



MaxPatrol VM версия 2.8

Руководство разработчика

© Positive Technologies, 2025.

Настоящий документ является собственностью Positive Technologies и защищен законодательством Российской Федерации и международными соглашениями об авторских правах и интеллектуальной собственности.

Копирование документа либо его фрагментов в любой форме, распространение, в том числе в переводе, а также их передача третьим лицам возможны только с письменного разрешения Positive Technologies.

Документ может быть изменен без предварительного уведомления.

Товарные знаки, использованные в тексте, приведены исключительно в информационных целях, исключительные права на них принадлежат соответствующим правообладателям.

Дата редакции документа: 10.06.2025

Содержание

1.	Об этом документе.....	6
1.1.	Условные обозначения.....	6
1.2.	Другие источники информации о MaxPatrol VM.....	7
2.	Создание файла расширения аудита.....	8
2.1.	Секция \$extractors.....	9
2.1.1.	Экстракторы классов File и FileInfo.....	9
2.1.2.	ODBC-экстракторы.....	12
2.1.3.	Registry Extractor.....	13
2.1.4.	Shell adapters.....	14
2.1.5.	Экстракторы WMI.....	15
2.2.	Секция \$transformers.....	17
2.2.1.	DictValue (JSON) Parser.....	17
2.2.2.	KeyValue Parsers.....	18
2.2.3.	Search Parser.....	19
2.2.4.	XML-парсеры классов XMLFindone и XMLFinditer.....	20
2.2.5.	Resolvers.....	22
2.3.	Секция \$loaders.....	25
2.4.	Примеры заполнения файла расширения аудита.....	29
3.	Виды запросов к API.....	33
3.1.	Управление токенами доступа.....	33
3.2.	Работа с группами активов.....	34
3.2.1.	Получение идентификатора инфраструктуры.....	34
3.2.2.	Получение идентификатора группы активов.....	35
3.2.3.	Создание группы активов.....	39
3.2.4.	Проверка статуса создания группы.....	43
3.2.5.	Удаление группы активов.....	44
3.3.	Получение токена PDQL-запросов.....	45
3.4.	Импорт активов из CSV.....	47
3.4.1.	Получение пользовательских полей.....	47
3.4.2.	Запрос формата CSV-файла.....	49
3.4.3.	Загрузка CSV-файла.....	50
3.4.4.	Выгрузка файла ошибок.....	53
3.4.5.	Запуск импорта активов.....	54
3.4.6.	Проверка статуса импорта.....	54
3.5.	Работа с учетными записями сканирования.....	56
3.5.1.	Добавление учетной записи типа «логин — пароль».....	57
3.5.2.	Получение учетной записи типа «логин — пароль».....	58
3.5.3.	Обновление учетной записи типа «логин — пароль».....	59
3.5.4.	Добавление учетной записи типа «пароль».....	60
3.5.5.	Получение учетной записи типа «пароль».....	61
3.5.6.	Обновление учетной записи типа «пароль».....	62
3.5.7.	Добавление учетной записи типа «сертификат».....	63
3.5.8.	Получение учетной записи типа «сертификат».....	64

3.5.9.	Обновление учетной записи типа «сертификат»	65
3.5.10.	Получение всех учетных записей	66
3.5.11.	Удаление учетной записи	66
3.5.12.	Получение списка меток учетных записей	67
3.6.	Работа со сканами	67
3.6.1.	Добавление нового скана	68
3.6.2.	Получение метаданных скана по идентификатору	69
3.6.3.	Получение содержания обработанного скана	70
3.6.4.	Получение коллекции данных обработанных сканов	70
3.6.5.	Получение пакета информации об обработанных сканах	71
3.6.6.	Получение метаданных сырого скана по идентификатору	72
3.6.7.	Получение содержания сырого скана в формате XML	73
3.6.8.	Получение коллекции данных сырых сканов	74
3.6.9.	Получение пакета информации о сырых сканах	75
3.7.	Работа с профилями сканирования	76
3.7.1.	Получение списка профилей сканирования	77
3.7.2.	Создание профиля сканирования	78
3.7.3.	Получение расширенной информации о профиле сканирования	79
3.7.4.	Обновление профиля сканирования	81
3.7.5.	Удаление профиля сканирования	82
3.7.6.	Генерация профиля PenTest	83
3.7.7.	Проверка наличия уязвимости при сканировании в режиме пентеста	84
3.7.8.	Получение списка схем профилей сканирования	85
3.7.9.	Получение локализации параметров схемы модуля	86
3.7.10.	Удаление из базы ошибок, возникших при миграции профиля	87
3.8.	Управление задачами сканирования	87
3.8.1.	Запрос отчета о всех задачах	88
3.8.2.	Создание задачи	91
3.8.3.	Запрос количества задач	97
3.8.4.	Запрос отчета о всех точках сохранения всех задач	98
3.8.5.	Запрос количества задач по инфраструктуре	99
3.8.6.	Запрос отчета по задаче по идентификатору	99
3.8.7.	Обновление задачи	102
3.8.8.	Удаление задачи	105
3.8.9.	Валидация задачи	106
3.8.10.	Сброс закладок для задачи сканирования	107
3.8.11.	Запуск задачи сканирования	107
3.8.12.	Остановка задачи	108
3.8.13.	Приостановка задачи	108
3.8.14.	Удаление из базы ошибок, возникших при миграции задачи	109
3.9.	Работа с задачами сканирования	109
3.9.1.	Создание задачи	110
3.9.2.	Запрос отчета о всех задачах	113
3.9.3.	Получение времени следующего запуска задачи	118
3.9.4.	Запрос отчета по задаче по идентификатору	119

3.9.5.	Обновление задачи.....	122
3.10.	Мониторинг задач сканирования.....	125
3.10.1.	Получение истории запусков задачи.....	126
3.10.2.	Получение списка подзадач в составе задачи.....	127
3.10.3.	Получение информации о подзадаче.....	129
3.10.4.	Получение ошибок, произошедших при выполнении подзадачи.....	130
3.10.5.	Получение информации об используемых в задаче коллекторах.....	131
3.11.	Журналирование действий пользователя.....	132
3.11.1.	Получение дерева категорий действий.....	132
3.11.2.	Получение возможного значения дополнительного поля.....	133
3.11.3.	Получение описания всех дополнительных полей для всех приложений.....	134
3.11.4.	Получение отчетов о действиях пользователей.....	135
3.11.5.	Получение отчетов о действиях пользователей по фильтру.....	137
3.11.6.	Регистрация событий пользователя.....	139
3.12.	Работа с активами.....	140
3.12.1.	Запуск задачи на удаление активов.....	140
3.12.2.	Получение состояния задачи по удалению активов.....	141
3.12.3.	Запуск задачи на изменение расположения активов в группах.....	142
3.12.4.	Получение состояния задачи по обновлению расположения активов в группах.....	143
3.12.5.	Запуск задачи на обновление паспортов активов.....	145
3.12.6.	Получение состояния задачи по обновлению паспорта активов.....	146
3.12.7.	Отмена задачи обновления паспорта активов.....	147
3.12.8.	Сохранение статуса завершенной задачи.....	148
3.13.	Получение метаданных актива.....	148
3.14.	Экспорт таблицы активов в CSV.....	149
3.15.	Массовые операции над задачами.....	150
3.15.1.	Запуск задач сканирования.....	150
3.15.2.	Остановка задач сканирования.....	151
3.15.3.	Приостановка задач сканирования.....	152
3.15.4.	Удаление задач сканирования.....	152
3.15.5.	Установка расписания задач сканирования.....	153
3.15.6.	Сравнение расписаний запуска задач.....	154
3.15.7.	Сброс закладок для задач сканирования.....	155
3.15.8.	Получение информации о массовой операции.....	155
3.15.9.	Удаление массовой операции.....	156
3.16.	Работа с выбором задач.....	156
3.16.1.	Выбор задач для использования в массовых операциях.....	157
3.16.2.	Изменение параметров выбора задач.....	159
3.16.3.	Получение информации о выбранных задачах по их идентификатору.....	161
3.16.4.	Отмена выбора задач.....	162
4.	Отправка уведомлений через POST-запрос.....	163
4.1.	Поля POST-запроса для уведомления.....	164
4.2.	Тестирование приема POST-запросов.....	165
5.	О технической поддержке.....	168

1. Об этом документе

Руководство разработчика содержит информацию о доступных в MaxPatrol VM функциях сервиса REST API.

Руководство адресовано разработчикам, выполняющим интеграцию MaxPatrol VM со сторонними приложениями.

Комплект документации MaxPatrol VM включает в себя следующие документы:

- Этот документ.
- Руководство по внедрению — содержит информацию для внедрения продукта в инфраструктуре организации: от типовых схем развертывания до инструкций по установке, первоначальной настройке и обновлению продукта.
- Руководство администратора — содержит справочную информацию и инструкции по настройке и администрированию продукта.
- Руководство оператора — содержит сценарии использования продукта для управления информационными активами организации.
- Руководство по настройке источников — содержит рекомендации по интеграции элементов ИТ-инфраструктуры организации с MaxPatrol VM для сбора событий с источников и аудита активов.
- Синтаксис языка запроса PDQL — содержит справочную информацию и примеры синтаксиса, основных функций и операторов языка PDQL, используемых при работе с MaxPatrol VM.
- PDQL-запросы для анализа активов — содержит информацию о стандартных запросах на языке PDQL, предназначенных для проверки конфигураций активов при работе в MaxPatrol VM.

В этом разделе

[Условные обозначения \(см. раздел 1.1\)](#)

[Другие источники информации о MaxPatrol VM \(см. раздел 1.2\)](#)

1.1. Условные обозначения

В документе приняты условные обозначения.

Таблица 1. Условные обозначения

Пример	Описание
Внимание! При выключении модуля снижается уровень защищенности сети	Предупреждения. Содержат информацию о действиях или событиях, которые могут иметь нежелательные последствия

Пример	Описание
Примечание. Вы можете создать дополнительные отчеты	Примечания. Содержат советы, описания важных частных случаев, дополнительную или справочную информацию, которая может быть полезна при работе с продуктом
▶ Чтобы открыть файл:	Начало инструкции выделено специальным значком
Нажмите OK	Названия элементов интерфейса (например, кнопок, полей, пунктов меню) выделены полужирным шрифтом
Выполните команду <code>Stop-Service</code>	Текст командной строки, примеры кода, прочие данные, которые нужно ввести с клавиатуры, выделены специальным шрифтом. Также выделены специальным шрифтом имена файлов и пути к файлам и папкам
<code>Ctrl+Alt+Delete</code>	Комбинация клавиш. Чтобы использовать комбинацию, клавиши нужно нажимать одновременно
<Название программы>	Переменные заключены в угловые скобки

1.2. Другие источники информации о MaxPatrol VM

Вы можете найти дополнительную информацию о MaxPatrol VM [на портале технической поддержки](#).

Портал содержит статьи базы знаний, новости обновлений продуктов Positive Technologies, ответы на часто задаваемые вопросы пользователей. Для доступа к базе знаний и всем новостям нужно создать на портале учетную запись.

Если вы не нашли нужную информацию или решение проблемы самостоятельно, обратитесь [в службу технической поддержки \(см. раздел 5\)](#).

См. также

[О технической поддержке \(см. раздел 5\)](#)

2. Создание файла расширения аудита

Для управления активами в MaxPatrol VM используется доменная модель, подробно описывающая активы для регулярного анализа защищенности инфраструктуры. Модель представлена в виде классов с набором атрибутов. Полный модельный тип актива можно посмотреть в веб-интерфейсе MaxPatrol VM.

Вершиной модели является корневая сущность, поддерживающая наследование. Одна из корневых сущностей модели актива Host с пространством имен Core используется для расширения аудита. Например, сервер с установленным программным обеспечением описывается классом Core.Host (или его наследником), который в качестве одного из атрибутов содержит коллекцию программного обеспечения класса Core.Software (или его наследника).

Для возможности расширения аудита в классы Core.Host и Core.Software добавлен атрибут `AdditionalProperties` с типом данных Dictionary, который позволяет сохранять в формате коллекции пар «ключ — значение» информацию, которой нет в модели. В отличие от остальной модели, содержимое `AdditionalProperties` не привязано к какой-то схеме данных.

Для задач на сбор данных с модулем Audit вы можете расширить область сбора данных с помощью файла в формате YAML. При выполнении задачи будет проведен аудит данных в соответствии с конфигурацией файла расширения.

Примечание. В названии файла не рекомендуется использовать точки и пробелы.

Файл расширения аудита состоит из трех секций:

- `$extractors` — секция, описывающая способ получения данных со сканируемого узла. Обязательная.
- `$transformers` — секция, описывающая способ преобразования данных. Необязательная.
- `$loaders` — секция, описывающая создание или обогащение модели актива, а также способ компоновки данных секций `$extractors` и `$transformers`. Обязательная.

Если секции заполнены некорректно, возможны следующие ошибки валидации:

- `String: False` — некорректный тип данных параметра `root`.
- `Allow: False` — некорректный тип данных для поля в схеме данных.
- `Correct type: False` — указанный тип данных поля не соответствует типу данных, которые возвращает XML-функция.
- `AllParamsExist: False` — не заполнены все обязательные параметры поля.

В этом разделе

[Секция `\$extractors` \(см. раздел 2.1\)](#)

[Секция `\$transformers` \(см. раздел 2.2\)](#)

[Секция \\$loaders \(см. раздел 2.3\)](#)

[Примеры заполнения файла расширения аудита \(см. раздел 2.4\)](#)

2.1. Секция \$extractors

В разделе описаны экстракторы, которые могут быть использованы в секции \$extractors.

В этом разделе

[Экстракторы классов File и FileInfo \(см. раздел 2.1.1\)](#)

[ODBC-экстракторы \(см. раздел 2.1.2\)](#)

[Registry Extractor \(см. раздел 2.1.3\)](#)

[Shell adapters \(см. раздел 2.1.4\)](#)

[Экстракторы WMI \(см. раздел 2.1.5\)](#)

2.1.1. Экстракторы классов File и FileInfo

Экстракторы класса File

Экстракторы класса File позволяют получить содержимое файлов.

Экстрактор Unix file adapter должен иметь следующую структуру:

```
$extractors:
  <Имя экстрактора>
    $template: <Имя плагина>
    $filepath: <Путь к файлу>
    $schema: <Схема получаемых данных>
      Path: String
      Data: String
```

Имя экстрактора, имя плагина, путь к файлу, а также схема данных являются обязательными полями.

Пример

```
$extractors:
  UnixFile:
    $template: unix_file
    $filepath: /etc/testfile
    $schema:
      Path: String
      Data: String
```

Экстрактор Windows file adapter должен иметь следующую структуру:

```
$extractors:
  <Имя экстрактора>
    $template: <Имя плагина>
    $filepath: <Путь к файлу>
    $encoding: <Кодировка для чтения файла>
    $schema: <Схема получаемых данных>
      Path: String
      Data: String
```

Имя экстрактора, имя плагина, путь к файлу, а также схема данных являются обязательными полями.

Пример

```
$extractors:
  WindowsFile:
    $template: windows_file
    $filepath: C:\Users\File.txt
    $encoding: utf-8
    $schema:
      Path: String
      Data: String
```

Экстраторы Unix file adapter и Windows file adapter recursive возвращают значения следующих полей:

- Data — содержимое файла. Обязательное поле;
- Path — путь к файлу. Необязательное поле.

Экстракторы класса FileInfo

Экстракторы класса FileInfo позволяют получить данные о файле.

Экстрактор Windows file info adapter должен иметь следующую структуру:

```
$extractors:
  <Имя экстрактора>
    $template: <Имя плагина>
    $filepath: <Путь к файлу>
```

Имя экстрактора, имя плагина и путь к файлу являются обязательными полями.

Пример

```
$extractors:
  WindowsFileInfo:
    $template: windows_file_info
    $filepath: C:\Users\File.txt
```

Экстрактор Windows file info recursive adapter должен иметь следующую структуру:

```
$extractors:
  <Имя экстрактора>
    $template: <Имя плагина>
    $filemasks: <Список масок регулярных выражений в именах файлов для поиска>
    $ignorlist: <Список масок регулярных выражений в именах каталогов,
исключаемых из результатов поиска>
    $dirs: <Список корневых каталогов для рекурсивного обхода>
    $usesymlinks: <Искать ли по символьным ссылкам>
```

Имя экстрактора, имя плагина, список масок регулярных выражений в именах файлов и список корневых каталогов для рекурсивного обхода являются обязательными полями.

Пример

```
$extractors:
  WindowsFileInfoRecursive
    $template: windows_file_info_recursive
    $filemasks:
      - File\.txt
      - File2\.txt
    $ignorlist:
      - Backup
      - Updates
    $dirs:
      - C:\Users
    $usesymlinks: true
```

Экстракторы Windows file info adapter и Windows file info recursive adapter возвращают следующие данные:

- `CompanyName` — название компании, создавшей файл;
- `FileDescription` — описание файла;
- `FileVersion` — версия файла;
- `FileVersionEx` — расширенная информация о версии файла;
- `InternalName` — внутреннее имя файла при наличии;
- `LegalCopyright` — копирайт, относящийся к файлу;
- `OriginalFilename` — имя файла;
- `ProductName` — название продукта, для которого создан файл;
- `ProductVersion` — версия продукта;
- `ProductVersionEx` — расширенная информация о версии продукта;
- `Path` — путь к файлу.

2.1.2. ODBC-экстракторы

ODBC-экстракторы позволяют делать SQL-запросы в СУБД.

Экстрактор Select Extractor

Экстрактор Select Extractor должен иметь следующую структуру:

```
$extractors:
  <Имя экстрактора>:
    $template: <Имя плагина>
    $table: <Имя таблицы, из которой будут собираться данные>
    $mapping: <Отображение параметров системы в параметрах источника данных>
      <Параметр источника данных>: <Поле в $scheme>
    $schema: <Схема получаемых данных>
      <Поле>: <Тип данных>
```

Имена экстрактора, плагина, таблицы, а также схема данных являются обязательными полями.

Пример

```
$extractors:
  MSSQLSelect:
    $template: mssql_select
    $table: ?db_name.system_table
    $mapping:
      table_catalog: TableCatalog
      table_schema: TableSchema
      table_name: TableName
    $schema:
      TableCatalog: String
      TableSchema: String
      TableName: String
```

Экстрактор Query Extractor

Экстрактор Query Extractor должен иметь следующую структуру:

```
$extractors:
  <Имя экстрактора>
    $template: <Имя плагина>
    $query: | <Запрос к СУБД, из которой будут собираться данные>
      USE [sys]
      SELECT table_catalog, table_schema, table_name
      FROM [information_schema].[?table_name]
    $mapping: <Отображение параметров системы в параметрах источника данных>
      <Параметр источника данных>: <Поле в $scheme>
```

```
$schema: <Схема получаемых данных>
      <Поле>: <Тип данных>
```

Пример

```
$extractors:
  MSSQLQuery:
    $template: mssql_query
    $query:
      USE [sys]
      SELECT table_catalog, table_schema, table_name
      FROM [information_schema].[?table_name]
    $mapping:
      table_catalog: TableCatalog
      table_schema: TableSchema
      table_name: TableName
    $schema:
      TableCatalog: String
      TableSchema: String
      TableName: String
```

Имя экстрактора, имя плагина, запрос к СУБД, а также схема данных являются обязательными полями.

Особенности заполнения полей ODBC-экстракторов:

- Допустимые значения `$template`:
 - `mssql_select`;
 - `mssql_query`;
 - `mysql_select`;
 - `mysql_query`;
 - `oracle_select`;
 - `oracle_query`;
 - `postgresql_select`;
 - `postgresql_query`.
- Поля `$table` и `$query` используют тип данных `String`.
- Значения полей `$mapping` должны соответствовать полям в `$schema`.
- Поля `$schema` используют типы данных `String` и `Int`.

2.1.3. Registry Extractor

Экстрактор Registry Extractor позволяет делать запросы WMI по ключам реестра Windows.

Структура экстрактора:

```
$extractors:
  Registry: <Имя экстрактора>
    $template: <Имя плагина>
    $paths: <Пути к ключам реестра>
    - HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\VirtualBox
    - HKEY_LOCAL_MACHINE\SOFTWARE\Sun\XVM VirtualBox
    - HKLM\SOFTWARE\{{{VirtualBox\d+\.d+ \x64\}}}}
    $mapping: <Отображение параметров системы в параметрах источника данных>
      <Параметр источника данных>: <Поле в $scheme>
    $schema: <Схема получаемых данных>
      <Поле>: <Тип данных>
      SourceRegistryKey: RegistryKey
```

Имя экстрактора, имя плагина, пути до ключей реестра, а также схема данных являются обязательными полями.

Пример

```
$extractors:
  Registry:
    $template: registry_select
    $paths:
    - HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\VirtualBox
    - HKEY_LOCAL_MACHINE\SOFTWARE\Sun\XVM VirtualBox
    - HKLM\SOFTWARE\{{{VirtualBox\d+\.d+ \x64\}}}}
    $mapping:
      InstallDir: InstallDirectory
    $schema:
      InstallDirectory: String
      Version: String
      VersionExt: String
      SourceRegistryKey: RegistryKey
```

Особенности заполнения полей экстракторов Registry Extractor:

- Допустимое значение `$template` — `registry_select`.
- Поля `$paths` используют тип данных `String`, передаются в виде списка и поддерживают регулярные выражения.
- Значения полей `$mapping` должны соответствовать полям в `$schema`.
- В `$schema` поле `SourceRegistryKey` с типом `RegistryKey` является обязательным.

2.1.4. Shell adapters

Экстрактор `shell adapters` используется для реализации пользовательских схем с помощью плагинов.

Структура экстрактора:

```
$extractors:
  <Имя экстрактора>:
    $template: <Имя плагина>
    $command: <Команда>
    $params:
      <Имя параметра>: <Тип данных>
```

Все поля, кроме поля `$params`, являются обязательными. Поле `$params` позволяет динамически задавать параметры для транспортного запроса во время выполнения аудита.

Допустимые имена плагинов: `windows_shell`, `unix_shell`, `device_shell`, `powershell`. Для плагина `powershell` может использоваться необязательное поле `$scheduler` со следующими значениями:

- `True` — запуск команды через планировщик задач;
- `False` — прямой запуск через процесс Windows.

Пример с использованием плагина `windows_shell`:

```
$extractors:
  RawReverseDNSLookup:
    $template: windows_shell
    $command: ping -n 1 -a ?host_field
```

Пример с использованием плагина `unix_shell`:

```
$extractors:
  ExtName:
    $template: unix_shell
    $command: 'tail -n ?n_param /var/log/some.log'
    $params:
      n_param: int
```

2.1.5. Экстракторы WMI

Экстрактор `wmi_select_adapter`

Экстрактор `wmi_select_adapter` предназначен для WMI-запросов формата `SELECT [%field%,] + FROM [%wmi_class%]`.

Структура экстрактора:

```
$extractors:
  <Имя экстрактора>:
    $template: wmi_select
    $query: <Класс WMI, объекты которого необходимо получить>
    $ns: <Пространство WMI для подключения>
    $mapping: <Отображение параметров системы в параметрах источника данных>
      <Свойство класса WMI>: <Поле в $schema>
```

```
$schema: <Схема получаемых данных>
  <Поле>: <Тип данных>
```

Имя экстрактора, класс WMI и схема данных являются обязательными полями.

Пример

Экстрактор для запроса `SELECT Name, Domain, TotalPhysicalMemory, Manufacturer FROM Win32_ComputerSystem` с преобразованием поля `Manufacturer` в поле `Vendor`.

```
$extractors:
  Win32ComputerSystem:
    $template: wmi_select
    $query: Win32_ComputerSystem
    $mapping:
      Manufacturer: Vendor
    $schema:
      Name: String
      Domain: String
      TotalPhysicalMemory: Int
      Vendor: String
```

Экстрактор wmi_query_adapter

Экстрактор `wmi_query_adapter` предназначен для произвольных WMI-запросов.

