для сбора событий (см. пример команды установки ниже). Обратите внимание, что должна быть обеспечена сетевая связность компонентов системы и открыты порты.

Подробнее см. [в онлайн-справке](#).

Сохранить и создать сервис

Сервисы, использующие этот коллектор

Тип	Название
коллектор	[Main] WEC
коллектор	[Main] WEC

Сохранить и перезапустить сервисы

Сохранить и обновить параметры сервисов

4. Создайте правило корреляции с фильтром:

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

Параметры

Локальные переменные

Параметры фильтра

Фильтр*

Создать

▼

☐ Сохранить фильтр

Конструктор

</> Код

И ▼

+ Добавить условие

+ Добавить группу

Если

е:

DeviceEventClassID

=

4688

×

Если

е:

DeviceProduct

=

Windows

×

Если

е:

DeviceVendor

=

Microsoft

×

Если

е:

SourceProcessName

contains /i

ИЛИ (cmd.exe, powershell.exe)

×

Если

Любой T1 поток

содержит запись с ключом

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