Структура экстрактора:

```
$extractors:
  <Имя экстрактора>:
    $template: wmi_query
    $query: <Шаблон выполняемого запроса с обязательными операторами SELECT и
FROM>
    $ns: <Пространство WMI для подключения>
    $mapping: <Отображение параметров системы в параметрах источника данных>
      <Свойство из WMI-запроса>: <Поле в $schema>
    $schema: <Схема получаемых данных>
      <Поле>: <Тип данных>
```

Имя экстрактора, шаблон запроса и схема данных являются обязательными полями.

Пример

Экстрактор для запроса `Get-WmiObject -namespace "root\cimv2" -query "SELECT * FROM Win32_Group WHERE LocalAccount = TRUE"`

```
$extractors:
  Win32LocalGroup:
    $template: wmi_query
    $query: SELECT * FROM Win32_Group WHERE LocalAccount = TRUE
```

```

$ns: root\cimv2
$schema:
  Name: String
  Domain: String
  SID: String

```

Особенности заполнения полей экстракторов WMI:

- В названии класса WMI в поле `$query` можно использовать латинские буквы, цифры или символы подчеркивания. Название класса должно начинаться с буквы и не должно заканчиваться символом подчеркивания.
- Если поле `$ns` не заполнено, запрос будет выполнен в пространство `root\cimv2`.

2.2. Секция \$transformers

В разделе описаны трансформеры, которые могут быть использованы в секции `$transformers`.

В этом разделе

[DictValue \(JSON\) Parser](#) (см. раздел 2.2.1)

[KeyValue Parsers](#) (см. раздел 2.2.2)

[Search Parser](#) (см. раздел 2.2.3)

[XML-парсеры классов XMLFindone и XMLFinditer](#) (см. раздел 2.2.4)

[Resolvers](#) (см. раздел 2.2.5)

2.2.1. DictValue (JSON) Parser

Структура трансформера:

```

$transformers:
  <Имя трансформера>
    $template: <Имя плагина>
    $prefix: <Строка ключей, разделяется точкой>
    $ignorecase: <Игнорировать ли регистр в полях>
    $schema: <Схема получаемых данных>
    <Поле>:
      type: <Тип данных>
      attribute: <Значение>

```

Имя трансформера, имя плагина и схема данных являются обязательными полями.

Пример

```

$transformers:
  FindOneParser:

```

```

$template: json_find_one_parser
$prefix: test_prefix
$ignorecase: true
$schema:
  Field_one:
    type: String
    attribute: field_one
  Field_two:
    type: String
    attribute: field_two

```

Пример

```

$transformers:
  FinditerParser:
    $template: json_finditer_parser
    $prefix: test_prefix
    $ignorecase: true
    $schema:
      Field_one:
        type: String
        attribute: field_one
      Field_two:
        type: String
        attribute: field_two

```

2.2.2. KeyValue Parsers

Структура трансформера:

```

$transformers:
  <Имя трансформера>:
    $template: <Имя плагина>
    $separator: <Разделительный символ>
    $strip_chars: <Символ для удаления>
    $ignorecase: <Игнорировать ли регистр в полях>
    $schema: <Схема получаемых данных>
    <Поле>:
      type: <Тип данных>
      attribute: <Значение>

```

Имя трансформера, имя плагина и схема данных являются обязательными полями.

Пример

```
$transformers:
  KeyValueParser:
    $template: key_value_parser
    $separator: test_separator
    $strip_chars: test_strip_chars
    $skip_empty: true
    $ignorecase: true
    $schema:
      Field_one:
        type: String
        attribute: STRING_FIELD
      Field_two:
        type: Int
        attribute: INT_FIELD
      Field_three:
        type: Array(Int)
        attribute: ARRAY_INT_FIELD
```

2.2.3. Search Parser

Для передачи данных в секцию `$loaders` из трансформера Search Parser используются функции.

Трансформер Search Parser ищет текст по заданному регулярному выражению, в котором можно определить именованные группы. В результате поиска в группы попадает текст, соответствующий описанному шаблону. Чтобы получить текст из именованной группы, необходимо применить функцию, которая извлекает значение только из этой группы.

Структура трансформера:

```
$transformers:
  <Имя трансформера>
    $template: <Имя плагина>
    $pattern: <Регулярное выражение>
    $schema: <Схема получаемых данных>
      <Поле>: <Тип данных>
    <Название функции>:
      $template: function
      $apply: <Имя трансформера>
      $output: <Поле регулярного выражения>
```

Имя трансформера, имя плагина и схема данных являются обязательными полями. Функции используются в секции `$loaders` для получения результата работы трансформера.

Пример

```
$transformers:
  NameVersionParser:
    $template: regexp_search
    $pattern: '^.*(?:P<Name>\S+)\s+(?:P<Version>\d+\.\d+\.\d+).*'$
    $schema:
      Name: String
      Version: String
  get_name:
    $template: function
    $apply: NameVersionParser
    $output: Name
  get_version:
    $template: function
    $apply: NameVersionParser
    $output: Version
```

2.2.4. XML-парсеры классов XMLFindone и XMLFinditer

Структура трансформера:

```
$transformers:
  <Имя пользовательского трансформера>
    $template: <Имя плагина>
    $schema: <Схема получаемых данных>
    <Поле>:
      data_type: <Тип данных>
      xml_function: <Название функции поиска>
      xml_path: <Путь до элемента XML>
    $root: null
    $prefix: '*'
  <Имя исходного трансформера>:
    $template: <Имя плагина>
    $schema: <Схема получаемых данных>
    <Поле>:
      data_type: <Тип данных>
      xml_function: <Название функции поиска>
      xml_path: <Путь до элемента XML>
    $root: null
    $prefix: '*'
```

Пример

```
$transformers:
  RuleTermParser:
    $template: xml_findone_parser
```

```

$schema:
  Name:
    data_type: String
    xml_function: text
    xml_path: name
  Source:
    data_type: String
    xml_function: text
    xml_path: from/source-address/name
  Gateway:
    data_type: String
    xml_function: text
    xml_path: then/backup-remote-gateway
  $root: 'root'
  $prefix: '*'
AddressFamilyFinditer:
  $template: xml_finditer_parser
  $schema:
    AddressFamily:
      data_type: String
      xml_function: text
      xml_path: ns:address-family-name
    InterfaceID:
      data_type: String
      xml_function: text
      xml_path: ../ns:name
    FamilyFlags:
      data_type: Array(String)
      xml_function: tags
      xml_path: ns:address-family-flags
  $root: null
  $prefix: '*'

```

Тип данных, название функции поиска, путь до элемента XML в `$schema` являются обязательными полями.

Пример заполнения секции `$loaders`:

```

$loaders:
  VirtualBox:
    - $template: mapping
      $target: Core.Software
      $kind: detect
      $parent: OperatingSystem.Windows.WindowsHost
      $origin:
        $template: get_extractor_path
        $name: VirtualBoxRegistry
      $field_map:

```

```

Name: VirtualBox
Vendor: Oracle
OsFamily: Windows
Architecture: $row.SourceRegistryKey.Arch
InstallPath: $row.InstallDirectory
RuleTerm:
  $template: apply_transformer
  $name: RuleTermParser
  $source: $row.InstallDirectory
AddressFamily:
  $template: apply_transformer
  $name: AddressFamilyFinditer
  $source: $row.InstallDirectory

```

2.2.5. Resolvers

Трансформер Resolvers позволяет отображать полученные значения параметров системы в нормализованном или измененном виде. Трансформер представляет собой расширенный словарь, ключом которого являются данные из экстракторов или других трансформеров.

Структура трансформера:

```

$transformers:
  <Имя трансформера>
    $template: <Имя плагина>
    $default: <Выходное значение, если не найден ключ>
    $undefined: <Выходное значение, если на вход получено значение null>
    $mapping: <Отображение параметров системы в параметрах источника данных>
      <Ключ>: <Значение>
      <Ключ>: <Значение>

```

Имя трансформера и имя плагина являются обязательными полями.

Особенности заполнения полей трансформера:

- Допустимые значения `$template`:
 - `boolean_to_string_resolver`;
 - `int_to_boolean_resolver`;
 - `int_to_int_array_resolver`;
 - `int_to_string_resolver`;
 - `int_to_string_array_resolver`;
 - `major_version_to_string_resolver`;
 - `string_to_boolean_resolver`;
 - `string_to_int_resolver`;

- `string_to_int_array_resolver`;
 - `string_to_string_resolver`;
 - `string_to_string_array_resolver`;
 - `strip_string_to_string_resolver`.
- Строковые поля `$mapping` необходимо указывать в одинарных кавычках. Для обозначения списка в `$mapping` необходимо использовать квадратные скобки.
- Значения `$default` и `$undefined` должны соответствовать типу выходных данных.

Пример

```
$transformers:
  Resolver:
    $template: boolean_to_string_resolver
    $default: '100'
    $undefined: '200'
    $mapping:
      false: '10'
      true: '1'
```

Пример

```
$transformers:
  Resolver:
    $template: int_to_boolean_resolver
    $mapping:
      12: true
      14: false
```

Пример

```
$transformers:
  Resolver:
    $template: int_to_int_array_resolver
    $mapping:
      12: [10, 2]
      13: [1]
```

Пример

```
$transformers:
  Resolver:
    $template: int_to_string_resolver
    $mapping:
```

```
12: '10'
13: '1'
```

Пример

```
$transformers:
  Resolver:
    $template: int_to_string_array_resolver
    $mapping:
      12: ['10', '2']
      13: ['1']
```

Пример

```
$transformers:
  Resolver:
    $template: major_version_to_string_resolver
    $mapping:
      '2.2': '2'
      '3.10': '3'
```

Пример

```
$transformers:
  Resolver:
    $template: string_to_boolean_resolver
    $mapping:
      '12': true
      'true': false
```

Пример

```
$transformers:
  Resolver:
    $template: string_to_int_resolver
    $mapping:
      '12': 10
      'true': 1
```

Пример

```
$transformers:
  Resolver:
    $template: string_to_int_array_resolver
    $mapping:
      '12': [10, 2]
      'true': [1]
```

Пример

```
$transformers:
  Resolver:
    $template: string_to_string_resolver
    $mapping:
      '12': '10'
      'true': '1'
```

Пример

```
$transformers:
  Resolver:
    $template: string_to_string_array_resolver
    $mapping:
      '12': ['true', '2']
      'true': ['1']
```

Пример

```
$transformers:
  Resolver:
    $template: strip_string_to_string_resolver
    $mapping:
      '12': 'str1'
      'true': 'str2'
```

2.3. Секция \$loaders

Секция `$loaders` определяет порядок размещения данных, полученных из секций `$transformers` и `$extractors`.

Экстракторы возвращают список с различным количеством элементов, зависящим от типа экстрактора. Каждый элемент списка представляет из себя структуру, которую декларирует экстрактор, или структуру, встроенную в движок процесса расширения аудита.

Обязательные поля секции

В секции `$loaders` необходимо последовательно заполнить следующие обязательные поля:

- `<Имя_лоадера>` — определяет название ладера;
- `$template` — объявляет используемый плагин (для ладера всегда `mapping`). Все последующие использования `$template` объявляются внутри других полей и нужны для указания плагина, с помощью которого будет заполнено это поле. Поле может использоваться для передачи данных из секции `$transformers`. Источником данных для одного плагина может быть другой плагин, указанный в поле `$source`;

- `$target` — объявляет целевую сущность, для которой собираются данные;
- `$kind` — определяет режим работы экстрактора и может принимать следующие значения:
 - `detect` — создание новой сущности. Например, если необходимо создать новую сущность `Core.Software`, выполнится итерация по всем элементам экстрактора с созданием сущностей `Core.Software` по количеству элементов в экстракторе. Значение может использоваться с подтипом `detect-for`, в этом случае итерация выполняется по возвращаемому значению `$origin` со всеми фильтрами;
 - `scan` — обновление существующей сущности. Значение может использоваться с подтипами `scan-one` и `scan-table`. Подтип `scan-one` запускается на единственном возвращаемом экстрактором значении `$origin` со всеми фильтрами: если экстрактор не вернул ни одного элемента, поля, которые заполняет экстрактор, не будут заполнены. Подтип `scan-table` запускается с полем `$table`, значение которого содержит результат запуска `$origin` со всеми фильтрами: все элементы экстрактора собираются в одно поле. Подтип `scan-table` используется вместе с полем `AdditionalProperties` и сериализацией в форматы JSON, YAML, XML.
- `$origin` — указывает источник данных, где содержатся данные для передачи их в поля скана или в функцию обработки. Как правило, представлен в виде объявления `$template: get_extractor_path` и `$name: <Имя созданного файла из блока $extractors>`.
- `$field_map` — указывает все необходимые поля модели актива для заполняемой сущности, а также содержит поле `AdditionalProperties`. В качестве источника данных можно указать плагин в поле `$template`.

Поле `AdditionalProperties` использует тип данных `Dictionary`, а создаваемые в нем поля — `String`. Поля `$serialize` и `$to` используются для преобразования нестроковых данных в секциях `$extractors` и `$transformers` в строковые.

Необязательные поля секции

В секции `$loaders` вы можете заполнить следующие необязательные поля:

- `$parent` — указывает родительскую сущность для заполняемой;
- `$params` — указывает источник для подстановки параметра в раздел `$origin`.

Вы также можете описать условие для определения режима `scan` с помощью необязательного блока `$guard`. Например, условие для наследников класса `Core.Software`, у которых в поле `Name` указано значение `MS Office VBA`, а в поле `Version` — `2019`, будет выглядеть следующим образом:

```
$target: Core.Software
$kind: scan
$parent: OperatingSystem.Windows.WindowsHost
$guard: !eq-str
  $target.Name: MS Office VBA
  $target.Version: '2019'
```

Если предусловие в блоке определится как ложное, режим `scan` не будет определен.

В блоке `$guard` могут быть использованы следующие операторы сравнения:

- `!eq-str` — проверяет равенство строк: если все пары равны, возвращает `True`;
- `!neq-str` — проверяет неравенство строк: если все пары не равны, возвращает `True`;
- `!str-startswith` и `!str-endswith` — проверяют, что значение поля начинается или заканчивается с определенной строки. Если условие выполнено, возвращают `True`;
- `!str-in-str` — проверяет, содержится ли одна строка внутри другой строки.

Также в блоке `$guard` могут использоваться следующие логические операторы:

- `$and` — проверяет, выполняются ли все условия внутри него: если условия выполняются, возвращает `True`;
- `$or` — проверяет, выполняется ли хотя бы одно условие внутри него: если условие выполняется, возвращает `True`;
- `$not` — инвертирует результаты проверки: возвращает `True`, если условие не выполняется, и `False`, если условие выполняется.

Например, если первым условием является, что в полях `$target.Name` и `$target.Version` указаны значения `MS Office VBA` и `'2019'` соответственно, а вторым условием — что значение поля `$target.Architecture` имеет значение, отличное от `x86`, то блок будет выглядеть следующим образом:

```
$guard:
  $or:
    $and:
      !eq-str
      $target.Name: MS Office VBA
    !eq-str
      $target.Version: '2019'
  $not: !eq-str
    $target.Architecture: x86
```

Структура секции `$loaders`

```
$loaders:
  <Имя ладера>:
  - $template: <Имя плагина>
    $target: <Сущность, в которую запишутся данные>
    $kind: <Режим работы экстрактора>
    $parent: <Родительский узел для заполняемой сущности>
    $origin: <Источник, который содержит данные для передачи их в поля скана или
в функцию обработки>
    $template: get_extractor_path
    $name: <Имя экстрактора>
    $params: <Имя параметра, объявленного в экстракторе>: <Данные,
передаваемые в параметр>
```

```

    $field_map: <Поля модели актива для заполняемой сущности и специальное поле
AdditionalProperties>
    <Поле>: <Значение>
    AdditionalProperties: <Поле для добавления в скан полей, отсутствующих в
модели актива>
    <Поле>:
        $serialize: <Источник данных>
        $template: <Имя плагина>
        $to: <Формат файла>

```

Плагины секции

В секции могут быть использованы следующие плагины:

- `mapping` — определяет, что будет генерироваться XML-файл с информацией об активе;
- `apply_transformer` — позволяет применить функцию, созданную в секции `$transformers`, указать входные данные и записать результат в поле модели;
- `apply_resolver` — позволяет применить резолвер, созданный в секции `$transformers`, указать входные данные с помощью поля `$value` и записать результат в поле модели;
- `first_not_empty_string` — позволяет получить первое значение из списка (пустые значения не учитываются);
- `get_extractor_path` — объявляет источник данных для раздела `$origin` с помощью экстрактора из секции `$extractors`.

Пример

```

$loaders:
  VirtualBox:
  - $template: mapping
    $target: Core.Software
    $kind: detect
    $parent: OperatingSystem.UNIX.Linux.LinuxHost
    $origin:
      $template: get_extractor_path
      $name: VirtualBoxUnix
    $field_map:
      Name: VirtualBox
      Vendor: Oracle
      OsFamily: $parent.OsFamily
      Version: 6.1.5
      AdditionalProperties:
        SomeField1:
          $serialize:
            $template: apply_transformer
            $name: get_virtualbox_version

```

```

        $source: $row.Data
    $to: json
    SomeField2: $parent.OsName
    SomeField3:
        $serialize: $parent.IpAddress
        $to: yaml
    SomeField4:
        $template: apply_transformer
        $name: get_virtualbox_version
        $source: $row.Data

```

Пример

```

loaders:
  VirtualBox:
  - $template: mapping
    $target: Software.Oracle.VirtualBox
    $kind: detect
    $parent: OperatingSystem.UNIX.Linux.LinuxHost
    $origin:
      $template: get_extractor_path
      $name: VirtualBoxUnix
    $field_map:
      AdditionalProperties:
        SomeUserField1: $row.StringData
        SomeUserField2:
          $serialize: $row.UserSource
          $to: json
        SomeUserField3:
          $serialize:
            $template: apply_transformer
            $name: get_virtualbox_version
            $source: $row.Data
          $to: json

```

2.4. Примеры заполнения файла расширения аудита

Пример заполнения файла расширения аудита для поиска приложения Telegram с сохранением только первых двух секций версии

```

extractors:
  GetTelegramExeInfo:
    $template: windows_file_info
    $filepath: C:\Users\aivanov\AppData\Roaming\Telegram Desktop\Telegram.exe

```

```

$transformers:
  TelegramTwoSections:
    $template: regexp_search
    $pattern: '(?P<Value>^\d+\\.\\d+).*'
    $schema:
      Value: String
  FirstTwoSections:
    $template: function
    $apply: TelegramTwoSections
    $output: Value
$loaders:
  TelegramExeInfo:
  - $template: mapping
    $target: Core.Software
    $kind: detect
    $parent: OperatingSystem.Windows.WindowsHost
    $origin:
      $template: get_extractor_path
      $name: GetTelegramExeInfo
    $field_map:
      Vendor: $row.CompanyName
      Name: $row.ProductName
      InstallPath: $row.Path
      Version:
        $template: apply_transformer
        $name: FirstTwoSections
        $source: $row.FileVersion

```

Пример заполнения файла расширения аудита для приложения VirtualBox

```

$extractors:
  VirtualBoxRegistry:
    $template: registry_select
    $paths:
      - HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\VirtualBox
      - HKEY_LOCAL_MACHINE\SOFTWARE\Sun\XVM VirtualBox
    $mapping:
      InstallDir: InstallDirectory
    $schema:
      InstallDirectory: String
      Version: String
      VersionExt: String
      SourceRegistryKey: RegistryKey
$transformers:
  VirtualBoxVersionParser:

```

```

    $template: regexp_search
    $pattern: '^.*(?:P<Value>\d+\.\d+\.\d+).*$'
    $schema:
      Value: String
  get_virtualbox_version:
    $template: function
    $apply: VirtualBoxVersionParser
    $output: Value
$loaders:
  VirtualBox:
  - $template: mapping
    $target: Core.Software
    $kind: detect
    $parent: OperatingSystem.Windows.WindowsHost
    $origin:
      $template: get_extractor_path
      $name: VirtualBoxRegistry
    $field_map:
      Name: VirtualBox
      Vendor: Oracle
      OsFamily: Windows
      Architecture: $row.SourceRegistryKey.Arch
      InstallPath: $row.InstallDirectory
      Version:
        $template: apply_transformer
        $name: get_virtualbox_version
        $source:
          $template: first_not_empty_string
          $source:
            - $row.Version
            - $row.VersionExt

```

Пример заполнения файла расширения аудита для поиска приложения Telegram в папке C:\Users

```

$extractors:
  TelegramFileInfo:
    $template: windows_file_info_recursive
    $filemasks:
      - telegram.exe
      - телеграм.exe
    $usesymlinks: false
    $dirs:
      - C:\Users
$loaders:
  Telegram:

```

```
- $template: mapping
  $target: Core.Software
  $kind: detect
  $parent: OperatingSystem.Windows.WindowsHost
  $origin:
    $template: get_extractor_path
    $name: TelegramFileInfo
  $field_map:
    Name: Telegram
    Vendor: Telegram
    OsFamily: Windows
    InstallPath: $row.Path
    Version:
      $template: first_not_empty_string
      $source:
        - $row.ProductVersion
        - $row.ProductVersionEx
```

3. Виды запросов к API

Раздел содержит информацию о протоколе обмена данными между сторонними приложениями и MaxPatrol VM. Обмен данными производится через программный интерфейс продукта (REST API) по протоколу HTTPS.

Для выполнения некоторых запросов необходимо использовать токен доступа. Бессрочный персональный токен доступа формата reference вы можете [создать \(см. раздел 3.1\)](#) в веб-интерфейсе PT MC.

В этом разделе

[Управление токенами доступа \(см. раздел 3.1\)](#)

[Работа с группами активов \(см. раздел 3.2\)](#)

[Получение токена PDQL-запросов \(см. раздел 3.3\)](#)

[Импорт активов из CSV \(см. раздел 3.4\)](#)

[Работа с учетными записями сканирования \(см. раздел 3.5\)](#)

[Работа со сканами \(см. раздел 3.6\)](#)

[Работа с профилями сканирования \(см. раздел 3.7\)](#)

[Управление задачами сканирования \(см. раздел 3.8\)](#)

[Работа с задачами сканирования \(см. раздел 3.9\)](#)

[Мониторинг задач сканирования \(см. раздел 3.10\)](#)

[Журналирование действий пользователя \(см. раздел 3.11\)](#)

[Работа с активами \(см. раздел 3.12\)](#)

[Получение метаданных актива \(см. раздел 3.13\)](#)

[Экспорт таблицы активов в CSV \(см. раздел 3.14\)](#)

[Массовые операции над задачами \(см. раздел 3.15\)](#)

[Работа с выбором задач \(см. раздел 3.16\)](#)

3.1. Управление токенами доступа

Для выполнения запросов к API, требующих аутентификации пользователя, необходимо создать персональный токен доступа формата reference. Токен доступа создается в интерфейсе сервиса управления пользователями и доступом PT Management and Configuration.

Вы можете создать не более 100 токенов. Список всех созданных токенов пользователя доступен на странице **Токены доступа**.

► Чтобы создать токен доступа:

1. В главном меню MaxPatrol VM перейдите в приложение PT Management and Configuration.
2. В главном меню PT MC выберите  → **Токены доступа**.
3. Нажмите **Создать**.
4. Введите название токена.
5. Выберите приложение.
6. Нажмите **Создать**.
7. Сохраните токен, чтобы он был доступен только вам.

Внимание! Скопируйте и сохраните токен сразу: после перехода на другую страницу скопировать созданный токен будет невозможно. Потребуется создавать новый.

Вы также можете изменить токен доступа по кнопке  и удалить токен доступа по кнопке .

3.2. Работа с группами активов

С помощью запросов к API вы можете получить идентификаторы созданных в MaxPatrol VM пользовательских инфраструктур и групп активов. Также вы можете создавать и удалять группы активов.

В этом разделе

[Получение идентификатора инфраструктуры \(см. раздел 3.2.1\)](#)

[Получение идентификатора группы активов \(см. раздел 3.2.2\)](#)

[Создание группы активов \(см. раздел 3.2.3\)](#)

[Проверка статуса создания группы \(см. раздел 3.2.4\)](#)

[Удаление группы активов \(см. раздел 3.2.5\)](#)

3.2.1. Получение идентификатора инфраструктуры

Запрос для получения идентификаторов созданных инфраструктур.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scopes/v2/scopes
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 2. Параметры в теле запроса /api/scopes/v2/scopes

Параметр	Обязательный	Тип данных	Описание
limit	Нет	Number	Максимальное количество элементов инфраструктуры в ответе
offset	Нет	Number	Количество отбрасываемых элементов

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 3. Поля ответа на запрос /api/scopes/v2/scopes

Поле	Тип данных	Описание
id	UUID	Идентификатор инфраструктуры
name	String	Название инфраструктуры
tenantId	UUID	Идентификатор дочерней организации

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/scopes/v2/scopes
```

Ответ:

```
[
  {
    "id": "00000000-0000-0000-0000-000000000005",
    "name": "Инфраструктура по умолчанию",
    "tenantId": "32b17948-5755-4905-b0f3-17db2402256a"
  }
]
```

3.2.2. Получение идентификатора группы активов

Запрос для получения идентификаторов групп активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/assets_temporal_readmodel/v2/groups/hierarchy

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 4. Параметры в теле запроса /api/assets_temporal_readmodel/v2/groups/hierarchy

Параметр	Обязательный	Тип данных	Описание
type	Нет	Enum	Тип группы: – all – любой; – static – статическая; – dynamic – динамическая
groupId	Нет	UUID	Идентификатор группы

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 5. Поля ответа на запрос /api/assets_temporal_readmodel/v2/groups/hierarchy

Поле	Тип данных	Описание
id	UUID	Идентификатор группы
name	String	Имя группы для отображения в интерфейсе
groupType	Enum	Тип группы: – all – любой; – static – статическая; – dynamic – динамическая
isReadOnly	Bool	Возможно ли изменение группы
isRemovable	Bool	Возможно ли удаление группы
isRoot	Bool	Является ли группа корневой
isInvalidPredicate	Bool	–
isSlow	Bool	–

Поле	Тип данных	Описание
treePath	String	Полный путь к группе в списке
children	Array	Список вложенных групп

Возможные коды ошибок и их значения:

- 400 (Bad Request) – синтаксическая ошибка в запросе;
- 401 (Unauthorized) – ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_temporal_readmodel/v2/groups/hierarchy
```

Ответ:

```
[
  {
    "id": "00000000-0000-0000-0000-000000000002",
    "name": "Root",
    "groupType": "static",
    "isReadOnly": true,
    "isRemovable": false,
    "isRoot": true,
    "isInvalidPredicate": false,
    "isSlow": false,
    "treePath": "Root",
    "children": [
      {
        "id": "1685ce7c-e440-0001-0000-000000000005",
        "name": "Import",
        "groupType": "static",
        "isReadOnly": false,
        "isRemovable": true,
        "isRoot": false,
        "isInvalidPredicate": false,
        "isSlow": false,
        "treePath": "Import",
        "children": []
      },
      {
        "id": "1685cf07-70c0-0001-0000-000000000009",
        "name": "Windows",
        "groupType": "dynamic",
        "isReadOnly": false,
        "isRemovable": true,
        "isRoot": false,
        "isInvalidPredicate": false,
        "isSlow": false,
        "treePath": "Windows",
        "children": []
      },
      {
        "id": "00000000-0000-0000-0000-000000000003",
        "name": "Unmanaged hosts",
        "groupType": "dynamic",
        "isReadOnly": true,
        "isRemovable": false,
        "isRoot": false,
        "isInvalidPredicate": false,
        "isSlow": false,

```

```

    "treePath": "Unmanaged hosts",
    "children": []
  }
]
}
]

```

3.2.3. Создание группы активов

Запрос для создания группы активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/assets_processing/v2/groups

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 6. Параметры в теле запроса api/assets_processing/v2/groups

Параметр	Обязательный	Тип данных	Описание
description	Нет	String	Описание группы
groupType	Да	Enum	Тип группы: <ul style="list-style-type: none"> – undefined – не определен; – static – статическая; – dynamic – динамическая
metadata	Нет	Array of objects	Метаданные группы активов
metadata → sourceId	Да	String	Идентификатор источника метаданных. Допустимые значения: <ul style="list-style-type: none"> – userInput
metadata → typeId	Да	String	Тип данных. Допустимые значения: <ul style="list-style-type: none"> – cvss
metadata → payload	Нет	String	Контекстные метрики CVSS v2: <ul style="list-style-type: none"> – ar – требования к доступности: <ul style="list-style-type: none"> • unknown – неизвестны; • ND – не определены; • L – низкие;

Параметр	Обязательный	Тип данных	Описание
			<ul style="list-style-type: none"> • M – средние; • H – высокие; – cdp – вероятность нанесения косвенного ущерба: <ul style="list-style-type: none"> • unknown – неизвестна; • ND – не определена; • N – отсутствует; • L – низкая; • LM – средняя пониженная; • MH – средняя повышенная; • H – высокая; – cr – требования к конфиденциальности: <ul style="list-style-type: none"> • unknown – неизвестны; • ND – не определены; • L – низкие; • M – средние; • H – высокие; – ir – требования к целостности: <ul style="list-style-type: none"> • unknown – неизвестны; • ND – не определены; • L – низкие; • M – средние; • H – высокие; – td – плотность целей: <ul style="list-style-type: none"> • unknown – неизвестна; • ND – не определена; • N – отсутствует; • L – низкая;

Параметр	Обязательный	Тип данных	Описание
			<ul style="list-style-type: none"> • М – средняя; • Н – высокая;
name	Да	String	Название группы
organizationInformation	Нет	Array of objects	Информация об организации
organizationInformation → contactUserId	Нет	String	Идентификатор пользователя, являющегося контактным лицом
organizationInformation → address	Нет	String	Адрес
organizationInfrastructure	Нет	Array of objects	Информация об инфраструктуре
organizationInfrastructure → internetProviders	Нет	String	Провайдеры интернета
organizationInfrastructure → numberOfNodes	Нет	String	Количество узлов на периметре
organizationInfrastructure → registeredDomains	Нет	String	Зарегистрированные домены
organizationInfrastructure → usedNetworkApplications	Нет	String	Используемые веб-приложения
organizationInfrastructure → usedNetworks	Нет	String	Используемые сети
parentId	Да	UUID	Идентификатор родительской группы

Параметр	Обязательный	Тип данных	Описание
predicate	Да для динамических групп	String	PDQL-фильтр активов для динамической группы

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created). Ответ может содержать поля, описанные в таблице ниже.

Таблица 7. Поля ответа на запрос `api/assets_processing/v2/groups`

Поле	Тип данных	Описание
operationId	UUID	Идентификатор операции создания группы

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации;
- 500 (Internal Server Error) — синтаксическая ошибка в запросе.

Пример

Запрос:

```
POST https://localhost/api/assets_processing/v2/groups
{
  "name": "Group name",
  "parentId": "00000000-0000-0000-0000-000000000002",
  "description": null,
  "groupType": "dynamic",
  "predicate": "WindowsHost",
  "metadata": [
    {
      "sourceId": "UserInput",
      "typeId": "Cvss",
      "payload": "{\"td\": \"L\", \"cr\": \"L\", \"cdp\": \"L\", \"ar\": \"L\", \"ir\": \"L\"}"
    }
  ],
  "organizationInformation": {
    "contactUserId": null,
    "address": null
  }
}
```

```

    },
    "organizationInfrastructure": {
      "usedNetworks": null,
      "numberOfNodes": null,
      "usedNetworkApplications": null,
      "internetProviders": null,
      "registeredDomains": null
    }
  }
}

```

Ответ:

```

{
  "operationId": "14a5ca6b-f780-a001-0000-00000000003f"
}

```

3.2.4. Проверка статуса создания группы

Запрос для получения статуса операции по созданию группы активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_processing/v2/groups/operations/<Идентификатор операции>
```

URL запроса должен содержать идентификатор операции по созданию группы активов.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 8. Поля ответа на запрос `api/assets_processing/v2/groups/operations/<Идентификатор операции>`

Поле	Тип данных	Описание
<Идентификатор группы>	UUID	Идентификатор группы

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция по созданию группы выполняется;
- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_processing/v2/groups/operations/168600a8-8f00-a001-0000-000000000008
```

Ответ:

```
"168600a9-8dc0-0001-0000-00000000000d"
```

3.2.5. Удаление группы активов

Запрос для удаления группы активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/assets_processing/v2/groups/removeOperation
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 9. Параметры в теле запроса /api/assets_processing/v2/groups/removeOperation

Параметр	Обязательный	Тип данных	Описание
groupIds	Да	Array [UUID]	Идентификаторы групп

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created). Ответ может содержать поля, описанные в таблице ниже.

Таблица 10. Поля ответа на запрос /api/assets_processing/v2/groups/removeOperation

Поле	Тип данных	Описание
operationId	UUID	Идентификатор операции удаления группы

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
POST https://localhost/api/assets_processing/v2/groups/removeOperation
{
  "groupIds": [
    "14a57a85-8b00-0001-0000-000000000015"
  ]
}
```

Ответ:

```
{
  "operationId": "14cdba4b-c680-a001-0000-00000000384e"
}
```

3.3. Получение токена PDQL-запросов

Запрос для получения токена PDQL-запросов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Время действия токена 30 минут.

Метод и URL запроса:

```
POST <Корневой URL API>/api/assets_temporal_readmodel/v1/assets_grid
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 11. Параметры в теле запроса /api/assets_temporal_readmodel/v1/assets_grid

Параметр	Обязательный	Тип данных	Описание
previousToken	Нет	String	Предыдущий PDQL-токен сеанса пользователя
pdql	Да	String	PDQL-запрос
utcOffset	Нет	String	Сдвиг временной зоны действия от UTC
additionalFilterParameters	Нет	Object	Дополнительные параметры фильтра
selectedGroupIds	Нет	Array	Идентификаторы выбранных групп активов
includeNestedGroups	Да	Bool	Включать ли вложенные группы
executionTime	Нет	String (date-time)	Время выполнения запроса

Параметр	Обязательный	Тип данных	Описание
Дополнительные параметры фильтра			
groupIds	Нет	Array [String]	Список групп, в которых будет происходить выборка
assetIds	Нет	Array	Список активов, по которым будет выполняться запрос

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 12. Поля ответа на запрос `/api/assets_temporal_readmodel/v1/assets_grid`

Поле	Тип данных	Описание
token	String	PDQL-токен для запроса данных
isPotentiallySlow	Bool	Является ли запрос потенциально медленным
hasTimepointPipe	Bool	Есть ли в запросе <code>timepoint</code>
hasTimeseriesPipe	Bool	Есть ли в запросе <code>timeseries</code>
fields	Array [Object]	Описание полей
Описание полей		
name	String	Имя поля
localizedName	String	Локализованное имя поля
type	String (Enum)	Тип поля
isArray	Bool	Является ли коллекцией
origin	String (Enum)	Примененная агрегация

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 500 (Internal Server Error) — внутренняя ошибка сервера;
- 503 (Service Unavailable) — ошибка доступа.

3.4. Импорт активов из CSV

С помощью запросов к API вы можете импортировать в MaxPatrol VM список активов из CSV-файла. Для этого необходимо:

1. Если в модель активов были добавлены пользовательские поля — получить данные этих полей.
2. Создать CSV-файл в соответствии с требуемым форматом.
3. Загрузить CSV-файл со списком активов.
4. Запустить процедуру импорта активов.

В этом разделе

[Получение пользовательских полей \(см. раздел 3.4.1\)](#)

[Запрос формата CSV-файла \(см. раздел 3.4.2\)](#)

[Загрузка CSV-файла \(см. раздел 3.4.3\)](#)

[Выгрузка файла ошибок \(см. раздел 3.4.4\)](#)

[Запуск импорта активов \(см. раздел 3.4.5\)](#)

[Проверка статуса импорта \(см. раздел 3.4.6\)](#)

3.4.1. Получение пользовательских полей

Запрос для получения данных пользовательских полей в модели активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_processing/v2/csv/metadata
```

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 13. Поля ответа на запрос /api/assets_processing/v2/csv/metadata

Поле	Тип	Описание
modelName	String	Название корневой группы параметров в пользовательской модели активов

Поле	Тип	Описание
propertyDeclarations	Array	Свойства пользовательского поля
propertyName	String	Название пользовательского поля
propertyType	String	Тип данных пользовательского поля

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_processing/v2/csv/metadata
```

Ответ:

```
[
  {
    "propertyName": "uf_assetnumber",
    "propertyDeclarations": [
      {
        "modelName": "Core.Host",
        "propertyType": "int"
      }
    ]
  },
  {
    "propertyName": "uf_assetowner",
    "propertyDeclarations": [
      {
        "modelName": "Core.Host",
        "propertyType": "string"
      }
    ]
  },
  {
    "propertyName": "uf_assetrevisiondate",
    "propertyDeclarations": [
      {
        "modelName": "Core.Host",
        "propertyType": "datetime"
      }
    ]
  }
]
```

3.4.2. Запрос формата CSV-файла

Запрос для получения формата CSV-файла.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_processing/v2/csv/example
```

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Тело ответа содержит пример содержимого CSV-файла с данными актива.

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_processing/v2/csv/example
```

Ответ:

```
"typealias";"Fqdn";"Hostname";"Ip";"Mac";"IsVirtual"
"";"dns.somedomain.ru";"w2k3sp2x86s5";"192.168.0.4|182.168.10.1";"00:50:56:A6:0C:36|
00:60:56:A6:0C:36";"false"
```

3.4.3. Загрузка CSV-файла

Запрос для загрузки CSV-файла с данными активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/assets_processing/v2/csv/import_operation
```

Параметры строки запроса описаны в таблице ниже.

Таблица 14. Параметры строки запроса /api/assets_processing/v2/csv/import_operation

Параметр	Обязательный	Тип данных	Описание
scopeId	Да	UUID	Идентификатор инфраструктуры

В запрос необходимо добавить заголовок `Content-Disposition` с описанием CSV-файла, а в тело запроса — CSV-файл с данными активов.

Формат CSV-файла

Файл должен быть представлен в кодировке UTF-8 с BOM. Первая строка файла должна содержать названия полей актива. Вторая и последующие строки — значения полей импортируемых активов (одна строка соответствует одному активу). Значения полей должны быть разделены точкой с запятой. Значения текстовых полей должны быть заключены в кавычки.

Таблица 15. Колонки CSV-файла

Колонка	Тип данных	Описание
Fqdn	String	Полное доменное имя
Hostname	String	Имя узла
Ip	Array [String]	IP-адрес. Поле может содержать несколько значений, которые должны быть разделены вертикальной чертой
IsVirtual	Bool	Является ли виртуальным
Mac	String [String]	MAC-адрес. Поле может содержать несколько значений, которые должны быть разделены вертикальной чертой
typealias	Enum	<p>Тип актива:</p> <ul style="list-style-type: none"> – Alcatel OmniSwitch (под управлением AOS) – <code>omniswitch</code>; – BSD, FreeBSD и macOS – <code>bsd</code>; – Check Point GAiA OS (межсетевой экран) – <code>checkpoint</code>; – Check Point SPLAT (межсетевой экран) – <code>checkpoint</code>; – Cisco ACS – <code>acs</code>; – Cisco ASA (межсетевой экран) – <code>asa</code>; – Cisco FWSM (межсетевой экран) – <code>fwsn</code>; – Cisco IOS – <code>ios</code>; – Cisco IOS XE – <code>ios</code>; – Cisco ISE – <code>ise</code>; – Cisco Nexus – <code>nexus</code>; – Cisco PIX (межсетевой экран) – <code>pix</code>; – FortiNet FortiGate (межсетевой экран) – <code>fortigate</code>; – HPE HP-UX – <code>hp_ux</code>; – Huawei VRP – <code>vrp</code>; – IBM AIX – <code>aix</code>; – Juniper JunOS – <code>junos</code>; – Linux (все семейство ОС) – <code>linux</code>; – Oracle Solaris – <code>solaris</code>;

Колонка	Тип данных	Описание
		<ul style="list-style-type: none"> – Palo Alto Networks PAN-OS (межсетевые экраны) – pan_os; – VMware vSphere Hypervisor (ESXi) – esxi; – Windows – windows
<Пользовательские поля>	–	Добавленные пользователем поля актива, не являющиеся стандартными

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 16. Поля ответа на запрос /api/assets_processing/v2/csv/import_operation

Поле	Тип данных	Описание
id	UUID	Идентификатор операции импорта
isLogFileCreated	Bool	Создан ли файл с ошибками
rowCountExceeded	Bool	Превышает ли количество строк в импортируемом файле установленное ограничение
totalRowCount	Number	Общее количество строк в импортируемом файле
validRowCount	Number	Количество строк с валидными данными активов в импортируемом файле

Возможные коды ошибок и их значения:

- 400 (Bad Request) – синтаксическая ошибка в запросе;
- 401 (Unauthorized) – ошибка аутентификации.

Пример

Запрос:

```
POST https://localhost/api/assets_processing/v2/csv/import_operation?
scopeId=00000000-0000-0000-0000-000000000005
```

Заголовок запроса:

```
Content-Disposition:form-data; name="upfile"; filename="Assets_list.csv"
```

В тело запроса (в представлении form-data) необходимо добавить CSV-файл Assets_list.csv с ключом upfile.

Ответ:

```
{
  "id": "0e372968-e18b-471e-b3b7-ad0d927c2dd9",
  "isLogFileCreated": true,
  "rowCountExceeded": false,
  "validRowCount": 2,
  "totalRowCount": 3
}
```

3.4.4. Выгрузка файла ошибок

Запрос для выгрузки списка ошибок, обнаруженных при импорте CSV-файла со списком активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_processing/v2/csv/import_operation/<Идентификатор операции>/logfile
```

URL запроса должен содержать идентификатор операции импорта CSV-файла.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Заголовок ответа *Content-Disposition* содержит информацию о выгруженном файле с ошибками. Тело ответа содержит список ошибок с указанием номера строки в импортированном CSV-файле. Если файл с ошибками не создан, сервис возвращает код 204 (No Content).

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_processing/v2/csv/import_operation/0e372968-e18b-471e-b3b7-ad0d927c2dd9/logfile
```

Ответ:

```
"3";"Значение [version] некорректно для псевдонима"
```

3.4.5. Запуск импорта активов

Запрос для запуска импорта активов из загруженного CSV-файла.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/assets_processing/v2/csv/import_operation/<Идентификатор операции>/start
```

URL запроса должен содержать идентификатор операции импорта CSV-файла.

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 17. Параметры в теле запроса `api/assets_processing/v2/csv/import_operation/<Идентификатор операции>/start`

Параметр	Обязательный	Тип данных	Описание
groupsId	Да	Array [UUID]	Идентификаторы групп, в которые нужно добавить активы

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
POST https://localhost/api/assets_processing/v2/csv/import_operation/0e372968-e18b-471e-b3b7-ad0d927c2dd9/start
{
  "groupsId": [
    "1499141b-4c00-0001-0000-00000000000d"
  ]
}
```

3.4.6. Проверка статуса импорта

Запрос для получения статуса импорта активов из CSV-файла.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_processing/v2/csv/import_operation/<Идентификатор операции>/state
```

URL запроса должен содержать идентификатор операции импорта CSV-файла.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 18. Поля ответа на запрос `api/assets_processing/v2/csv/import_operation/<Идентификатор файла>/state`

Поле	Тип данных	Описание
<code>errorModel</code>	Array	Данные ошибок
<code>errorModel → code</code>	Number (int32)	Код ошибки
<code>errorModel → errors</code>	Array	Название ошибки
<code>errorModel → message</code>	String	Текст ошибки
<code>state</code>	Enum	Статус импорта: <ul style="list-style-type: none"> – <code>completed</code> – завершен; – <code>failed</code> – не выполнен; – <code>inprogress</code> – выполняется; – <code>validated</code> – на проверке
<code>succeedCount</code>	Number	Количество импортированных активов
<code>updatedGroups</code>	Array	Данные обновленной группы
<code>updatedGroups → displayName</code>	String	Название группы
<code>updatedGroups → id</code>	UUID	Идентификатор группы

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации.

Пример

Запрос:

```
GET https://localhost/api/assets_processing/v2/csv/import_operation/0e372968-  
e18b-471e-b3b7-ad0d927c2dd9/state
```

Ответ:

```
{  
  "state": "completed",  
  "succeedCount": 1,  
  "updatedGroups": [  
    {  
      "id": "149f37fa-c2c0-0001-0000-000000000011",  
      "displayName": "Import"  
    }  
  ],  
  "errorModel": null  
}
```

3.5. Работа с учетными записями сканирования

С помощью запросов к API вы можете добавлять учетные записи сканирования с типами «логин — пароль», «пароль» и «сертификат», получать и обновлять информацию об учетных записях с такими типами, а также получать список всех учетных записей и их меток и удалять учетные записи.

В этом разделе

[Добавление учетной записи типа «логин — пароль» \(см. раздел 3.5.1\)](#)

[Получение учетной записи типа «логин — пароль» \(см. раздел 3.5.2\)](#)

[Обновление учетной записи типа «логин — пароль» \(см. раздел 3.5.3\)](#)

[Добавление учетной записи типа «пароль» \(см. раздел 3.5.4\)](#)

[Получение учетной записи типа «пароль» \(см. раздел 3.5.5\)](#)

[Обновление учетной записи типа «пароль» \(см. раздел 3.5.6\)](#)

[Добавление учетной записи типа «сертификат» \(см. раздел 3.5.7\)](#)

[Получение учетной записи типа «сертификат» \(см. раздел 3.5.8\)](#)

[Обновление учетной записи типа «сертификат» \(см. раздел 3.5.9\)](#)

[Получение всех учетных записей \(см. раздел 3.5.10\)](#)

[Удаление учетной записи \(см. раздел 3.5.11\)](#)

[Получение списка меток учетных записей \(см. раздел 3.5.12\)](#)

3.5.1. Добавление учетной записи типа «логин — пароль»

Запрос для добавления учетной записи типа «логин — пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/v3/credentials/login_passwords

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 19. Параметры в теле запроса /api/v3/credentials/login_passwords

Параметр	Обязательный	Тип данных	Описание
name	Да	String	Название учетной записи
description	Нет	String	Описание
credentialTags	Да	Array[String]	Метки учетной записи — если она создается для определенных методов сбора
domain	Нет	String	Имя домена — если для доступа к источнику используется доменная учетная запись
login	Да	String	Логин учетной записи
password	Нет	String	Пароль учетной записи

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 20. Поля ответа на запрос /api/v3/credentials/login_passwords

Поле	Тип данных	Описание
id	String	Идентификатор учетной записи

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

Пример

Запрос:

```
POST https://localhost/api/v3/credentials/login_passwords
{
  "name": "user",
  "login": "login",
  "credentialTags": []
}
```

3.5.2. Получение учетной записи типа «логин — пароль»

Запрос для получения учетной записи типа «логин — пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v3/credentials/login_passwords/{id}
```

URL запроса содержит path-параметр id — идентификатор учетной записи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 21. Параметры ответа на запрос /api/v3/credentials/login_passwords/{id}

Параметр	Обязательный	Тип данных	Описание
type	Да	String	Тип учетной записи
id	Да	String	Идентификатор учетной записи
name	Да	String	Название учетной записи

Параметр	Обязательный	Тип данных	Описание
description	Да	String	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
domain	Нет	String	Имя домена — если для доступа к источнику используется доменная учетная запись
login	Да	String	Логин учетной записи

Возможные коды ошибок и их значения:

- 404 (Not Found) — учетная запись с таким идентификатором не найдена.

3.5.3. Обновление учетной записи типа «логин — пароль»

Запрос для обновления учетной записи типа «логин — пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/v3/credentials/login_passwords/{id}

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры строки запроса описаны в таблице ниже.

Таблица 22. Параметры строки запроса /api/v3/credentials/login_passwords/{id}

Поле	Обязательный	Тип данных	Описание
name	Да	String	Название учетной записи
description	Да	String	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора

Поле	Обязательный	Тип данных	Описание
domain	Нет	String	Имя домена — если для доступа к источнику используется доменная учетная запись
login	Да	String	Логин учетной записи
shouldResetPassword	Да	Bool	Сбросить ли пароль от учетной записи
password	Да	String	Пароль учетной записи

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content). Ответ не содержит полей.

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 404 (Not Found) — учетная запись с таким идентификатором не найдена.

3.5.4. Добавление учетной записи типа «пароль»

Запрос для добавления учетной записи типа «пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/v3/credentials/passwords_only
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 23. Параметры в теле запроса /api/v3/credentials/passwords_only

Параметр	Обязательный	Тип данных	Описание
name	Да	String	Название учетной записи
description	Да	String	Описание

Параметр	Обязательный	Тип данных	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
password	Да	String	Пароль учетной записи

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 24. Поля ответа на запрос `/api/v3/credentials/passwords_only`

Поле	Тип данных	Описание
id	String	Идентификатор созданной учетной записи

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.5.5. Получение учетной записи типа «пароль»

Запрос для получения учетной записи типа «пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v3/credentials/passwords_only/{id}
```

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 25. Параметры ответа на запрос `/api/v3/credentials/passwords_only/{id}`

Поле	Тип данных	Описание
type	String	Тип учетной записи

Поле	Тип данных	Описание
id	String	Идентификатор учетной записи
name	String	Название учетной записи
description	String	Описание
credentialTags	String	Метки учетной записи — если она создается для определенных методов сбора

Возможные коды ошибок и их значения:

- 404 (Not Found) — учетная запись с таким идентификатором не найдена.

3.5.6. Обновление учетной записи типа «пароль»

Запрос для обновления учетной записи типа «пароль».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/v3/credentials/passwords_only/{id}

URL запроса содержит path-параметр id — идентификатор учетной записи.

Параметры строки запроса описаны в таблице ниже.

Таблица 26. Параметры строки запроса /api/v3/credentials/passwords_only/{id}

Поле	Обязательный	Тип данных	Описание
name	Да	String	Название учетной записи
description	Да	String	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
shouldResetPassword	Да	Bool	Сбросить ли пароль от учетной записи
password	Да	String	Пароль учетной записи

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content). Ответ не содержит полей.

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 404 (Not Found) — учетная запись с таким идентификатором не найдена.

3.5.7. Добавление учетной записи типа «сертификат»

Запрос для добавления учетной записи типа «сертификат».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/v3/credentials/certificates

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 27. Параметры в теле запроса /api/v3/credentials/certificates

Параметр	Обязательный	Тип данных	Описание
name	Да	String	Название учетной записи
description	Да	String	Описание
credentialTags	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
login	Нет	String	Логин учетной записи
password	Нет	String	Пароль учетной записи
certificate	Да	String	Путь к файлу сертификата

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 28. Поля ответа на запрос `/api/v3/credentials/certificates`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор учетной записи

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.5.8. Получение учетной записи типа «сертификат»

Запрос для получения учетной записи типа «сертификат».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v3/credentials/certificates/{id}
```

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 29. Параметры ответа на запрос `/api/v3/credentials/certificates/{id}`

Поле	Тип данных	Описание
<code>type</code>	String	Тип учетной записи
<code>id</code>	String	Идентификатор учетной записи
<code>name</code>	String	Название учетной записи
<code>description</code>	String	Описание
<code>login</code>	String	Логин учетной записи
<code>credentialTags</code>	String	Метки учетной записи — если она создается для определенных методов сбора

Возможные коды ошибок и их значения:

- 404 (Not Found) — учетная запись с таким идентификатором не найдена.

3.5.9. Обновление учетной записи типа «сертификат»

Запрос для обновления учетной записи типа «сертификат».

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/v3/credentials/certificates/{id}

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры строки запроса описаны в таблице ниже.

Таблица 30. Параметры строки запроса /api/v3/credentials/certificates/{id}

Поле	Обязательный	Тип данных	Описание
<code>name</code>	Да	String	Название учетной записи
<code>description</code>	Да	String	Описание
<code>credentialTags</code>	Нет	String	Метки учетной записи — если она создается для определенных методов сбора
<code>login</code>	Нет	String	Логин учетной записи
<code>shouldResetCertificatePassword</code>	Да	Bool	Сбросить ли пароль от учетной записи
<code>password</code>	Да	String	Пароль учетной записи
<code>certificate</code>	Да	String	Путь к файлу сертификата

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content). Ответ не содержит полей.

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 404 (Not Found) — учетная запись с таким идентификатором не найдена.

3.5.10. Получение всех учетных записей

Запрос для получения всех учетных записей в краткой форме.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/v3/credentials

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 31. Параметры в теле запроса /api/v3/credentials

Поле	Обязательный	Тип данных	Описание
credentialTags	Нет	String	Метки учетных записей — если они созданы для определенных методов сбора

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 32. Параметры ответа на запрос /api/v3/credentials

Поле	Тип данных	Описание
type	String	Тип учетной записи
id	String	Идентификатор учетной записи
name	String	Название учетной записи
description	String	Описание
credentialTags	String	Метки учетной записи — если она создается для определенных методов сбора

3.5.11. Удаление учетной записи

Запрос для удаления учетной записи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/v3/credentials/{id}
```

URL запроса содержит path-параметр `id` — идентификатор учетной записи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content). Ответ не содержит полей.

Возможные коды ошибок и их значения:

— 404 (Not Found) — учетная запись с таким идентификатором не найдена.

3.5.12. Получение списка меток учетных записей

Запрос для получения списка меток учетных записей.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v3/credential_tags
```

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ содержит список меток учетных записей.

3.6. Работа со сканами

С помощью запросов к API вы можете получать метаданные сырого и обработанного скана по идентификатору, получать содержание сырого и обработанного скана, получать коллекцию данных сырых и обработанных сканов, получать пакет информации о сырых и обработанных сканах, а также добавлять новые сканы.

В этом разделе

[Добавление нового скана \(см. раздел 3.6.1\)](#)

[Получение метаданных скана по идентификатору \(см. раздел 3.6.2\)](#)

[Получение содержания обработанного скана \(см. раздел 3.6.3\)](#)

[Получение коллекции данных обработанных сканов \(см. раздел 3.6.4\)](#)

[Получение пакета информации об обработанных сканах \(см. раздел 3.6.5\)](#)

[Получение метаданных сырого скана по идентификатору \(см. раздел 3.6.6\)](#)

[Получение содержания сырого скана в формате XML \(см. раздел 3.6.7\)](#)

[Получение коллекции данных сырых сканов \(см. раздел 3.6.8\)](#)

[Получение пакета информации о сырых сканах \(см. раздел 3.6.9\)](#)

3.6.1. Добавление нового скана

Запрос для добавления нового скана.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
PUT <Корневой URL API>/api/v1/scans/{scanId}
```

URL запроса содержит path-параметр `scanId` – идентификатор скана.

Параметры строки запроса описаны в таблице ниже.

Таблица 33. Параметры строки запроса `/api/v1/scans/{scanId}`

Параметр	Обязательный	Тип данных	Описание
<code>source</code>	Да	String	Источник сканирования
<code>scopeId</code>	Да	String	Идентификатор инфраструктуры
<code>time</code>	Да	String	Время сканирования
<code>jobId</code>	Нет	String	Идентификатор задачи сканирования
<code>noTtl</code>	Нет	Bool	Отключено ли устаревание актива
<code>replaceEntities</code>	Нет	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию – <code>false</code>
<code>createOnly</code>	Нет	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию – <code>false</code>

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 34. Параметры в теле запроса `/api/v1/scans/{scanId}`

Параметр	Обязательный	Тип данных	Описание
<code>content</code>	Да	String	Тело скана

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (Everything is OK. Scan accepted).

Возможные коды ошибок и их значения:

- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

3.6.2. Получение метаданных скана по идентификатору

Запрос для получения метаданных скана по идентификатору.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/{scanId}
```

URL запроса содержит path-параметр `scanId` — идентификатор скана.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 35. Поля ответа на запрос `/api/v1/scans/{scanId}`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор скана
<code>timeStamp</code>	String	Время сканирования
<code>source</code>	String	Источник сканирования
<code>scopeId</code>	String	Идентификатор инфраструктуры
<code>jobId</code>	String	Идентификатор задачи сканирования
<code>noTtl</code>	Bool	Отключено ли устаревание актива
<code>replaceEntities</code>	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <code>false</code>
<code>createOnly</code>	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>
<code>type</code>	String	Тип скана
<code>isValid</code>	Bool	Валиден ли скан
<code>orderedId</code>	Integer	Порядковый номер скана

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

3.6.3. Получение содержания обработанного скана

Запрос для получения содержания валидного обработанного скана в формате XML.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/{scanId}/content
```

URL запроса содержит path-параметр `scanId` — идентификатор скана.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

3.6.4. Получение коллекции данных обработанных сканов

Запрос для получения коллекции данных обработанных сканов начиная с указанного времени.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans
```

Параметры строки запроса описаны в таблице ниже.

Таблица 36. Параметры строки запроса `/api/v1/scans`

Параметр	Обязательный	Тип данных	Описание
<code>fromDate</code>	Да	String	Дата, начиная с которой нужно получить данные
<code>offset</code>	Да	Integer	Смещение от начала результатов выборки
<code>limit</code>	Да	Integer	Размер страницы выдачи
<code>scopeId</code>	Нет	String	Идентификатор инфраструктуры сканирования

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 37. Поля ответа на запрос /api/v1/scans

Поле	Тип данных	Описание
id	String	Идентификатор скана
timeStamp	String	Время сканирования
source	String	Источник сканирования
scopeId	String	Идентификатор инфраструктуры
jobId	String	Идентификатор задачи сканирования
noTtl	Bool	Отключено ли устаревание актива
replaceEntities	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <i>false</i>
createOnly	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <i>false</i>
type	String	Тип скана
isValid	Bool	Валиден ли скан
orderedId	Integer	Порядковый номер скана

Возможные коды ошибок и их значения:

- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

3.6.5. Получение пакета информации об обработанных сканах

Запрос для получения пакета информации об обработанных сканах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/packages
```

Параметры строки запроса описаны в таблице ниже.

Таблица 38. Параметры строки запроса /api/v1/scans/packages

Параметр	Обязательный	Тип данных	Описание
fromCheckpoint	Да	Integer	Токен предыдущего пакета

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 39. Поля ответа на запрос /api/v1/scans/packages

Поле	Тип данных	Описание
id	String	Идентификатор скана
timeStamp	String	Время сканирования
source	String	Источник сканирования
scopeId	String	Идентификатор инфраструктуры
jobId	String	Идентификатор задачи сканирования
noTtl	Bool	Отключено ли устаревание актива
replaceEntities	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <code>false</code>
createOnly	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>
type	String	Тип скана
isValid	Bool	Валиден ли скан
orderedId	Integer	Порядковый номер скана

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

3.6.6. Получение метаданных сырого скана по идентификатору

Запрос для получения метаданных сырого скана по идентификатору.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/raw/{scanId}
```

URL запроса содержит path-параметр `scanId` — идентификатор скана.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 40. Поля ответа на запрос `/api/v1/scans/raw/{scanId}`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор скана
<code>timeStamp</code>	String	Время сканирования
<code>source</code>	String	Источник сканирования
<code>scopeId</code>	String	Идентификатор инфраструктуры
<code>jobId</code>	String	Идентификатор задачи сканирования
<code>noTtl</code>	Bool	Отключено ли устаревание актива
<code>replaceEntities</code>	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию — <code>false</code>
<code>createOnly</code>	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию — <code>false</code>
<code>type</code>	String	Тип скана

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

3.6.7. Получение содержания сырого скана в формате XML

Запрос для получения содержания сырого скана в формате XML.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/raw/{scanId}/content
```

URL запроса содержит path-параметр `scanId` — идентификатор скана.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Badly formatted request) — синтаксическая ошибка в запросе.

3.6.8. Получение коллекции данных сырых сканов

Запрос для получения коллекции данных сырых сканов начиная с указанного времени.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/raw
```

Параметры строки запроса описаны в таблице ниже.

Таблица 41. Параметры строки запроса `/api/v1/scans/raw`

Параметр	Обязательный	Тип данных	Описание
<code>fromDate</code>	Да	String	Дата, начиная с которой нужно получить данные
<code>offset</code>	Да	Integer	Смещение от начала результатов выборки
<code>limit</code>	Да	Integer	Размер страницы выдачи
<code>scopeId</code>	Нет	String	Идентификатор инфраструктуры сканирования

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 42. Поля ответа на запрос `/api/v1/scans/raw`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор скана

Поле	Тип данных	Описание
timeStamp	String	Время сканирования
source	String	Источник сканирования
scopeId	String	Идентификатор инфраструктуры
jobId	String	Идентификатор задачи сканирования
noTtl	Bool	Отключено ли устаревание актива
replaceEntities	Bool	Замещаются ли все сущности при мердже. Значение по умолчанию – <code>false</code>
createOnly	Bool	Помечен ли скан как предназначенный только для создания актива. Значение по умолчанию – <code>false</code>
type	String	Тип скана

Возможные коды ошибок и их значения:

- 400 (Badly formatted request) – синтаксическая ошибка в запросе.

3.6.9. Получение пакета информации о сырых сканах

Запрос для получения пакета информации о сырых сканах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/v1/scans/raw/packages
```

Параметры строки запроса описаны в таблице ниже.

Таблица 43. Параметры строки запроса `/api/v1/scans/raw/packages`

Параметр	Обязательный	Тип данных	Описание
token	Нет	String	Токен предыдущего пакета

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 44. Поля ответа на запрос /api/v1/scans/raw/packages

Поле	Тип данных	Описание
token	String	Токен пакета
scans	Array	Перечень описаний сканов: <ul style="list-style-type: none"> – id – идентификатор скана; – timeStamp – время сканирования; – source – источник сканирования; – scopeId – идентификатор инфраструктуры; – jobId – идентификатор задачи сканирования; – noTtl – отключено ли устаревание актива; – replaceEntities – замещаются ли все сущности при мердже; – createOnly – помечен ли скан как предназначенный только для создания актива. Значение по умолчанию – false; – type – тип скана

Возможные коды ошибок и их значения:

- 204 (No Content) – скан не найден;
- 400 (Badly formatted request) – синтаксическая ошибка в запросе.

3.7. Работа с профилями сканирования

С помощью запросов к API вы можете создавать, обновлять и удалять профили сканирования, получать информацию о профиле сканирования, генерировать профили PenTest, проверять наличие уязвимости при сканировании в режиме пентеста, получать список схем профилей сканирования и локализацию параметров схемы модуля, а также удалять из базы ошибки, возникшие при миграции профиля.

В этом разделе

[Получение списка профилей сканирования \(см. раздел 3.7.1\)](#)

[Создание профиля сканирования \(см. раздел 3.7.2\)](#)

[Получение расширенной информации о профиле сканирования \(см. раздел 3.7.3\)](#)

[Обновление профиля сканирования \(см. раздел 3.7.4\)](#)

[Удаление профиля сканирования \(см. раздел 3.7.5\)](#)

[Генерация профиля PenTest \(см. раздел 3.7.6\)](#)

[Проверка наличия уязвимости при сканировании в режиме пентеста \(см. раздел 3.7.7\)](#)

[Получение списка схем профилей сканирования \(см. раздел 3.7.8\)](#)

[Получение локализации параметров схемы модуля \(см. раздел 3.7.9\)](#)

[Удаление из базы ошибок, возникших при миграции профиля \(см. раздел 3.7.10\)](#)

3.7.1. Получение списка профилей сканирования

Запрос для получения списка профилей сканирования с краткой информацией, с возможностью фильтрации по модулю.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_profiles
```

Параметры строки запроса описаны в таблице ниже.

Таблица 45. Параметры строки запроса /api/scanning/v3/scanner_profiles

Параметр	Обязательный	Тип данных	Описание
moduleId	Нет	String	Идентификатор модуля

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK) и список профилей сканирования с краткой информацией. Ответ может содержать поля, описанные в таблице ниже.

Таблица 46. Поля ответа на запрос /api/scanning/v3/scanner_profiles

Поле	Тип данных	Описание
id	String	Идентификатор профиля сканирования
name	String	Имя профиля сканирования
isSystem	Bool	Является ли системой
baseProfileName	String	Имя базового профиля
moduleName	String	Имя модуля
moduleId	String	Идентификатор модуля
output	String	Путь к профилям сканирования

Поле	Тип данных	Описание
agentRequired	Bool	Необходим ли коллектор
baseProfileId	String	Идентификатор базового профиля
validationErrors	Array	Ошибки валидации: – type – тип ошибки

3.7.2. Создание профиля сканирования

Запрос для создания профиля сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/scanner_profiles

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 47. Параметры в теле запроса /api/scanning/v3/scanner_profiles

Параметр	Обязательный	Тип данных	Описание
profile	Да	Array	Создаваемый профиль: – name – имя профиля, тип string; – description – описание профиля; тип string; – baseProfileId – идентификатор базового профиля; тип string; – overrides – переопределение параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created). Ответ может содержать поля, описанные в таблице ниже.

Таблица 48. Поля ответа на запрос `api/scanning/v3/scanner_profiles`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор созданного профиля

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 49. Поля ответа на запрос `api/scanning/v3/scanner_profiles`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

3.7.3. Получение расширенной информации о профиле сканирования

Запрос для получения расширенной информации о профиле сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_profiles/{profileId}
```

URL запроса содержит path-параметр `profileId` — идентификатор профиля сканирования.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 50. Поля ответа на запрос `/api/scanning/v3/scanner_profiles/{profileId}`

Поле	Тип данных	Описание
<code>output</code>	String	Путь к профилю сканирования
<code>description</code>	String	Описание
<code>isSystem</code>	Bool	Является ли системой
<code>moduleName</code>	String	Имя модуля
<code>moduleId</code>	String	Идентификатор модуля
<code>baseProfileId</code>	String	Идентификатор базового профиля
<code>baseProfileName</code>	String	Имя базового профиля
<code>validationErrors</code>	Array	Ошибки валидации: – <code>type</code> – тип ошибки, тип string
<code>profile</code>	Array	Профиль. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1, additionalProp2, additionalProp3</code>
<code>overrides</code>	Array	Перекрытие параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1, additionalProp2, additionalProp3</code>

Возможные коды ошибок и их значения:

- 400 (Bad request) – синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 51. Поля ответа на запрос `api/scanning/v3/scanner_profiles/{profileId}`

Поле	Тип данных	Описание
errors	Array	Ошибки: <ul style="list-style-type: none"> – error – ошибка: <ul style="list-style-type: none"> • type – тип ошибки; – source – источник ошибки: <ul style="list-style-type: none"> • displayName – отображаемое имя; • hostName – имя узла; • ipAddresses – IP-адрес; – sensitive – признак значимости ошибки
message	String	Сообщение
code	Integer	Код ошибки

3.7.4. Обновление профиля сканирования

Запрос для обновления профиля сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/scanning/v3/scanner_profiles/{profileId}

URL запроса содержит path-параметр `profileId` – идентификатор профиля сканирования.

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 52. Параметры в теле запроса `/api/scanning/v3/scanner_profiles/{profileId}`

Параметр	Обязательный	Тип данных	Описание
profile	Да	Array	Создаваемый профиль: <ul style="list-style-type: none"> – name – имя профиля, тип string; – description – описание профиля; тип string; – overrides – переопределение параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1</code>, <code>additionalProp2</code>, <code>additionalProp3</code>

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (ОК).

Возможные коды ошибок и их значения:

- 204 (No Content) — скан не найден;
- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 53. Поля ответа на запрос `api/scanning/v3/scanner_profiles/{profileId}`

Поле	Тип данных	Описание
errors	Array	Ошибки: <ul style="list-style-type: none"> — error — ошибка: <ul style="list-style-type: none"> • type — тип ошибки; — source — источник ошибки: <ul style="list-style-type: none"> • displayName — отображаемое имя; • hostName — имя узла; • ipAddresses — IP-адрес; — sensitive — признак значимости ошибки
message	String	Сообщение
code	Integer	Код ошибки

3.7.5. Удаление профиля сканирования

Запрос для удаления профиля сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/scanning/v3/scanner_profiles/{profileId}
```

URL запроса содержит path-параметр `profileId` — идентификатор профиля сканирования.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (ОК).

Возможные коды ошибок и их значения:

- 204 (No Content) — профиль сканирования не найден;
- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 54. Поля ответа на запрос `api/scanning/v3/scanner_profiles/{profileId}`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

3.7.6. Генерация профиля PenTest

Запрос для генерации профиля PenTest.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/scanner_profiles/generate_profile_for_pentest

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 55. Параметры в теле запроса `/api/scanning/v3/scanner_profiles/generate_profile_for_pentest`

Параметр	Обязательный	Тип данных	Описание
<code>settings</code>	Да	Object	Параметры: <ul style="list-style-type: none"> — <code>useStandardPorts</code> — признак использования стандартных портов, тип <code>bool</code>; — <code>identifiers</code> — идентификаторы; тип <code>string</code>

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 56. Поля ответа на запрос `/api/scanning/v3/scanner_profiles/generate_profile_for_pentest`

Поле	Тип данных	Описание
<code>baseProfileId</code>	String	Идентификатор базового профиля
<code>profile</code>	Array	Профиль сканирования. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1</code> , <code>additionalProp2</code> , <code>additionalProp3</code>

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 57. Поля ответа на запрос `api/scanning/v3/scanner_profiles/generate_profile_for_pentest`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

3.7.7. Проверка наличия уязвимости при сканировании в режиме пентеста

Запрос для проверки наличия уязвимости при сканировании в режиме пентеста.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/scanning/v3/scanner_profiles/pentest/validate_vulners_for_pentest
```

Параметры строки запроса описаны в таблице ниже.

Таблица 58. Параметры строки запроса /api/scanning/v3/scanner_profiles/pentest/validate_vulners_for_pentest

Параметр	Обязательный	Тип данных	Описание
identifier	Да	String	Идентификатор

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 59. Поля ответа на запрос /api/scanning/v3/scanner_profiles/pentest/validate_vulners_for_pentest

Поле	Тип данных	Описание
result	String	Результат поиска уязвимости

3.7.8. Получение списка схем профилей сканирования

Запрос для получения списка схем профилей сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_profile_schemas
```

Параметры строки запроса описаны в таблице ниже.

Таблица 60. Параметры строки запроса /api/scanning/v3/scanner_profile_schemas

Параметр	Обязательный	Тип данных	Описание
moduleId	Нет	String	Идентификатор модуля

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 61. Поля ответа на запрос `/api/scanning/v3/scanner_profile_schemas`

Поле	Тип данных	Описание
<code>moduleId</code>	String	Идентификатор модуля
<code>dataSchema</code>	Array	Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1, additionalProp2, additionalProp3</code>
<code>formSchema</code>	Array	Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1, additionalProp2, additionalProp3</code>

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 62. Поля ответа на запрос `api/scanning/v3/scanner_profile_schemas`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

3.7.9. Получение локализации параметров схемы модуля

Запрос для получения локализации параметров схемы модуля.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_profile_schemas/localization
```

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 63. Поля ответа на запрос /api/scanning/v3/scanner_profile_schemas/localization

Поле	Тип данных	Описание
additionalProp1	String	—
additionalProp2	String	—
additionalProp3	String	—

3.7.10. Удаление из базы ошибок, возникших при миграции профиля

Запрос для удаления из базы ошибок, возникших при миграции профиля.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/scanning/v3/scanner_profiles/{id}/migration_errors
```

URL запроса содержит path-параметр Id — идентификатор профиля сканирования.

Ответ на запрос

Если файл с ошибками не создан, сервис возвращает 204 (No Content).

3.8. Управление задачами сканирования

С помощью запросов к API вы можете создавать, обновлять, запускать, останавливать и удалять задачи сканирования, запрашивать отчеты по задачам (в том числе по идентификатору), точки сохранения задач, количество задач (в том числе по инфраструктуре), валидировать задачи, а также сбрасывать закладки для задач и удалять из базы ошибки, возникшие при миграции задач.

Контракт используется для поддержания обратной совместимости.

В этом разделе

[Запрос отчета о всех задачах \(см. раздел 3.8.1\)](#)

[Создание задачи \(см. раздел 3.8.2\)](#)

[Запрос количества задач \(см. раздел 3.8.3\)](#)

[Запрос отчета о всех точках сохранения всех задач \(см. раздел 3.8.4\)](#)

[Запрос количества задач по инфраструктуре \(см. раздел 3.8.5\)](#)

[Запрос отчета по задаче по идентификатору \(см. раздел 3.8.6\)](#)

[Обновление задачи \(см. раздел 3.8.7\)](#)

[Удаление задачи \(см. раздел 3.8.8\)](#)

[Валидация задачи \(см. раздел 3.8.9\)](#)

[Сброс закладок для задачи сканирования \(см. раздел 3.8.10\)](#)

[Запуск задачи сканирования \(см. раздел 3.8.11\)](#)

[Остановка задачи \(см. раздел 3.8.12\)](#)

[Приостановка задачи \(см. раздел 3.8.13\)](#)

[Удаление из базы ошибок, возникших при миграции задачи \(см. раздел 3.8.14\)](#)

3.8.1. Запрос отчета о всех задачах

Запрос для получения отчета о всех задачах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_tasks
```

Параметры строки запроса описаны в таблице ниже.

Таблица 64. Параметры строки запроса /api/scanning/v3/scanner_tasks

Параметр	Обязательный	Тип данных	Описание
mainFilter	Нет	String	Основной фильтр
additionalFilter	Нет	String	Дополнительный фильтр

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 65. Поля ответа на запрос `/api/scanning/v3/scanner_tasks`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор задачи
<code>name</code>	String	Название задачи
<code>description</code>	String	Описание задачи
<code>component</code>	Array	Компонент: – <code>type</code> – тип компонента
<code>agent</code>	Array	Коллектор: – <code>id</code> – идентификатор коллектора; – <code>name</code> – имя коллектора
<code>scope</code>	Array	Задача по инфраструктуре: – <code>id</code> – идентификатор задачи; – <code>name</code> – название задачи
<code>profile</code>	Array	Профиль: – <code>id</code> – идентификатор профиля; – <code>name</code> – название профиля
<code>module</code>	Array	Модуль: – <code>id</code> – идентификатор модуля; – <code>name</code> – название модуля
<code>metatransports</code>	Array	Метатранспорты и их параметры. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1</code> , <code>additionalProp2</code> , <code>additionalProp3</code>
<code>status</code>	Array	Статус задачи сканирования
<code>created</code>	String	Дата создания задачи в формате ISO 8601
<code>lastRun</code>	String	Дата последнего запуска в формате ISO 8601
<code>nextRun</code>	String	Дата следующего запуска в формате ISO 8601
<code>lastRunErrorLevel</code>	Array	Уровень ошибки последнего запуска

Поле	Тип данных	Описание
include	Array	<p>Признак включения целей сбора данных:</p> <ul style="list-style-type: none"> – <code>assets</code> – активы: <ul style="list-style-type: none"> • <code>id</code> – идентификатор актива; • <code>name</code> – имя актива; – <code>targets</code> – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – <code>assetsGroups</code> – группы активов: <ul style="list-style-type: none"> • <code>id</code> – идентификатор группы активов; • <code>name</code> – название группы активов
exclude	Array	<p>Признак исключения целей сбора данных:</p> <ul style="list-style-type: none"> – <code>assets</code> – активы: <ul style="list-style-type: none"> • <code>id</code> – идентификатор актива; • <code>name</code> – имя актива; – <code>targets</code> – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – <code>assetsGroups</code> – группы активов: <ul style="list-style-type: none"> • <code>id</code> – идентификатор группы активов; • <code>name</code> – название группы активов
isFqdnPriority	Bool	Является ли сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
validationState	Array	Состояние валидности
lastRunError	Array	<p>Ошибка последнего запуска:</p> <ul style="list-style-type: none"> – <code>type</code> – тип ошибки; – <code>category</code> – категория ошибки
hostDiscovery	Array	<p>Признак обнаружения узлов до начала сбора данных:</p> <ul style="list-style-type: none"> – <code>enabled</code> – признак сканирования отвечающих узлов; – <code>profile</code> – профиль: <ul style="list-style-type: none"> • <code>id</code> – идентификатор профиля; • <code>name</code> – название профиля
hasBookmarks	Bool	Есть ли сохраненное состояние источника

Поле	Тип данных	Описание
credentials	Array	Учетные записи: <ul style="list-style-type: none"> – id – идентификатор учетной записи; – name – название учетной записи
triggerParameters	Array	Базовые параметры срабатывания по таймеру: <ul style="list-style-type: none"> – type – тип; – fromDate – дата начала срабатывания; – toDate – дата окончания срабатывания; – isEnabled – разрешено ли срабатывание по таймеру

3.8.2. Создание задачи

Запрос для создания задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/scanner_tasks

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 66. Параметры в теле запроса /api/scanning/v3/scanner_tasks

Параметр	Обязательный	Тип данных	Описание
name	Да	String	Название задачи
description	Нет	String	Описание задачи
component	Нет	Array of strings	Компонент, к которому подключены коллекторы
component → type	Да	String	Тип компонента: <ul style="list-style-type: none"> – CoreComponent – коллекторы подключены к MP Core; – SiemComponent – коллекторы подключены к MP SIEM Server

Параметр	Обязательный	Тип данных	Описание
component → siemId	Да, если указан тип SiemComponent	String	Идентификатор MP SIEM Server
agent	Нет	String	Идентификатор коллектора
scope	Да	String	Идентификатор задачи по инфраструктуре
profile	Да	String	Идентификатор основного профиля сканирования
include	Да	Array of objects	Признак включения целей сбора данных
include → assets	Да	Array of objects	Активы
include → assets → id	Да	String	Идентификатор актива
include → targets	Да	Array of strings	IP-адреса, FQDN или маски подсетей конкретных сетевых адресов
include → assetsGroups	Да	Array of objects	Группы активов
include → assetsGroups → id	Да	String	Идентификатор группы активов
exclude	Да	Array of objects	Признак исключения целей сбора данных
exclude → assets	Да	Array of objects	Активы
exclude → assets → id	Да	String	Идентификатор актива
exclude → targets	Да	Array of strings	IP-адреса, FQDN или маски подсетей конкретных сетевых адресов
exclude → assetsGroups	Да	Array of objects	Группы активов
exclude → assetsGroups → id	Да	String	Идентификатор группы активов
hostDiscovery	Да	Array of strings	Признак обнаружения узлов модулем HostDiscovery до начала сбора данных

Параметр	Обязательный	Тип данных	Описание
hostDiscovery → enabled	Да	Boolean	Обнаружение выполняется (<code>true</code>) или не выполняется (<code>false</code>)
hostDiscovery → profile	Нет	String	Идентификатор профиля модуля HostDiscovery
triggerParameters	Да	Array of objects	Расписание сканирования
triggerParameters → type	Да	String	Параметры расписания: <ul style="list-style-type: none"> – <code>Daily</code> — ежедневно; – <code>Periodic</code> — раз в указанное число дней; – <code>RepeatableDaily</code> — несколько раз в день; – <code>Weekly</code> — в указанные дни недели; – <code>Fortnightly</code> — в указанные дни с интервалом в две недели; – <code>Monthly</code> — в указанные дни месяца; – <code>CronScheduler</code> — в формате <code>cron</code>
triggerParameters → type → atTime	Да, если указан тип <code>Daily</code> , <code>Periodic</code> , <code>Weekly</code> или <code>Fortnightly</code>	Time-span	Время срабатывания таймера
triggerParameters → type → timeZone	Да, если указан тип <code>Daily</code> , <code>Periodic</code> , <code>Weekly</code> , <code>Fortnightly</code> , <code>Monthly</code> или <code>CronScheduler</code>	String	Часовой пояс

Параметр	Обязательный	Тип данных	Описание
triggerParameters → type → daysOfWeek	Да, если указан тип Daily, Weekly или Fortnightly	String	Дни недели
triggerParameters → type → repeatInterval	Да, если указан тип Periodic или RepeatableDaily	Time-span	Интервал между срабатываниями
triggerParameters → type → onDays	Да, если указан тип Monthly	Integer	Дни месяца
triggerParameters → type → cronExpressions	Да, если указан тип CronScheduler	Array of strings	Время выполнения в формате cron
triggerParameters → fromDate	Нет	Date-time	Дата и время начала действия расписания
triggerParameters → toDate	Нет	Date-time	Дата и время окончания действия расписания
triggerParameters → isEnabled	Да	Boolean	Задача запущена (true) или остановлена (false)
isFqdnPriority	Нет	Boolean	Подключаться к активам по FQDN (true) или по IP-адресу (false)
overrides	Нет	Array of objects	Переопределение параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3

Параметр	Обязательный	Тип данных	Описание
savepoints	Нет	Array of objects	Точки сохранения
savepoints → target	Да	String	Цель задачи сканирования
savepoints → savepoint	Да	String	Точка сохранения в формате 64base
deniedScanSettings	Нет	Array of objects	Запрещенное время сканирования
deniedScanSettings → isEnabled	Да	Boolean	Запрещенные интервалы действуют (true) или отключены (false)
deniedScanSettings → periods	Нет	Array of objects	Параметры запрещенного интервала
deniedScanSettings → periods → daysOfWeek	Да	Array of strings	Дни недели
deniedScanSettings → periods → start	Да	String	Время начала запрещенного интервала в формате hh:mm:ss
deniedScanSettings → periods → end	Да	String	Время окончания запрещенного интервала в формате hh:mm:ss
deniedScanSettings → periods → timeZone	Да	String	Часовой пояс
deniedScanSettings → periods → isWholeDayChosen	Да	Boolean	Запрещенный интервал действует весь день (true) или в указанное время (false)

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 67. Поля ответа на запрос /api/scanning/v3/scanner_tasks

Поле	Тип данных	Описание
id	String	Идентификатор задачи

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 68. Поля ответа на запрос /api/scanning/v3/scanner_tasks

Поле	Тип данных	Описание
errors	Array of objects	Ошибки
errors → error	Array of objects	Ошибка
errors → error → category	Enum	Категория ошибки
errors → error → type	String	Тип ошибки
errors → source	Object	Источник ошибки
errors → source → displayName	String	Отображаемое имя
errors → source → hostName	String	Имя узла
errors → source → ipAddresses	Array of strings	IP-адреса
errors → sensitive	Boolean	Признак значимости ошибки
message	String	Сообщение
code	Integer	Код ошибки

Пример

Запрос:

```
{
  "name": "Service Discovery ежедневно",
  "description": "Service Discovery по группе Unmanaged hosts",
  "agent": "8f6c33a5-858e-4bcb-aaf2-311c70dacebb",
```

```

"scope": "00000000-0000-0000-0000-000000000005",
"profile": "bed53c75-f7eb-41b8-a784-3c343abd8d4c",
"include": {
  "assets": [],
  "targets": [],
  "assetsGroups": [
    {
      "id": "00000000-0000-0000-0000-000000000003"
    }
  ]
},
"exclude": {
  "assets": [],
  "targets": [],
  "assetsGroups": []
},
"hostDiscovery": {
  "enabled": true,
  "profile": "e27ac04f-95fb-41e0-ad0e-185b6713f08c"
},
"triggerParameters": {
  "isEnabled": false,
  "fromDate": "2025-02-17",
  "timeZone": "+03:00",
  "type": "Daily",
  "atTime": "09:00:00",
  "daysOfWeek": [
    "monday",
    "tuesday",
    "wednesday",
    "thursday",
    "friday",
    "saturday",
    "sunday"
  ]
}
}

```

Ответ:

```

{
  "operationId": "14a5ca6b-f780-a001-0000-00000000003f"
}

```

3.8.3. Запрос количества задач

Запрос для получения количества задач.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/scanning/v3/scanner_tasks/count

Параметры строки запроса описаны в таблице ниже.

Таблица 69. Параметры строки запроса /api/scanning/v3/scanner_tasks/count

Параметр	Обязательный	Тип данных	Описание
mainFilter	Нет	String	Основной фильтр
additionalFilter	Нет	String	Дополнительный фильтр

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (ОК). Ответ может содержать поля, описанные в таблице ниже.

Таблица 70. Поля ответа на запрос /api/scanning/v3/scanner_tasks/count

Поле	Тип данных	Описание
total	Integer	Количество задач сканирования

3.8.4. Запрос отчета о всех точках сохранения всех задач

Запрос для получения отчета о всех точках сохранения всех задач.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/scanning/v3/scanner_tasks/bookmarks

Параметры строки запроса описаны в таблице ниже.

Таблица 71. Параметры строки запроса /api/scanning/v3/scanner_tasks/bookmarks

Параметр	Обязательный	Тип данных	Описание
offset	Нет	Integer	Смещение страницы от начала выборки
limit	Нет	Integer	Количество запрашиваемых элементов выборки

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 72. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/bookmarks`

Поле	Тип данных	Описание
<code>items</code>	Array	Представление конкретной точки сохранения: <ul style="list-style-type: none"> – <code>taskId</code> – идентификатор задачи; – <code>scopeId</code> – идентификатор скоупа; – <code>target</code> – цель; – <code>savepoint</code> – точка сохранения
<code>totalItems</code>	Integer	Количество точек сохранения

3.8.5. Запрос количества задач по инфраструктуре

Запрос для получения количества задач по инфраструктуре.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_tasks/scope/{scopeId}/count
```

URL запроса содержит path-параметр `scopeId` – идентификатор задачи по инфраструктуре.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 73. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/scope/{scopeId}/count`

Поле	Тип данных	Описание
<code>total</code>	Integer	Количество задач по инфраструктуре

3.8.6. Запрос отчета по задаче по идентификатору

Запрос для получения отчета по задаче по идентификатору.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 74. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/{id}`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор задачи
<code>name</code>	String	Название задачи
<code>description</code>	String	Описание задачи
<code>include</code>	Array	Признак включения целей сбора данных: <ul style="list-style-type: none"> — <code>assets</code> — активы: <ul style="list-style-type: none"> • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: <ul style="list-style-type: none"> • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов
<code>exclude</code>	Array	Признак исключения целей сбора данных: <ul style="list-style-type: none"> — <code>assets</code> — активы: <ul style="list-style-type: none"> • <code>id</code> — идентификатор актива; • <code>name</code> — имя актива; — <code>targets</code> — IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; — <code>assetsGroups</code> — группы активов: <ul style="list-style-type: none"> • <code>id</code> — идентификатор группы активов; • <code>name</code> — название группы активов

Поле	Тип данных	Описание
isFqdnPriority	Bool	Является ли сканирование сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
component	Array	Компонент: – type – тип компонента
agent	Array	Коллектор: – id – идентификатор коллектора; – name – имя коллектора
module	Array	Модуль: – id – идентификатор модуля; – name – название модуля
scope	Array	Задача по инфраструктуре: – id – идентификатор задачи; – name – название задачи
profile	Array	Профиль сканирования: – id – идентификатор профиля; – name – название профиля
completed	String	Дата завершения задачи
status	Array	Статус задачи сканирования
created	String	Дата создания задачи
lastRun	String	Дата последнего запуска
hasBookmarks	Bool	Сохранено ли состояние источника
lastRunErrorLevel	Array	Уровень ошибки последнего запуска
validationState	Array	Состояние валидности
hostDiscovery	Array	Признак обнаружения узлов до начала сбора данных: – enabled – признак сканирования отвечающих узлов; – profile – профиль: • id – идентификатор профиля; • name – название профиля

Поле	Тип данных	Описание
credentials	Array	Учетные записи: <ul style="list-style-type: none"> – id – идентификатор учетной записи – name – название учетной записи
triggerParameters	Array	Базовые параметры срабатывания по таймеру: <ul style="list-style-type: none"> – type – тип; – fromDate – дата начала срабатывания; – toDate – дата окончания срабатывания; – isEnabled – разрешено ли срабатывание по таймеру
overrides	Array	Переопределение параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3

Возможные коды ошибок и их значения:

404 (Task not found) – задача не найдена.

3.8.7. Обновление задачи

Запрос для обновления задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

PUT <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}

URL запроса содержит path-параметр id – идентификатор задачи.

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 75. Параметры в теле запроса /api/scanning/v3/scanner_tasks/{id}

Параметр	Обязательный	Тип данных	Описание
name	Да	String	Название задачи
description	Нет	String	Описание задачи
component	Нет	Object	Компонент
component → type	Да	String	Тип компонента

Параметр	Обязательный	Тип данных	Описание
agent	Нет	String	Идентификатор коллектора
scope	Да	String	Идентификатор задачи по инфраструктуре
profile	Да	String	Идентификатор основного профиля сканирования
include	Да	Array	Цели сбора данных
include → assets	Да	Array	Активы
include → assets → id	Да	String	Идентификатор актива
include → targets	Да	Array[String]	IP-адреса, FQDN или маски подсетей конкретных сетевых адресов
include → assetsGroups	Да	Array	Группы активов
include → assetsGroups → id	Да	String	Идентификатор группы активов
exclude	Да	Array	Исключения из целей сбора данных
exclude → assets	Да	Array	Активы
exclude → assets → id	Да	String	Идентификатор актива
exclude → targets	Да	Array[String]	IP-адреса, FQDN или маски подсетей конкретных сетевых адресов
exclude → assetsGroups	Да	Array	Группы активов
exclude → assetsGroups → id	Да	String	Идентификатор группы активов
hostDiscovery	Да	Array	Признак обнаружения узлов до начала сбора данных
hostDiscovery → enabled	Да	Bool	Признак сканирования отвечающих узлов
hostDiscovery → profile	Нет	String	Идентификатор профиля
triggerParameters	Да	Array	Базовые параметры срабатывания по таймеру

Параметр	Обязательный	Тип данных	Описание
triggerParameters → type	Да	String	Тип
triggerParameters → fromDate	Нет	String	Дата начала срабатывания
triggerParameters → toDate	Нет	String	Дата окончания срабатывания
triggerParameters → isEnabled	Да	Bool	Разрешено ли срабатывание по таймеру
isFqdnPriority	Нет	Bool	Является ли сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
overrides	Нет	Object	Переопределение параметров профиля
savepoints	Нет	Array	Точки сохранения
savepoints → target	Да	String	Цель задачи сканирования
savepoints → savepoint	Да	String	Точка сохранения в формате base64

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 76. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/{id}`

Поле	Тип данных	Описание
id	String	Идентификатор задачи

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 77. Поля ответа на запрос /api/scanning/v3/scanner_tasks/{id}

Поле	Тип данных	Описание
errors	Array	Ошибки
errors → error	Object	Ошибка
errors → error → type	String	Тип ошибки
errors → error → category	String	Категория ошибки: <ul style="list-style-type: none"> – target – цель сканирования; – scope – задача по инфраструктуре; – profile – профиль; – task – задача; – agent – коллектор; – migration – миграция; – taskgroups – группа задач
errors → source	Object	Источник ошибки
errors → source → displayName	String	Отображаемое имя
errors → source → hostName	String	Имя узла
errors → source → ipAddresses	Array[String]	IP-адреса
errors → sensitive	Bool	Признак значимости ошибки
message	String	Сообщение
code	Integer	Код ошибки

3.8.8. Удаление задачи

Запрос для удаления задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}
```

URL запроса содержит path-параметр `id` – идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (Успешное удаление).

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.

3.8.9. Валидация задачи

Запрос для валидации задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/validation
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (ОК).

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 78. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/{id}/validation`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; • <code>category</code> — категория ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

3.8.10. Сброс закладок для задачи сканирования

Запрос для сброса закладок для задачи сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/cleanBookmarks
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (Успешный сброс).

3.8.11. Запуск задачи сканирования

Запрос для запуска задачи сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/start
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 79. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/{id}/start`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор задачи

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 80. Поля ответа на запрос `/api/scanning/v3/scanner_tasks/{id}/start`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> – <code>error</code> – ошибка: <ul style="list-style-type: none"> • <code>type</code> – тип ошибки; • <code>category</code> – категория ошибки; – <code>source</code> – источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> – отображаемое имя; • <code>hostName</code> – имя узла; • <code>ipAddresses</code> – IP-адрес; – <code>sensitive</code> – признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

3.8.12. Остановка задачи

Запрос для остановки задачи сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/stop
```

URL запроса содержит path-параметр `id` – идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (Успешная остановка).

Возможные коды ошибок и их значения:

– 400 (Bad Request) – синтаксическая ошибка в запросе.

3.8.13. Приостановка задачи

Запрос для приостановки задачи сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/suspend
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

— 400 (Bad Request) — синтаксическая ошибка в запросе.

3.8.14. Удаление из базы ошибок, возникших при миграции задачи

Запрос для удаления из базы ошибок, возникших при миграции задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/scanning/v3/scanner_tasks/{id}/migration_errors
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content).

3.9. Работа с задачами сканирования

С помощью запросов к API вы можете создавать и обновлять задачи сканирования, запрашивать отчеты по задачам (в том числе по идентификатору), а также получать время следующего запуска задачи.

В этом разделе

[Создание задачи \(см. раздел 3.9.1\)](#)

[Запрос отчета о всех задачах \(см. раздел 3.9.2\)](#)

[Получение времени следующего запуска задачи \(см. раздел 3.9.3\)](#)

[Запрос отчета по задаче по идентификатору \(см. раздел 3.9.4\)](#)

[Обновление задачи \(см. раздел 3.9.5\)](#)

3.9.1. Создание задачи

Запрос для создания задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v4/scanner_tasks/create

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 81. Параметры в теле запроса /api/scanning/v4/scanner_tasks/create

Параметр	Обязательный	Тип данных	Описание
task	Нет	Object	Задача
Параметры задачи			
name	Да	String	Название задачи
description	Нет	String	Описание задачи
agents	Нет	Array	Коллекторы и их компоненты, которые выбираются для выполнения задачи (расширенный вариант): <ul style="list-style-type: none"> — <code>agentIds</code> — идентификаторы коллекторов; — <code>components</code> — принадлежность заданных для выполнения задачи коллекторов к конкретному конвейеру или ядру: <ul style="list-style-type: none"> • <code>siemIds</code> — конвейеры, коллекторы которых можно использовать для выполнения подзадачи; • <code>useCoreAgents</code> — возможно ли использовать коллекторы ядра для выполнения подзадачи
scope	Да	String	Задача по инфраструктуре
profile	Да	String	Основной профиль сканирования

Параметр	Обязательный	Тип данных	Описание
include	Да	Array	<p>Признак включения целей сбора данных:</p> <ul style="list-style-type: none"> – assets – активы: <ul style="list-style-type: none"> • id – идентификатор актива; • name – имя актива; – targets – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – assetsGroups – группы активов: <ul style="list-style-type: none"> • id – идентификатор группы активов; • name – название группы активов
exclude	Да	Array	<p>Признак исключения целей сбора данных:</p> <ul style="list-style-type: none"> – assets – активы: <ul style="list-style-type: none"> • id – идентификатор актива; • name – имя актива; – targets – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – assetsGroups – группы активов: <ul style="list-style-type: none"> • id – идентификатор группы активов; • name – название группы активов
hostDiscovery	Да	Array	<p>Признак обнаружения узлов до начала сбора данных:</p> <ul style="list-style-type: none"> – enabled – сканировать ли отвечающие узлы; – profile – профиль: <ul style="list-style-type: none"> • id – идентификатор профиля; • name – название профиля
triggerParameters	Да	Array	<p>Базовые параметры срабатывания по таймеру:</p> <ul style="list-style-type: none"> – type – тип срабатывания; – fromDate – дата начала срабатывания; – toDate – дата окончания срабатывания; – isEnabled – разрешено ли срабатывание по таймеру

Параметр	Обязательный	Тип данных	Описание
<code>isFqdnPriority</code>	Нет	Bool	Является ли сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
<code>overrides</code>	Нет	Array	Переопределение параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: <code>additionalProp1, additionalProp2, additionalProp3</code>
<code>savepoints</code>	Нет	Array	Точки сохранения: <ul style="list-style-type: none"> — <code>target</code> — цель задачи сканирования; — <code>savepoint</code> — точка сохранения

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 82. Поля ответа на запрос `/api/scanning/v4/scanner_tasks/create`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор задачи

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 83. Поля ответа на запрос `/api/scanning/v4/scanner_tasks/create`

Поле	Тип данных	Описание
errors	Array	Ошибки: <ul style="list-style-type: none"> – error – ошибка: <ul style="list-style-type: none"> • type – тип ошибки; • category – категория ошибки; – source – источник ошибки: <ul style="list-style-type: none"> • displayName – отображаемое имя; • hostName – имя узла; • ipAddresses – IP-адрес; – sensitive – признак значимости ошибки
message	String	Сообщение
code	Integer	Код ошибки

3.9.2. Запрос отчета о всех задачах

Запрос для получения отчета о всех задачах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v4/scanner_tasks

Параметры строки запроса описаны в таблице ниже.

Таблица 84. Параметры строки запроса `/api/scanning/v4/scanner_tasks`

Параметр	Обязательный	Тип данных	Описание
mainFilter	Нет	String	Основной фильтр
additionalFilter	Нет	String	Дополнительный фильтр. Значение по умолчанию: all
orderBy	Нет	String	Сортировка в списке
orderDirection	Нет	String	Порядок сортировки
validate	Нет	Bool	Валидировать ли задачи. Значение по умолчанию: true

Параметр	Обязательный	Тип данных	Описание
token	Нет	String	Токен выборки. Возвращает задачи из кэша. Если параметр не задан, задачи возвращаются из базы данных
offset	Нет	Integer	Смещение от начала выборки
limit	Нет	Integer	Количество запрашиваемых элементов выборки. Значение по умолчанию: 1000
filter	Нет	Object	Параметры фильтра
Параметры фильтра			
text	Нет	String	Текст
agents	Нет	Array	Коллектор: <ul style="list-style-type: none"> – agentIds – идентификаторы коллекторов; – autoSelect – включен ли автовыбор
modules	Нет	String	Модули
profiles	Нет	String	Профили
statuses	Нет	String (Enum)	Статус задачи сканирования
credentials	Нет	String	Учетные записи
target → target	Да	Array	Атакованные активы: <ul style="list-style-type: none"> – assets – активы: <ul style="list-style-type: none"> • id – идентификаторы активов; • name – названия активов – targets – атакованные активы; – assetsGroups – группы активов: <ul style="list-style-type: none"> • id – идентификаторы групп активов; • name – названия групп активов
target → type	Да	String	Тип атакованного актива
scopes	Нет	String	Задача по инфраструктуре

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 85. Поля ответа на запрос /api/scanning/v4/scanner_tasks

Поле	Тип данных	Описание
id	String	Идентификатор задачи
name	String	Название задачи
description	String	Описание задачи
agents	Array	<p>Коллекторы и их компоненты, которые выбираются для выполнения задачи (расширенный вариант):</p> <ul style="list-style-type: none"> – agents – коллекторы: <ul style="list-style-type: none"> • id – идентификаторы коллекторов; • name – названия коллекторов – components – принадлежность заданных для выполнения задачи коллекторов к конкретному конвейеру или ядру: <ul style="list-style-type: none"> • siemIds – конвейеры, коллекторы которых можно использовать для выполнения подзадачи; • useCoreAgents – возможно ли использовать коллекторы ядра для выполнения подзадачи
scope	Array	<p>Задача по инфраструктуре:</p> <ul style="list-style-type: none"> – id – идентификатор задачи; – name – название задачи
profile	Array	<p>Профиль сканирования:</p> <ul style="list-style-type: none"> – id – идентификатор профиля; – name – название профиля
module	Array	<p>Модуль:</p> <ul style="list-style-type: none"> – id – идентификатор модуля; – name – название модуля
metatransports	Array	Метатранспорты и их параметры
status	String (Enum)	Статус задачи сканирования
created	String	Дата создания задачи
lastRun	String	Дата последнего запуска
nextRun	String	Дата следующего запуска

Поле	Тип данных	Описание
<code>lastRunErrorLevel</code>	String	Уровень ошибки последнего запуска
<code>include</code>	Array	Признак включения целей сбора данных: <ul style="list-style-type: none"> – <code>assets</code> – активы: <ul style="list-style-type: none"> • <code>id</code> – идентификатор актива; • <code>name</code> – имя актива; – <code>targets</code> – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – <code>assetsGroups</code> – группы активов: <ul style="list-style-type: none"> • <code>id</code> – идентификатор группы активов; • <code>name</code> – название группы активов
<code>exclude</code>	Array	Признак исключения целей сбора данных: <ul style="list-style-type: none"> – <code>assets</code> – активы: <ul style="list-style-type: none"> • <code>id</code> – идентификатор актива; • <code>name</code> – имя актива; – <code>targets</code> – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – <code>assetsGroups</code> – группы активов: <ul style="list-style-type: none"> • <code>id</code> – идентификатор группы активов; • <code>name</code> – название группы активов
<code>isFqdnPriority</code>	Bool	Является ли сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
<code>validationState</code>	Array	Состояние валидности
<code>lastRunError</code>	String	Последняя ошибка запуска: <ul style="list-style-type: none"> – <code>type</code> – тип ошибки; – <code>category</code> – категория ошибки
<code>hostDiscovery</code>	Array	Признак обнаружения узлов до начала сбора данных: <ul style="list-style-type: none"> – <code>enabled</code> – признак сканирования отвечающих узлов; – <code>profile</code> – профиль: <ul style="list-style-type: none"> • <code>id</code> – идентификатор профиля; • <code>name</code> – название профиля

Поле	Тип данных	Описание
hasBookmarks	Bool	Сохранено ли состояние источника
credentials	Array	Учетные записи: <ul style="list-style-type: none"> — <code>id</code> — идентификатор учетной записи — <code>name</code> — название учетной записи
triggerParameters	Array	Базовые параметры срабатывания по таймеру: <ul style="list-style-type: none"> — <code>type</code> — тип; — <code>fromDate</code> — дата начала срабатывания; — <code>toDate</code> — дата окончания срабатывания; — <code>isEnabled</code> — разрешено ли срабатывание по таймеру
token	String	Токен
totalCount	Integer	Общее количество

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 86. Поля ответа на запрос `/api/scanning/v4/scanner_tasks`

Поле	Тип данных	Описание
errors	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; • <code>category</code> — категория ошибки — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес — <code>sensitive</code> — признак значимости ошибки
message	String	Сообщение
code	Integer	Код ошибки

3.9.3. Получение времени следующего запуска задачи

Запрос для получения времени следующего запуска задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v4/scanner_tasks/next_run

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 87. Параметры в теле запроса /api/scanning/v4/scanner_tasks/next_run

Параметр	Обязательный	Тип данных	Описание
schedulerParameters	Да	Object	Базовые параметры срабатывания по таймеру: <ul style="list-style-type: none"> — type — тип срабатывания; — fromDate — дата начала срабатывания; — toDate — дата завершения; — isEnabled — — включен ли запуск задачи по расписанию

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 88. Поля ответа на запрос /api/scanning/v4/scanner_tasks/next_run

Поле	Тип данных	Описание
time	String	Время запуска

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 89. Поля ответа на запрос /api/scanning/v4/scanner_tasks/next_run

Поле	Тип данных	Описание
errors	Array	Ошибки: <ul style="list-style-type: none"> – error – ошибка: <ul style="list-style-type: none"> • type – тип ошибки; • category – категория ошибки; – source – источник ошибки: <ul style="list-style-type: none"> • displayName – отображаемое имя; • hostName – имя узла; • ipAddresses – IP-адрес; – sensitive – признак значимости ошибки
message	String	Сообщение
code	Integer	Код ошибки

3.9.4. Запрос отчета по задаче по идентификатору

Запрос для получения отчета по задаче по ее идентификатору.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v4/scanner_tasks/{id}
```

URL запроса содержит path-параметр `id` – идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 90. Поля ответа на запрос /api/scanning/v4/scanner_tasks/{id}

Поле	Тип данных	Описание
id	String	Идентификатор задачи
name	String	Название задачи
description	String	Описание задачи

Поле	Тип данных	Описание
<code>include</code>	Array	<p>Признак включения целей сбора данных:</p> <ul style="list-style-type: none"> – <code>assets</code> – активы: <ul style="list-style-type: none"> • <code>id</code> – идентификатор актива; • <code>name</code> – имя актива; – <code>targets</code> – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – <code>assetsGroups</code> – группы активов: <ul style="list-style-type: none"> • <code>id</code> – идентификатор группы активов; • <code>name</code> – название группы активов
<code>exclude</code>	Array	<p>Признак исключения целей сбора данных:</p> <ul style="list-style-type: none"> – <code>assets</code> – активы: <ul style="list-style-type: none"> • <code>id</code> – идентификатор актива; • <code>name</code> – имя актива; – <code>targets</code> – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – <code>assetsGroups</code> – группы активов: <ul style="list-style-type: none"> • <code>id</code> – идентификатор группы активов; • <code>name</code> – название группы активов
<code>isFqdnPriority</code>	Bool	Является ли сканирование сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу

Поле	Тип данных	Описание
agents	Array	<p>Коллекторы и их компоненты, которые выбираются для выполнения задачи (расширенный вариант):</p> <ul style="list-style-type: none"> – agents – коллекторы: <ul style="list-style-type: none"> • id – идентификаторы коллекторов; • name – названия коллекторов – components – принадлежность заданных для выполнения задачи коллекторов к конкретному конвейеру или ядру: <ul style="list-style-type: none"> • siemIds – конвейеры, коллекторы которых можно использовать для выполнения подзадачи; • useCoreAgents – возможно ли использовать коллекторы ядра для выполнения подзадачи
module	Array	<p>Модуль:</p> <ul style="list-style-type: none"> – id – идентификатор модуля; – name – название модуля
scope	Array	<p>Задача по инфраструктуре:</p> <ul style="list-style-type: none"> – id – идентификатор задачи; – name – название задачи
profile	Array	<p>Профиль сканирования:</p> <ul style="list-style-type: none"> – id – идентификатор профиля; – name – название профиля
completed	String	Дата завершения задачи
status	String	Статус задачи сканирования
created	String	Дата создания задачи
lastRun	String	Дата последнего запуска
hasBookmarks	Bool	Сохранено ли состояние источника
lastRunErrorLevel	String	Уровень ошибки последнего запуска
validationState	String (Enum)	Состояние валидности

Поле	Тип данных	Описание
hostDiscovery	Array	Признак обнаружения узлов до начала сбора данных: <ul style="list-style-type: none"> – enabled – признак сканирования отвечающих узлов; – profile – профиль: <ul style="list-style-type: none"> • id – идентификатор профиля; • name – название профиля
credentials	Array	Учетные записи: <ul style="list-style-type: none"> – id – идентификатор учетной записи – name – название учетной записи
triggerParameters	Array	Базовые параметры срабатывания по таймеру: <ul style="list-style-type: none"> – type – тип; – fromDate – дата начала срабатывания; – toDate – дата окончания срабатывания; – isEnabled – разрешено ли срабатывание по таймеру
overrides	Array	Переопределение параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3

Возможные коды ошибок и их значения:

- 404 (Not Found) – учетная запись с таким идентификатором не найдена.

3.9.5. Обновление задачи

Запрос для обновления задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
PUT <Корневой URL API>/api/scanning/v4/scanner_tasks/{id}
```

URL запроса содержит path-параметр id – идентификатор задачи.

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 91. Параметры в теле запроса /api/scanning/v4/scanner_tasks/{id}

Параметр	Обязательный	Тип данных	Описание
ScannerTaskParam	Нет	Object	Команда обновления
Параметры задачи			
name	Да	String	Название задачи
description	Нет	String	Описание задачи
agents	Нет	Array	<p>Коллекторы и их компоненты, которые выбираются для выполнения задачи (расширенный вариант):</p> <ul style="list-style-type: none"> – agentIds – идентификаторы коллекторов; – components – принадлежность заданных коллекторов для выполнения задачи к конкретному конвейеру или ядру: <ul style="list-style-type: none"> • siemIds – конвейеры, коллекторы которых можно использовать для выполнения подзадачи; • useCoreAgents – возможно ли использовать коллекторы ядра для выполнения подзадачи
scope	Да	String	Задача по инфраструктуре
profile	Да	String	Основной профиль сканирования
include	Да	Array	<p>Признак включения целей сбора данных:</p> <ul style="list-style-type: none"> – assets – активы: <ul style="list-style-type: none"> • id – идентификатор актива; • name – имя актива; – targets – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – assetsGroups – группы активов: <ul style="list-style-type: none"> • id – идентификатор группы активов; • name – название группы активов

Параметр	Обязательный	Тип данных	Описание
exclude	Да	Array	Признак исключения целей сбора данных: <ul style="list-style-type: none"> – assets – активы: <ul style="list-style-type: none"> • id – идентификатор актива; • name – имя актива; – targets – IP-адреса, FQDN или маски подсетей конкретных сетевых адресов; – assetsGroups – группы активов: <ul style="list-style-type: none"> • id – идентификатор группы активов; • name – название группы активов
hostDiscovery	Да	Array	Признак обнаружения узлов до начала сбора данных: <ul style="list-style-type: none"> – enabled – сканировать ли отвечающие узлы; – profile – профиль: <ul style="list-style-type: none"> • id – идентификатор профиля; • name – название профиля
triggerParameters	Да	Array	Базовые параметры срабатывания по таймеру: <ul style="list-style-type: none"> – type – тип срабатывания; – fromDate – дата начала срабатывания; – toDate – дата окончания срабатывания; – isEnabled – разрешено ли срабатывание по таймеру
isFqdnPriority	Нет	Bool	Является ли сканирование активов по FQDN более приоритетным, чем сканирование по IP-адресу
overrides	Нет	Array	Переопределение параметров профиля. Каждая строка представлена массивом значений столбцов строки в порядке, объявленном в схеме: additionalProp1, additionalProp2, additionalProp3
savepoints	Нет	Array	Точки сохранения: <ul style="list-style-type: none"> – target – цель задачи сканирования; – savepoint – точка сохранения

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 92. Поля ответа на запрос `/api/scanning/v4/scanner_tasks/{id}`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор задачи

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе. Ответ может содержать поля, описанные в таблице ниже.

Таблица 93. Поля ответа на запрос `/api/scanning/v4/scanner_tasks/create`

Поле	Тип данных	Описание
<code>errors</code>	Array	Ошибки: <ul style="list-style-type: none"> — <code>error</code> — ошибка: <ul style="list-style-type: none"> • <code>type</code> — тип ошибки; • <code>category</code> — категория ошибки; — <code>source</code> — источник ошибки: <ul style="list-style-type: none"> • <code>displayName</code> — отображаемое имя; • <code>hostName</code> — имя узла; • <code>ipAddresses</code> — IP-адрес; — <code>sensitive</code> — признак значимости ошибки
<code>message</code>	String	Сообщение
<code>code</code>	Integer	Код ошибки

3.10. Мониторинг задач сканирования

С помощью запросов к API вы можете получать историю запусков задачи, получать список подзадач и информацию об используемых в задаче коллекторах, получать информацию о подзадаче и ошибках при ее выполнении.

В этом разделе

[Получение истории запусков задачи \(см. раздел 3.10.1\)](#)

[Получение списка подзадач в составе задачи \(см. раздел 3.10.2\)](#)

[Получение информации о подзадаче \(см. раздел 3.10.3\)](#)

[Получение ошибок, произошедших при выполнении подзадачи \(см. раздел 3.10.4\)](#)

[Получение информации об используемых в задаче коллекторах \(см. раздел 3.10.5\)](#)

3.10.1. Получение истории запусков задачи

Запрос для получения истории запусков задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v2/scanner_tasks/{taskId}/runs
```

URL запроса содержит path-параметр `taskId` — идентификатор задачи.

Параметры строки запроса описаны в таблице ниже.

Таблица 94. Параметры строки запроса `/api/scanning/v2/scanner_tasks/{taskId}/runs`

Параметр	Обязательный	Тип данных	Описание
<code>withErrors</code>	Нет	Bool	Применять ли значение «с ошибками» в фильтре
<code>timeFrom</code>	Нет	String (time-date)	Начало периода в фильтре в формате ISO 8601
<code>timeTo</code>	Нет	String (time-date)	Время завершения периода в фильтре в формате ISO 8601
<code>offset</code>	Нет	Integer	Смещение от начала результатов выборки
<code>limit</code>	Нет	Integer	Количество запрашиваемых в выборке элементов

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 95. Поля ответа на запрос /api/scanning/v2/scanner_tasks/{taskId}/runs

Поле	Тип данных	Описание
Поля с информацией о запусках задачи		
items	String	Список запусков задачи
totalItems	Integer	Общее количество запусков, попадающих в выборку
Параметры списка запусков задачи		
id	String	Идентификатор запуска
status	Array	Статусы запусков задачи
startedAt	String (time-date)	Время начала запуска в формате ISO 8601
finishedAt	String (time-date)	Время завершения запуска в формате ISO 8601
jobCount	Integer	Количество подзадач в запуске
triggerType	String	Способ запуска: вручную или по расписанию
startedBy	Array	Информация о пользователе, запустившем задачу: <ul style="list-style-type: none"> – id – идентификатор пользователя; – login – логин учетной записи; – firstName – имя; – lastName – фамилия
stoppedBy	Array	Информация о пользователе, остановившем задачу: <ul style="list-style-type: none"> – id – идентификатор пользователя; – login – логин учетной записи; – firstName – имя; – lastName – фамилия
errorStatus	Array	Статус ошибки запуска задачи

Возможные коды ошибок и их значения:

- 400 (Bad request) – синтаксическая ошибка в запросе.

3.10.2. Получение списка подзадач в составе задачи

Запрос для получения списка подзадач в составе задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/scanning/v2/runs/{runId}/jobs

URL запроса содержит path-параметр runId – идентификатор запуска задачи.

Параметры строки запроса описаны в таблице ниже.

Таблица 96. Параметры строки запроса /api/scanning/v2/runs/{runId}/jobs

Параметр	Обязательный	Тип данных	Описание
status	Нет	String	Фильтр по статусам
errorStatus	Нет	Array	Статус ошибки выполнения подзадачи
agentId	Нет	String	Фильтр по коллектору
targetPattern	Нет	String	Поиск по целям
offset	Нет	Integer	Смещение от начала результатов выборки
limit	Нет	Integer	Количество запрашиваемых в выборке элементов
orderby	Нет	String	Сортировка по колонкам startedAt или finishedAt в направлении asc или desc. Пример значения: orderby="startedAt desc"

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 97. Поля ответа на запрос /api/scanning/v2/runs/{runId}/jobs

Поле	Тип данных	Описание
Поля списка подзадач		
items	String	Список подзадач
totalItems	Integer	Общее количество подзадач, попадающих в выборку
Параметры списка подзадач		
id	String	Идентификатор подзадачи
status	Array	Статусы подзадач
errorStatus	Array	Статус ошибки выполнения подзадачи
startedAt	String (time-date)	Время начала подзадачи в формате ISO 8601

Поле	Тип данных	Описание
finishedAt	String (time-date)	Время завершения подзадачи в формате ISO 8601
agent	Array	Информация о коллекторе: <ul style="list-style-type: none"> – id – идентификатор коллектора; – name – наименование; – status – статус
targets	String	Список целей сканирования
profile	Array	Профиль подзадачи: <ul style="list-style-type: none"> – id – идентификатор профиля; – name – наименование профиля

Возможные коды ошибок и их значения:

- 400 (Bad request) – синтаксическая ошибка в запросе.

3.10.3. Получение информации о подзадаче

Запрос для получения информации о подзадаче.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Параметры строки запроса описаны в таблице ниже.

```
GET <Корневой URL API>/api/scanning/v2/jobs/{jobId}
```

URL запроса содержит path-параметр `jobId` – идентификатор подзадачи.

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 98. Поля ответа на запрос `/api/scanning/v2/jobs/{jobId}`

Поле	Тип данных	Описание
id	String	Идентификатор подзадачи
status	Array	Статусы подзадач
errorStatus	Array	Статус ошибки выполнения подзадачи
startedAt	String (time-date)	Время начала подзадачи в формате ISO 8601

Поле	Тип данных	Описание
finishedAt	String (time-date)	Время завершения подзадачи в формате ISO 8601
agent	Array	Информация о коллекторе: <ul style="list-style-type: none"> – id – идентификатор коллектора; – name – наименование; – status – статус
targets	String	Список целей сканирования
profile	Array	Профиль подзадачи: <ul style="list-style-type: none"> – id – идентификатор профиля; – name – наименование профиля

Возможные коды ошибок и их значения:

- 400 (Bad request) – синтаксическая ошибка в запросе.

3.10.4. Получение ошибок, произошедших при выполнении подзадачи

Запрос для получения ошибок, произошедших при выполнении подзадачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Параметры строки запроса описаны в таблице ниже.

```
GET <Корневой URL API>/api/scanning/v2/jobs/{jobId}/job_errors
```

URL запроса содержит path-параметр jobId – идентификатор подзадачи.

Параметры строки запроса описаны в таблице ниже.

Таблица 99. Параметры строки запроса /api/scanning/v2/jobs/{jobId}/job_errors

Параметр	Обязательный	Тип данных	Описание
offset	Нет	Integer	Смещение от начала результатов выборки
limit	Нет	Integer	Количество запрашиваемых в выборке элементов
orderby	Нет	String	Сортировка по колонке occurredAt в направлении asc или desc. Пример значения: orderby="startedAt desc"

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 100. Поля ответа на запрос `/api/scanning/v2/jobs/{jobId}/job_errors`

Поле	Тип данных	Описание
Поля списка ошибок подзадач		
<code>items</code>	String	Список ошибок
<code>totalItems</code>	Integer	Общее количество ошибок, попадающих в выборку
<code>logsDownloadUnavailable</code>	Bool	Возможно ли скачать журнал подзадачи с помощью коллектора
Параметры списка ошибок подзадач		
<code>occurredAt</code>	String	Время ошибки в формате ISO 8601
<code>sourceName</code>	String	Псевдоним источника ошибки
<code>type</code>	String	Тип ошибки, идентификатор шаблона для вывода
<code>parameters</code>	Array	Параметры для подстановки в шаблон
<code>status</code>	String	Статус ошибки

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.

3.10.5. Получение информации об используемых в задаче коллекторах

Запрос для получения информации об используемых в задаче коллекторах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Параметры строки запроса описаны в таблице ниже.

```
GET <Корневой URL API>/api/scanning/v2/runs/{runId}/agents
```

URL запроса содержит path-параметр `runId` — идентификатор запуска задачи.

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 101. Поля ответа на запрос `/api/scanning/v2/runs/{runId}/agents`

Поле	Тип данных	Описание
<code>id</code>	String	Идентификатор коллектора
<code>name</code>	String	Наименование коллектора
<code>status</code>	String	Статус коллектора

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе.

3.11. Журналирование действий пользователя

С помощью запросов к API вы можете получать отчеты о действиях пользователей (в том числе по фильтру), дерево категорий действий, возможные значения дополнительного поля, описания всех дополнительных полей для всех приложений, а также регистрировать события пользователя.

В этом разделе

[Получение дерева категорий действий \(см. раздел 3.11.1\)](#)

[Получение возможного значения дополнительного поля \(см. раздел 3.11.2\)](#)

[Получение описания всех дополнительных полей для всех приложений \(см. раздел 3.11.3\)](#)

[Получение отчетов о действиях пользователей \(см. раздел 3.11.4\)](#)

[Получение отчетов о действиях пользователей по фильтру \(см. раздел 3.11.5\)](#)

[Регистрация событий пользователя \(см. раздел 3.11.6\)](#)

3.11.1. Получение дерева категорий действий

Запрос для получения дерева категорий действий.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>:3334/ptms/api/ual/v2/action_categories
```

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 102. Поля ответа на запрос /api/uai/v2/action_categories

Поле	Тип данных	Описание
domains	Array	Домены объектов
Параметры доменов объектов		
id	String	Идентификатор домена объекта
name	String	Название домена объекта
applicationId	UUID	Идентификатор приложения
applicationName	String	Название приложения
types	Array	Тип объекта
Параметры типа объекта		
id	String	Идентификатор типа объекта
name	String	Название типа объекта

3.11.2. Получение возможного значения дополнительного поля

Запрос для получения возможного значения дополнительного поля.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>:3334/ptms/api/uai/v2/user_actions/applications/
{applicationId}/field_values/{key}
```

URL запроса содержит path-параметры: applicationId — идентификатор приложения и key — ключ поля.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 103. Поля ответа на запрос /api/uai/v2/user_actions/applications/{applicationId}/field_values/{key}

Поле	Тип данных	Описание
type	Array	Тип поля

Поле	Тип данных	Описание
Параметры типа поля		
value	String	Значение
relatedKeyValue	String	Значение поля, к которому относится данное поле

3.11.3. Получение описания всех дополнительных полей для всех приложений

Запрос для получения описания всех дополнительных полей для всех приложений.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>:3334/ptms/api/uai/v2/user_actions/additional_fields

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 104. Поля ответа на запрос /api/uai/v2/user_actions/applications/{applicationId}/field_values/{key}

Поле	Тип данных	Описание
type	Array	Тип поля
Параметры типа поля		
applicationId	String	Идентификатор приложения
fieldDefinitions	Array	Список описаний дополнительных полей
Параметры списка описаний дополнительных полей		
key	String	Ключ
localizedName	String	Локализованное название
isHidden	String	Признак сокрытия поля для отображения в списке дополнительных полей. Значение по умолчанию – false
useForFullTextSearch	Bool	Используются ли значения в полнотекстовом поиске
filter	Array	Описание доступного фильтра по данному полю

Поле	Тип данных	Описание
Параметры фильтра		
type	String	Дискриминатор типа фильтра
selectAllLocalizedName	String	Локализованное название строки «Выбрать все» при открытии фильтра

3.11.4. Получение отчетов о действиях пользователей

Запрос для получения отчетов о действиях пользователей.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>:3334/ptms/api/ual/v2/user_actions
```

Параметры строки запроса описаны в таблице ниже.

Таблица 105. Параметры строки запроса /api/ual/v2/user_actions

Параметр	Обязательный	Тип данных	Описание
userId	Да	String	Идентификатор пользователя для фильтрации. Примечание. Вы можете узнать идентификатор пользователя в приложении PT MC. Для этого необходимо в главном меню приложения выбрать Пользователи → Пользователи и выбрать учетную запись пользователя. Идентификатор вида XX-XXXXXX-XXXX-XXXX-XXXXXXXXXXXX будет указан в адресной строке браузера в качестве значения параметра userId
category	Нет	String	Категория в формате applicationId.domainId.typeId
orderby	Нет	String	Сортировка, поддерживается только по времени. Пример – orderby=\"time desc\"
query	Нет	String	Строка для поиска, введенная пользователем
limit	Да	Integer (int32)	Количество запрашиваемых действий. Должен быть больше 0 и меньше 1000
offset	Нет	Integer (int32)	Смещение относительно начала списка действий

Параметр	Обязательный	Тип данных	Описание
timeFrom	Нет	Long (int64)	Начало периода фильтрации (Unix time). Значение по умолчанию — 0
timeTo	Нет	Long (int64)	Конец периода фильтрации (Unix time). Значение по умолчанию — текущее время

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 106. Поля ответа на запрос /api/ual/v2/user_actions

Поле	Тип данных	Описание
id	UUID	Идентификатор действия
beginDateTime	Date (date-time)	Время действия в формате ISO 8601
beginTime	Long (int64)	Время действия (Unix time)
beginUtcOffset	String	Сдвиг временной зоны действия относительно UTC
duration	String	Длительность
userId	String	Идентификатор пользователя
userLogin	String	Отображаемое имя пользователя
userDomain	String	Домен пользователя
objectDomainId	String	Идентификатор домена объекта
objectDomain	String	Домен объекта
objectTypeId	String	Идентификатор типа объекта
objectType	String	Тип объекта
objectId	String	Идентификатор объекта
objectDisplayName	String	Отображаемое имя объекта
applicationId	String	Идентификатор приложения
applicationName	String	Название приложения
code	String	Код действия
result	String	Результат действия

Поле	Тип данных	Описание
failReason	String	Причина ошибки
details	String	Подробное описание изменений
additionalFields	Object	Значение дополнительных полей событий

Пример

Запрос:

```
GET https://localhost:3334/ptms/api/ual/v2/user_actions?
userId=c3e157b1-329a-45ed-9b1b-f3b5da1b7dc4&limit=5&offset=50
```

3.11.5. Получение отчетов о действиях пользователей по фильтру

Запрос для получения отчетов о действиях пользователей по фильтру.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>:3334/ptms/api/ual/v2/user_actions
```

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 107. Параметры в теле запроса /api/ual/v2/user_actions

Параметр	Обязательный	Тип данных	Описание
filter	Да	UserActionFilter	Фильтр сбора событий
Параметры фильтра			
userIds	Нет	Array [String]	Массив идентификаторов пользователей
categories	Нет	Array [String]	Массив категорий в формате applicationId.domainId.typeId
orderby	Нет	String	Сортировка, поддерживается только по времени. Пример – orderby=\"time desc\"
query	Нет	String	Строка для поиска, введенная пользователем
timeFrom	Нет	Date (date-time)	Дата начала периода фильтрации

Параметр	Обязательный	Тип данных	Описание
timeTo	Нет	Date (date-time)	Дата конца периода фильтрации
additionalFields	Нет	Array	Фильтры по дополнительным полям

Параметры фильтра по дополнительному полю

key	Да	String	Ключ дополнительного поля
value	Да	String	Значение для поиска

Параметры строки запроса описаны в таблице ниже.

Таблица 108. Параметры URL запроса /api/ual/v2/user_actions

Параметр	Обязательный	Тип данных	Описание
limit	Да	Integer	Количество запрашиваемых действий
offset	Нет	Integer	Смещение относительно начала списка действий

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (Success). Ответ может содержать поля, описанные в таблице ниже.

Таблица 109. Поля ответа на запрос /api/ual/v2/user_actions

Поле	Тип данных	Описание
totalItems	Integer	Общее количество событий, удовлетворяющих условиям запроса
items	Array	Список действий пользователя

Параметры списка действий пользователя

id	UUID	Идентификатор действия
beginDateTime	Date (date-time)	Время действия в формате ISO 8601
beginTime	Long (int64)	Время действия (Unix time)
beginUtcOffset	String	Сдвиг временной зоны действия от UTC
duration	String	Длительность

Поле	Тип данных	Описание
userId	String	Идентификатор пользователя
userLogin	String	Отображаемое имя пользователя
userDomain	String	Домен пользователя
objectDomainId	String	Идентификатор домена объекта
objectDomain	String	Домен объекта
objectTypeId	String	Идентификатор типа объекта
objectType	String	Тип объекта
objectId	String	Идентификатор объекта
objectDisplayName	String	Отображаемое имя объекта
applicationId	String	Идентификатор приложения
applicationName	String	Название приложения
code	String	Код действия
result	String	Результат действия
failReason	String	Причина ошибки
details	String	Подробное описание изменений
additionalFields	Object	Значение дополнительных полей событий

3.11.6. Регистрация событий пользователя

Запрос для регистрации событий пользователя.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>:3334/ptms/api/ual/v2/user_events

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 110. Параметры в теле запроса /api/ual/v2/user_events

Параметр	Обязательный	Тип данных	Описание
actions	Да	Array	Список событий пользователя

Параметр	Обязательный	Тип данных	Описание
Параметры списка событий пользователя			
id	Да	UUID	Идентификатор действия
time	Да	Date (date-time)	Время действия
type	Да	String	Тип события
utcOffset	Да	String	Сдвиг временной зоны действия от UTC

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created).

3.12. Работа с активами

С помощью запросов к API вы можете сохранить статус завершенной задачи, отменить задачу обновления паспорта активов, запускать задачи на удаление активов, изменение расположения активов в группах, обновление паспортов активов, а также получать состояние этих задач.

В этом разделе

[Запуск задачи на удаление активов \(см. раздел 3.12.1\)](#)

[Получение состояния задачи по удалению активов \(см. раздел 3.12.2\)](#)

[Запуск задачи на изменение расположения активов в группах \(см. раздел 3.12.3\)](#)

[Получение состояния задачи по обновлению расположения активов в группах \(см. раздел 3.12.4\)](#)

[Запуск задачи на обновление паспортов активов \(см. раздел 3.12.5\)](#)

[Получение состояния задачи по обновлению паспорта активов \(см. раздел 3.12.6\)](#)

[Отмена задачи обновления паспорта активов \(см. раздел 3.12.7\)](#)

[Сохранение статуса завершенной задачи \(см. раздел 3.12.8\)](#)

3.12.1. Запуск задачи на удаление активов

Запрос для запуска задачи на удаление активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/assets_processing/v1/asset_operations/removeAssets

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 111. Параметры в теле запроса /api/assets_processing/v1/asset_operations/removeAssets

Параметр	Обязательный	Тип данных	Описание
assetsIds	Да	String	Идентификаторы удаляемых активов

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created. Возвращаем token для получения состояния задачи). Ответ может содержать поля, описанные в таблице ниже.

Таблица 112. Поля ответа на запрос /api/assets_processing/v1/asset_operations/removeAssets

Поле	Тип данных	Описание
operationId	String	Идентификатор операции

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации;
- 503 (ServiceUnavailable) — ошибка доступа.

3.12.2. Получение состояния задачи по удалению активов

Запрос для получения состояния задачи по удалению активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/assets_processing/v1/asset_operations/removeAssets

Параметры строки запроса описаны в таблице ниже.

Таблица 113. Параметры строки запроса /api/assets_processing/v1/asset_operations/removeAssets

Параметр	Обязательный	Тип данных	Описание
operationId	Да	String	Идентификатор задачи

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 114. Поля ответа на запрос `/api/assets_processing/v1/asset_operations/removeAssets`

Поле	Тип данных	Описание
<code>type</code>	String	Тип
<code>totalCount</code>	Integer	Общее количество активов
<code>succeedCount</code>	Integer	Количество успешных удалений активов
<code>failedCount</code>	Integer	Количество неуспешных удалений активов
<code>errorModel</code>	Integer	Модель ошибок: <ul style="list-style-type: none"> – <code>errors</code> – ошибки: <ul style="list-style-type: none"> • <code>error</code> – ошибка: <code>type</code> – тип ошибки; • <code>source</code> – источник; • <code>sensitive</code> – признак значимости ошибки; – <code>message</code> – сообщение; – <code>code</code> – код ошибки
Параметры источника (source)		
<code>displayName</code>	String	Отображаемое имя
<code>hostName</code>	String	Имя узла
<code>ipAddresses</code>	String	IP-адрес

Возможные коды ошибок и их значения:

- 202 (Accepted) – задача еще выполняется;
- 400 (Bad request) – синтаксическая ошибка в запросе.

3.12.3. Запуск задачи на изменение расположения активов в группах

Запрос для запуска задачи на изменение расположения активов в группах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/assets_processing/v1/asset_operations/updateGroupEntries

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 115. Параметры в теле запроса /api/assets_processing/v1/asset_operations/updateGroupEntries

Параметр	Обязательный	Тип данных	Описание
assetsIds	Да	String	Идентификаторы активов
includeInGroups	Да	String	Включенные в группы активы
excludeFromGroups	Да	String	Исключенные из групп активы

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created. Возвращаем token для получения состояния задачи). Ответ может содержать поля, описанные в таблице ниже.

Таблица 116. Поля ответа на запрос /api/assets_processing/v1/asset_operations/updateGroupEntries

Поле	Тип данных	Описание
operationId	String	Идентификатор операции

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации;
- 503 (Service Unavailable) — ошибка доступа.

3.12.4. Получение состояния задачи по обновлению расположения активов в группах

Запрос для получения состояния задачи по обновлению расположения активов в группах.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/assets_processing/v1/asset_operations/updateGroupEntries

Параметры строки запроса описаны в таблице ниже.

Таблица 117. Параметры строки запроса /api/assets_processing/v1/asset_operations/updateGroupEntries

Параметр	Обязательный	Тип данных	Описание
operationId	Да	String	Идентификатор задачи

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 118. Поля ответа на запрос /api/assets_processing/v1/asset_operations/updateGroupEntries

Поле	Тип данных	Описание
type	String	Тип
totalCount	String	Общее количество
succeedCount	String	Количество успешных задач
failedCount	String	Количество неуспешных задач
errorModel	Array	Данные ошибок
errorModel → errors	Array	Название ошибки: — error — ошибка: <ul style="list-style-type: none"> • type — тип ошибки; — source — источник ошибки: <ul style="list-style-type: none"> • displayName — отображаемое имя; • hostName — имя узла; • ipAddresses — IP-адрес; — sensitive — признак значимости ошибки
errorModel → message	String	Текст ошибки
errorModel → code	Integer	Код ошибки
updatedGroups	Array	Данные обновленной группы
updatedGroups → id	String	Идентификатор группы
updatedGroups → displayName	String	Название группы

Возможные коды ошибок и их значения:

- 202 (Accepted) — задача еще выполняется;
- 400 (Bad request) — синтаксическая ошибка в запросе.

3.12.5. Запуск задачи на обновление паспортов активов

Запрос для запуска задачи на обновление паспортов активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/assets_processing/v1/asset_operations/updatePassports

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 119. Параметры в теле запроса /api/assets_processing/v1/asset_operations/updatePassports

Параметр	Обязательный	Тип данных	Описание
assetsIds	Да	Array	Идентификаторы активов
changes	Да	Array	Изменение: <ul style="list-style-type: none"> — ChangeCommand: <ul style="list-style-type: none"> • changeType — тип изменения
operationId	Нет	UUID	Идентификатор операции

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 201 (Created. Возвращаем token для получения состояния задачи). Ответ может содержать поля, описанные в таблице ниже.

Таблица 120. Поля ответа на запрос /api/assets_processing/v1/asset_operations/updatePassports

Поле	Тип данных	Описание
operationId	String	Идентификатор операции

Возможные коды ошибок и их значения:

- 400 (Bad request) — синтаксическая ошибка в запросе;
- 401 (Unauthorized) — ошибка аутентификации;
- 503 (ServiceUnavailable) — ошибка доступа.

3.12.6. Получение состояния задачи по обновлению паспорта активов

Запрос для получения состояния задачи по обновлению паспорта активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

GET <Корневой URL API>/api/assets_processing/v1/asset_operations/updatePassports

Параметры строки запроса описаны в таблице ниже.

Таблица 121. Параметры строки запроса /api/assets_processing/v1/asset_operations/updatePassports

Параметр	Обязательный	Тип данных	Описание
operationId	Да	String	Идентификатор задачи

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 122. Поля ответа на запрос /api/assets_processing/v1/asset_operations/updatePassports

Поле	Тип данных	Описание
type	String	Тип
totalCount	Integer	Общее количество задач
succeedCount	Integer	Количество успешных задач
failedCount	Integer	Количество неуспешных задач
errorModel	Array	Данные ошибок

Поле	Тип данных	Описание
errorModel → errors	Array	Название ошибки: — error — ошибка: <ul style="list-style-type: none"> • type — тип ошибки; — source — источник ошибки: <ul style="list-style-type: none"> • displayName — отображаемое имя; • hostName — имя узла; • ipAddresses — IP-адрес; — sensitive — признак значимости ошибки
errorModel → message	String	Текст ошибки
errorModel → code	Integer	Код ошибки

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция еще выполняется;
- 400 (Bad request) — синтаксическая ошибка в запросе.

3.12.7. Отмена задачи обновления паспорта активов

Запрос для отмены задачи по обновлению паспорта активов.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/assets_processing/v1/asset_operations/updatePassports/{operationId}
```

URL запроса содержит path-параметр operationId — идентификатор задачи обновления активов.

Параметры в теле запроса отсутствуют.

Ответ на запрос

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция еще выполняется;
- 400 (Bad request) — синтаксическая ошибка в запросе.

3.12.8. Сохранение статуса завершенной задачи

Запрос для сохранения статуса завершенной задачи.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
PUT <Корневой URL API>/api/assets_processing/v1/asset_operations/completed/
{operationType}/{operationId}
```

URL запроса содержит path-параметры `operationType` — тип задачи и `operationId` — идентификатор задачи.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content).

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.13. Получение метаданных актива

Запрос для получения метаданных актива.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets/metadata/{type}
```

URL запроса содержит path-параметр `type` — тип модели метаданных.

Параметры строки запроса описаны в таблице ниже.

Таблица 123. Параметры строки запроса `/api/assets/metadata/{type}`

Параметр	Обязательный	Тип данных	Описание
<code>onlyUserProperties</code>	Нет	Bool	Используются ли только свойства пользователя

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 124. Поля ответа на запрос /api/assets/metadata/{type}

Поле	Тип данных	Описание
name	String	Имя типа модели метаданных
root	String	Корневая группа типа модели метаданных
kind	String	Род типа модели метаданных
title	String	Заголовок типа модели метаданных
description	String	Описание типа модели метаданных
properties	String	Свойства типа модели: <ul style="list-style-type: none"> – <code>isCollection</code> – является ли свойство коллекцией; – <code>name</code> – имя свойства; – <code>kind</code> – род свойства; – <code>type</code> – тип свойства; – <code>title</code> – заголовок свойства; – <code>description</code> – описание свойства

Возможные коды ошибок и их значения:

- 304 (Not modified) – запрашиваемая страница не обновлялась с момента последнего обращения;
- 400 (Invalid asset type) – неверный тип актива, синтаксическая ошибка в запросе.

3.14. Экспорт таблицы активов в CSV

Запрос для экспорта таблицы активов в CSV.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/assets_temporal_readmodel/v1/assets_grid/export
```

Параметры строки запроса описаны в таблице ниже.

Таблица 125. Параметры строки запроса /api/assets_temporal_readmodel/v1/assets_grid/export

Параметр	Обязательный	Тип данных	Описание
pdqlToken	Да	String	Токен (см. раздел 3.3) PDQL-запроса

Параметр	Обязательный	Тип данных	Описание
gridSide	Нет	Enum	Сторона таблицы при наличии группировки по полю

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK).

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе;
- 403 (Forbidden) — доступ к запрашиваемой странице запрещен;
- 404 (Not Found) — учетная запись с таким идентификатором не найдена;
- 500 (Internal Server Error) — внутренняя ошибка сервера.

3.15. Массовые операции над задачами

С помощью запросов к API вы можете запускать, останавливать, приостанавливать и удалять выбранные задачи сканирования, устанавливать и сравнивать расписание запуска задач, сбрасывать закладки для задач сканирования, получать информацию о массовой операции над задачами и удалять созданную массовую операцию.

В этом разделе

[Запуск задач сканирования \(см. раздел 3.15.1\)](#)

[Остановка задач сканирования \(см. раздел 3.15.2\)](#)

[Приостановка задач сканирования \(см. раздел 3.15.3\)](#)

[Удаление задач сканирования \(см. раздел 3.15.4\)](#)

[Установка расписания задач сканирования \(см. раздел 3.15.5\)](#)

[Сравнение расписаний запуска задач \(см. раздел 3.15.6\)](#)

[Сброс закладок для задач сканирования \(см. раздел 3.15.7\)](#)

[Получение информации о массовой операции \(см. раздел 3.15.8\)](#)

[Удаление массовой операции \(см. раздел 3.15.9\)](#)

3.15.1. Запуск задач сканирования

Запрос для запуска выбранных задач сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/task_bulk_operations/start_tasks

Параметры строки запроса описаны в таблице ниже.

Таблица 126. Параметры строки запроса /api/scanning/v3/task_bulk_operations/start_tasks

Параметр	Обязательный	Тип данных	Описание
selectionId	Да	String	Идентификатор выборки задач для использования в массовых операциях

Параметры в теле запроса отсутствуют.

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция выполняется. Ответ содержит идентификатор массовой операции;
- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.15.2. Остановка задач сканирования

Запрос для остановки выбранных задач сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/task_bulk_operations/stop_tasks

Параметры строки запроса описаны в таблице ниже.

Таблица 127. Параметры строки запроса /api/scanning/v3/task_bulk_operations/stop tasks

Параметр	Обязательный	Тип данных	Описание
selectionId	Да	String	Идентификатор выборки задач для использования в массовых операциях

Параметры в теле запроса отсутствуют.

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция выполняется. Ответ содержит идентификатор массовой операции;
- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.15.3. Приостановка задач сканирования

Запрос для приостановки выбранных задач сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/task_bulk_operations/suspend_tasks

Параметры строки запроса описаны в таблице ниже.

Таблица 128. Параметры строки запроса /api/scanning/v3/task_bulk_operations/suspend_tasks

Параметр	Обязательный	Тип данных	Описание
selectionId	Да	String	Идентификатор выборки задач для использования в массовых операциях

Параметры в теле запроса отсутствуют.

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция выполняется. Ответ содержит идентификатор массовой операции;
- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.15.4. Удаление задач сканирования

Запрос для удаления выбранных задач сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/task_bulk_operations/delete_tasks

Параметры строки запроса описаны в таблице ниже.

Таблица 129. Параметры строки запроса /api/scanning/v3/task_bulk_operations/delete_tasks

Параметр	Обязательный	Тип данных	Описание
selectionId	Да	String	Идентификатор выборки задач для использования в массовых операциях

Параметры в теле запроса отсутствуют.

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция выполняется. Ответ содержит идентификатор массовой операции;
- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.15.5. Установка расписания задач сканирования

Запрос для установки расписания выбранных задач сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/task_bulk_operations/update_tasks_schedule

Параметры строки запроса описаны в таблице ниже.

Таблица 130. Параметры строки запроса /api/scanning/v3/task_bulk_operations/update_tasks_schedule

Параметр	Обязательный	Тип данных	Описание
selectionId	Да	String	Идентификатор выборки задач для использования в массовых операциях

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 131. Параметры в теле запроса /api/scanning/v3/task_bulk_operations/update_tasks_schedule

Параметр	Обязательный	Тип данных	Описание
schedulerParameters → trigger	Нет	Object	Базовые параметры запуска задачи по расписанию: <ul style="list-style-type: none"> — type — тип запуска; — fromDate — дата запуска задачи по расписанию; — toDate — дата завершения задачи по расписанию; — isEnabled — включен ли запуск задачи по расписанию

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция выполняется. Ответ содержит идентификатор массовой операции;
- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.15.6. Сравнение расписаний запуска задач

Запрос для сравнения расписания запуска выбранных задач сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

POST <Корневой URL API>/api/scanning/v3/task_bulk_operations/compare_scheduler_tasks

Параметры строки запроса описаны в таблице ниже.

Таблица 132. Параметры строки запроса /api/scanning/v3/task_bulk_operations/compare_scheduler_tasks

Параметр	Обязательный	Тип данных	Описание
selectionId	Да	String	Идентификатор выборки задач для использования в массовых операциях

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 133. Поля ответа на запрос /api/scanning/v3/task_bulk_operations/compare_scheduler_tasks

Поле	Тип данных	Описание
triggersAreSame	Bool	Одинаковы ли расписания запуска для задач, выбранных для использования в массовых операциях
schedulersState	String	Состояние расписания всех задач
schedulerParameters → trigger	Object	Базовые параметры запуска задачи по расписанию: <ul style="list-style-type: none"> — type — тип запуска; — fromDate — дата запуска задачи по расписанию;

Поле	Тип данных	Описание
		<ul style="list-style-type: none"> — <code>toDate</code> — дата завершения задачи по расписанию; — <code>isEnabled</code> — включен ли запуск задачи по расписанию

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.15.7. Сброс закладок для задач сканирования

Запрос для сброса закладок для выбранных задач сканирования.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/scanning/v3/task_bulk_operations/clean_bookmarks
```

Параметры строки запроса описаны в таблице ниже.

Таблица 134. Параметры строки запроса `/api/scanning/v3/task_bulk_operations/clean_bookmarks`

Параметр	Обязательный	Тип данных	Описание
<code>selectionId</code>	Да	String	Идентификатор выборки задач для использования в массовых операциях

Параметры в теле запроса отсутствуют.

Возможные коды ошибок и их значения:

- 202 (Accepted) — операция выполняется. Ответ содержит идентификатор массовой операции;
- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.15.8. Получение информации о массовой операции

Запрос для получения информации о массовой операции.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
GET <Корневой URL API>/api/scanning/v3/task_bulk_operations/{id}
```

URL запроса содержит `path`-параметр `id` — идентификатор массовой операции.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 135. Поля ответа на запрос `/api/scanning/v3/task_bulk_operations/{id}`

Поле	Тип данных	Описание
<code>status</code>	String	Статус массовой операции

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.15.9. Удаление массовой операции

Запрос для удаления массовой операции.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/scanning/v3/task_bulk_operations/{id}
```

URL запроса содержит path-параметр `id` — идентификатор массовой операции.

Параметры в теле запроса отсутствуют.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content).

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.16. Работа с выбором задач

С помощью запросов к API вы можете выбирать задачи для использования в массовых операциях, изменять параметры выбора, отменить выбор задач, а также получать информацию о количестве выбранных задач и доступных операциях с ними по их идентификатору.

В этом разделе

[Выбор задач для использования в массовых операциях \(см. раздел 3.16.1\)](#)

[Изменение параметров выбора задач \(см. раздел 3.16.2\)](#)

[Получение информации о выбранных задачах по их идентификатору \(см. раздел 3.16.3\)](#)

[Отмена выбора задач \(см. раздел 3.16.4\)](#)

3.16.1. Выбор задач для использования в массовых операциях

Запрос для выбора задач для использования в массовых операциях.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
POST <Корневой URL API>/api/scanning/v3/task_selections
```

Параметры строки запроса описаны в таблице ниже.

Таблица 136. Параметры строки запроса /api/scanning/v3/task_selections

Параметр	Обязательный	Тип данных	Описание
mainFilter	Нет	String	Основной фильтр
additionalFilter	Нет	String	Дополнительный фильтр. Значение по умолчанию: all
token	Нет	String	Токен выборки. Возвращает задачи из кэша. Если параметр не задан, задачи возвращаются из базы данных

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 137. Параметры в теле запроса /api/scanning/v3/task_selections

Параметр	Обязательный	Тип данных	Описание
filter	Нет	Object	Параметры фильтра

Параметр	Обязательный	Тип данных	Описание
tasksIds	Да	Array	Список идентификаторов выбранных и невыбранных задач: <ul style="list-style-type: none"> – <code>allSelected</code> – выбраны ли все задачи; – <code>unselected</code> – список идентификаторов невыбранных задач. Если параметр задан, параметр <code>selected</code> не задается; – <code>selected</code> – список идентификаторов выбранных задач. Если параметр задан, параметр <code>unselected</code> не задается
Параметры фильтра			
text	Нет	String	Текст
agents	Нет	Array	Коллектор: <ul style="list-style-type: none"> – <code>agentIds</code> – идентификаторы коллекторов; – <code>autoSelect</code> – включен ли автовыбор
modules	Нет	String	Модули
profiles	Нет	String	Профили
statuses	Нет	String (Enum)	Статус задачи сканирования
credentials	Нет	String	Учетные записи
target → target	Да	Array	Атакованные активы: <ul style="list-style-type: none"> – <code>assets</code> – активы: <ul style="list-style-type: none"> • <code>id</code> – идентификаторы активов; • <code>name</code> – названия активов – <code>targets</code> – атакованные активы; – <code>assetsGroups</code> – группы активов: <ul style="list-style-type: none"> • <code>id</code> – идентификаторы групп активов; • <code>name</code> – названия групп активов
target → type	Да	String	Тип атакованного актива
scopes	Нет	String	Задача по инфраструктуре

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 138. Поля ответа на запрос `/api/scanning/v3/task_selections`

Поле	Тип данных	Описание
<code>selectionId</code>	String	Идентификатор выборки
<code>selectedTasksCount</code>	Integer	Количество выбранных задач
<code>availableOperations</code>	Array	Список доступных операций над задачей: — <code>Operation</code> — операция над задачей
<code>token</code>	String	Токен

Возможные коды ошибок и их значения:

— 400 (Bad Request) — синтаксическая ошибка в запросе.

3.16.2. Изменение параметров выбора задач

Запрос для изменения параметров выбора задач.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
PUT <Корневой URL API>/api/scanning/v3/task_selections/{id}
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Параметры строки запроса описаны в таблице ниже.

Таблица 139. Параметры строки запроса `/api/scanning/v3/task_selections/{id}`

Параметр	Обязательный	Тип данных	Описание
<code>mainFilter</code>	Нет	String	Основной фильтр
<code>additionalFilter</code>	Нет	String	Дополнительный фильтр
<code>token</code>	Нет	String	Токен выборки. Возвращает задачи из кэша. Если параметр не задан, задачи возвращаются из базы данных

Тело запроса может содержать параметры, описанные в таблице ниже.

Таблица 140. Параметры в теле запроса `/api/scanning/v3/task_selections/{id}`

Параметр	Обязательный	Тип данных	Описание
<code>filter</code>	Нет	Object	Параметры фильтра
<code>tasksIds</code>	Да	Array	Список идентификаторов выбранных и невыбранных задач: <ul style="list-style-type: none"> – <code>allSelected</code> – выбраны ли все задачи; – <code>unselected</code> – список идентификаторов невыбранных задач. Если параметр задан, параметр <code>selected</code> не задается; – <code>selected</code> – список идентификаторов выбранных задач. Если параметр задан, параметр <code>unselected</code> не задается
Параметры фильтра			
<code>text</code>	Нет	String	Текст
<code>agents</code>	Нет	Array	Коллектор: <ul style="list-style-type: none"> – <code>agentIds</code> – идентификаторы коллекторов; – <code>autoSelect</code> – включен ли автовыбор
<code>modules</code>	Нет	String	Модули
<code>profiles</code>	Нет	String	Профили
<code>statuses</code>	Нет	String (Enum)	Статус задачи сканирования
<code>credentials</code>	Нет	String	Учетные записи
<code>target → target</code>	Да	Array	Атакованные активы: <ul style="list-style-type: none"> – <code>assets</code> – активы: <ul style="list-style-type: none"> • <code>id</code> – идентификаторы активов; • <code>name</code> – названия активов – <code>targets</code> – атакованные активы; – <code>assetsGroups</code> – группы активов: <ul style="list-style-type: none"> • <code>id</code> – идентификаторы групп активов; • <code>name</code> – названия групп активов
<code>target → type</code>	Да	String	Тип атакованного актива

Параметр	Обязательный	Тип данных	Описание
scopes	Нет	String	Задача по инфраструктуре

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 141. Поля ответа на запрос `/api/scanning/v3/task_selections/{id}`

Поле	Тип данных	Описание
selectionId	String	Идентификатор выборки
selectedTasksCount	Integer	Количество выбранных задач
availableOperations	Array	Список доступных операций над задачей: — <code>Operation</code> — операция над задачей
token	String	Токен

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.16.3. Получение информации о выбранных задачах по их идентификатору

Запрос для получения информации о количестве выбранных задач и доступных операциях с ними по их идентификатору.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
PUT <Корневой URL API>/api/scanning/v3/task_selections/{id}
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 200 (OK). Ответ может содержать поля, описанные в таблице ниже.

Таблица 142. Поля ответа на запрос `/api/scanning/v3/task_selections/{id}`

Поле	Тип данных	Описание
<code>selectionId</code>	String	Идентификатор выборки
<code>selectedTasksCount</code>	Integer	Количество выбранных задач
<code>availableOperations</code>	Array	Список доступных операций над задачей: — <code>Operation</code> — операция над задачей
<code>token</code>	String	Токен

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

3.16.4. Отмена выбора задач

Запрос для отмены выбора задач.

Для выполнения запроса требуется аутентификация по протоколу OAuth с токеном доступа типа Bearer.

Метод и URL запроса:

```
DELETE <Корневой URL API>/api/scanning/v3/task_selections/{id}
```

URL запроса содержит path-параметр `id` — идентификатор задачи.

Ответ на запрос

В ответ на успешный запрос сервис возвращает код 204 (No Content).

Возможные коды ошибок и их значения:

- 400 (Bad Request) — синтаксическая ошибка в запросе.

4. Отправка уведомлений через POST-запрос

В MaxPatrol VM предусмотрена возможность отправки уведомлений на внешний сервер через POST-запросы. Вы можете настроить отправку таких уведомлений для следующих случаев:

- изменение состава активов в системе или в выбранной группе;
- получение события, удовлетворяющего выбранному фильтру;
- запуск или остановка задачи сбора данных;
- выход параметров потока событий от источника из выбранной группы за пределы допустимых значений;
- появление уведомления о состоянии системы.

Для приема запросов на сервере должно быть установлено специализированное ПО.

Для отправки уведомлений через POST-запросы в веб-интерфейсе MaxPatrol VM в разделе **Система** на странице **Уведомления** нужно создать задачу на отправку уведомлений. В параметрах задачи нужно включить отправку уведомления через POST-запрос и указать URL внешнего сервера. Вы можете настроить отправку мгновенных уведомлений или уведомлений за период времени (см. Руководство администратора).

После создания задачи в случае выполнения условия отправки уведомления MaxPatrol VM автоматически собирает данные для уведомления и отправляет POST-запрос на внешний сервер. Специализированное ПО на внешнем сервере принимает POST-запрос и по ссылке в запросе получает из MaxPatrol VM данные уведомления.

Для приема POST-запросов и получения данных уведомления к ПО выдвигаются следующие требования:

- поддержка обмена данными по протоколам HTTP или HTTPS;
- прием POST-запросов без аутентификации;
- отправка подтверждения с кодом ответа 2xx при приеме POST-запроса;

Примечание. При отсутствии подтверждения получения запроса от внешнего сервера (с кодом ответа 2xx) POST-запрос будет отправляться повторно до 10 раз с интервалом в минуту.

- получение данных уведомления GET-запросом по ссылке в POST-запросе;
- при получении данных по протоколу HTTPS поддержка проверки подлинности сервера с помощью сертификата SSL.

В этом разделе

[Поля POST-запроса для уведомления \(см. раздел 4.1\)](#)

[Тестирование приема POST-запросов \(см. раздел 4.2\)](#)

4.1. Поля POST-запроса для уведомления

MaxPatrol VM отправляет POST-запрос в формате JSON. В зависимости от типа запрос может содержать следующие поля:

- `message_id` — идентификатор POST-запроса.
- `notification_type` — тип POST-запроса: `test` — тестовый запрос, `event` — для мгновенного уведомления, `schedule` — для уведомления за период времени.
- `notification_source` — условие создания уведомления: `AssetStateMetaTrigger` — изменение состава активов в системе; `EventsMetaTrigger` — получение события, удовлетворяющего выбранному фильтру; `EventsMonitoringControlsMetaTrigger` — выход параметров потока событий от источника из выбранной группы за пределы допустимых значений; `GroupContentMetaTrigger` — изменение состава активов в выбранной группе; `HealthMonitoringIssuesMetaTrigger` — появление уведомления о состоянии системы; `ScannerTaskNotificationsMetaTrigger` — запуск или остановка задачи на сбор данных.
- `notification_uid` — идентификатор уведомления.
- `notification_name` — название задачи MaxPatrol VM на отправку уведомления.
- `uri` — ссылка на данные уведомления.

Внимание! Данные уведомления доступны по ссылке в течение 24 часов с момента создания уведомления.

- `schema_uri` — ссылка на схему данных уведомления в формате JSON.
- `event_time_stamp` — дата и время создания мгновенного уведомления (UTC+0).
- `time_interval` — для уведомления за период содержит поля `event_time_start` и `event_time_end` с датами и временем начала и конца периода (UTC+0).
- `time_stamp` — дата и время создания POST-запроса (UTC+0).

Пример тестового POST-запроса:

```
{
  "notification_type": "test",
  "notification_source": "AssetStateMetaTrigger",
  "schema_uri": "https://vm-server.ru:8733/api/assets_triggers/v1/triggers_data/meta_triggers/AssetStateMetaTrigger/reactions/webhook_notification/schema",
  "time_stamp": "2019-04-15T11:20:49.7031727Z"
}
```

Пример POST-запроса для уведомления за период времени:

```
{
  "message_id": 1194,
  "notification_type": "schedule",
  "notification_source": "AssetStateMetaTrigger",
  "notification_uid": "110a1602-d980-0001-0000-000000000006",
}
```

```

    "notification_name": "1",
    "uri": "https://vm-server.ru:8733/api/assets_triggers/v1/triggers_data/meta_triggers/
AssetStateMetaTrigger/reactions/webhook_notification/
110a718e730000010000000000000261",
    "schema_uri": "https://vm-server.ru:8733/api/assets_triggers/v1/triggers_data/
meta_triggers/AssetStateMetaTrigger/reactions/webhook_notification/schema",
    "event_time_stamp": "2019-04-12T09:50:21.4309566Z",
    "time_interval": {
        "event_time_start": "2019-04-12T09:50:21.4309566Z",
        "event_time_end": "2019-04-12T09:55:21.4309566Z"
    },
    "time_stamp": "2019-04-12T09:55:33.0096243Z"
}

```

4.2. Тестирование приема POST-запросов

В качестве примера ПО для приема POST-запросов и получения данных уведомления приводится сценарий на языке Python. В сценарии реализованы следующие возможности:

- При получении POST-запроса в интерфейс командной строки выводится его текст.
- При получении POST-запроса в MaxPatrol VM отправляется подтверждение с кодом ответа 2xx.
- По ссылке в POST-запросе из MaxPatrol VM скачиваются не более 10 страниц данных уведомления и выводятся в интерфейс командной строки.
- При необходимости по ссылке в POST-запросе из MaxPatrol VM скачивается схема данных уведомления и выводится в интерфейс командной строки.
- При получении данных из MaxPatrol VM по протоколу HTTPS для проверки подлинности сервера может быть предоставлен сертификат SSL.

Для выполнения сценария на сервере должны быть установлены операционные системы Windows 2012 R2 или Debian 9 и интерпретатор языка Python версии 3.6 или 3.7 с библиотеками Flask и Requests.

Примечание. Для отправки уведомлений на сервер в веб-интерфейсе MaxPatrol VM нужно создать задачу на отправку уведомлений через POST-запросы. В параметрах задачи нужно указать URL сервера в виде `http://<IP-адрес или FQDN>/handle`.

► Чтобы настроить прием POST-запросов и получение данных уведомления:

1. Создайте файл с расширением .py и скопируйте в него код сценария.
2. Если требуется, с помощью параметра HOST укажите сетевой интерфейс сервера.
3. Если требуется выводить в интерфейс командной строки схему данных уведомления, для параметра SHOW_SCHEMA укажите значение True.
4. Если требуется, с помощью параметров HTTP_PORT и HTTPS_PORT измените порты для приема запросов по протоколам HTTP и HTTPS.

5. Если требуется использовать протокол HTTPS, для параметра USE_HTTPS укажите значение True.

Внимание! Для проверки подлинности при получении сценарием данных из MaxPatrol VM по протоколу HTTPS нужно выпустить сертификат SSL. Файлы сертификата и ключа к нему нужно поместить в одну папку с файлом сценария. С помощью параметров сценария `certfile` и `keyfile` нужно указать имена файлов сертификата и ключа.

6. Сохраните файл.
7. Откройте интерфейс командной строки и запустите сценарий:


```
python <Имя файла сценария>.py
```

Прием POST-запросов настроен. Данные уведомлений будут выводиться в интерфейс командной строки.

Код сценария для приема POST-запросов

```
"""pip install flask requests"""
import requests
from pathlib import Path
from flask import Flask, request
HOST = '0.0.0.0'
HTTP_PORT = 10080
HTTPS_PORT = 10443
SHOW_SCHEMA = False
USE_HTTPS = False
cert_dir = Path(__file__).parent
certfile = cert_dir / '<Имя файла сертификата>.crt'
keyfile = cert_dir / '<Имя файла ключа>.pem'
app = Flask('ExampleTriggerService')
@app.route('/handle', methods=['POST'])
def handle():
    data = request.get_json()
    print('INCOMING DATA', data)
    if data['notification_type'] == 'test':
        return print('TEST') or 'test'
    if SHOW_SCHEMA:
        print('GET SCHEMA', data['schema_uri'])
        schema_data = requests.get(data['schema_uri'], verify=False).text
        print('SCHEMA DATA', schema_data)
    page_uri = data['uri']
    for page_num in range(10):
        if not page_uri:
            break
        print('GET PAGE', page_num, page_uri)
        page_data = requests.get(page_uri, verify=False).json()
```

```
    print('PAGE DATA', page_data)
    page_uri = page_data['dataSetInfo']['nextDataSetUri']
    return print('OK') or 'ok'
if __name__ == '__main__':
    if USE_HTTPS:
        import ssl
        context = ssl.SSLContext(ssl.PROTOCOL_TLS)
        # context = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2) # для Python версии 3.6
        context.load_cert_chain(str(certfile), str(keyfile))
        port = HTTPS_PORT
    else:
        context = None
        port = HTTP_PORT
    app.run(ssl_context=context, host=HOST, port=port, debug=True)
```

5. О технической поддержке

Базовая техническая поддержка доступна для всех владельцев действующих лицензий на MaxPatrol VM в течение периода предоставления обновлений и включает в себя следующий набор услуг.

Предоставление доступа к обновленным версиям продукта

Обновления продукта выпускаются в порядке и в сроки, определяемые Positive Technologies. Positive Technologies предоставляет обновленные версии продукта в течение оплаченного периода получения обновлений, указанного в бланке лицензии приобретенного продукта Positive Technologies.

Positive Technologies не производит установку обновлений продукта в рамках технической поддержки и не несет ответственности за инциденты, возникшие в связи с некорректной или несвоевременной установкой обновлений продукта.

Восстановление работоспособности продукта

Специалист Positive Technologies проводит диагностику заявленного сбоя и предоставляет рекомендации для восстановления работоспособности продукта. Восстановление работоспособности может заключаться в выдаче рекомендаций по установке продукта заново с потенциальной потерей накопленных данных либо в восстановлении версии продукта из доступной резервной копии (резервное копирование должно быть настроено заблаговременно). Positive Technologies не несет ответственности за потерю данных в случае неверно настроенного резервного копирования.

Примечание. Помощь оказывается при условии, что конечным пользователем продукта соблюдены все аппаратные, программные и иные требования и ограничения, описанные в документации к продукту.

Устранение ошибок и дефектов в работе продукта в рамках выпуска обновленных версий

Если в результате диагностики обнаружен дефект или ошибка в работе продукта, Positive Technologies обязуется включить исправление в ближайшие возможные обновления продукта (с учетом релизного цикла продукта, приоритета дефекта или ошибки, сложности требуемых изменений, а также экономической целесообразности исправления). Сроки выпуска обновлений продукта остаются на усмотрение Positive Technologies.

Рассмотрение предложений по доработке продукта

При обращении с предложением по доработке продукта необходимо подробно описать цель такой доработки. Вы можете поделиться рекомендациями по улучшению продукта и оптимизации его функциональности. Positive Technologies обязуется рассмотреть все предложения, однако не принимает на себя обязательств по реализации каких-либо

доработок. Если Positive Technologies принимает решение о доработке продукта, то способы ее реализации остаются на усмотрение Positive Technologies. Заявки принимаются [на портале технической поддержки](#).

Портал технической поддержки

[На портале технической поддержки](#) вы можете круглосуточно создавать и обновлять заявки и читать новости продуктов.

Для получения доступа к portalу технической поддержки нужно создать учетную запись, используя адрес электронной почты в официальном домене вашей организации. Вы можете указать другие адреса электронной почты в качестве дополнительных. Добавьте в профиль название вашей организации и контактный телефон – так нам будет проще с вами связаться.

Техническая поддержка на портале предоставляется на русском и английском языках.

Условия предоставления технической поддержки

Для получения технической поддержки необходимо оставить заявку [на портале технической поддержки](#) и предоставить следующие данные:

- номер лицензии на использование продукта;
- файлы журналов и наборы диагностических данных, которые требуются для анализа;
- снимки экрана.

Positive Technologies не берет на себя обязательств по оказанию услуг технической поддержки в случае вашего отказа предоставить запрашиваемую информацию или отказа от внедрения предоставленных рекомендаций.

Если заказчик не предоставляет необходимую информацию по прошествии 20 календарных дней с момента ее запроса, специалист технической поддержки имеет право закрыть заявку. Оказание услуг может быть возобновлено по вашей инициативе при условии предоставления запрошенной ранее информации.

Услуги по технической поддержке продукта не включают в себя услуги по переустановке продукта, решение проблем с операционной системой, инфраструктурой заказчика или сторонним программным обеспечением.

Заявки могут направлять уполномоченные сотрудники заказчика, владеющего продуктом на законном основании (включая наличие действующей лицензии). Специалисты заказчика должны иметь достаточно знаний и навыков для сбора и предоставления диагностической информации, необходимой для решения заявленной проблемы.

Услуги по технической поддержке оказываются в отношении поддерживаемых версий продукта. Информация о поддерживаемых версиях содержится в «Политике поддержки версий программного обеспечения» и (или) иных информационных материалах, размещенных на официальном веб-сайте Positive Technologies.

Время реакции и приоритизация заявок

При получении заявки ей присваивается регистрационный номер, специалист службы технической поддержки классифицирует ее (присваивает тип и уровень значимости) и выполняет дальнейшие шаги по ее обработке.

Время реакции рассчитывается с момента получения заявки до первичного ответа специалиста технической поддержки с уведомлением о взятии заявки в работу. Время реакции зависит от уровня значимости заявки. Специалист службы технической поддержки оставляет за собой право переопределить уровень значимости в соответствии с приведенными ниже критериями. Указанные сроки являются целевыми, но возможны отклонения от них по объективным причинам.

Таблица 143. Время реакции на заявку

Уровень значимости заявки	Критерии значимости заявки	Время реакции на заявку
Критический	Аварийные сбои, приводящие к невозможности штатной работы продукта (исключая первоначальную установку) либо оказывающие критически значимое влияние на бизнес заказчика	До 4 часов
Высокий	Сбои, проявляющиеся в любых условиях эксплуатации продукта и оказывающие значительное влияние на бизнес заказчика	До 8 часов
Средний	Сбои, проявляющиеся в специфических условиях эксплуатации продукта либо не оказывающие значительного влияния на бизнес заказчика	До 8 часов
Низкий	Вопросы информационного характера либо сбои, не влияющие на эксплуатацию продукта	До 8 часов

Указанные часы относятся к рабочему времени (рабочие дни с 9:00 до 18:00 UTC+3) специалистов технической поддержки. Под рабочим днем понимается любой день за исключением субботы, воскресенья и дней, являющихся официальными нерабочими днями в Российской Федерации.

Обработка и закрытие заявок

По мере рассмотрения заявки и выполнения необходимых действий специалист технической поддержки сообщает вам:

- о ходе диагностики по заявленной проблеме и ее результатах;
- планах выпуска обновленной версии продукта (если требуется для устранения проблемы).

Если по итогам обработки заявки необходимо внести изменения в продукт, Positive Technologies включает работы по исправлению в ближайшее возможное плановое обновление продукта (с учетом релизного цикла продукта, приоритета дефекта или ошибки, сложности требуемых изменений, а также экономической целесообразности исправления). Сроки выпуска обновлений продукта остаются на усмотрение Positive Technologies.

Специалист закрывает заявку, если:

- предоставлены решение или возможность обойти проблему, не влияющие на критически важную функциональность продукта (по усмотрению Positive Technologies);
- диагностирован дефект продукта, собрана техническая информация о дефекте и условиях его воспроизведения, исправление дефекта запланировано к выходу в рамках планового обновления продукта;
- заявленная проблема вызвана программным обеспечением или оборудованием сторонних производителей, на которые не распространяются обязательства Positive Technologies;
- заявленная проблема классифицирована специалистами Positive Technologies как неподдерживаемая.

Примечание. Если продукт приобретался вместе с аппаратной платформой в виде программно-аппаратного комплекса (ПАК), решение заявок, связанных с ограничением или отсутствием работоспособности аппаратных компонентов, происходит согласно условиям и срокам, указанным в гарантийных обязательствах (гарантийных талонах) на такие аппаратные компоненты.



Positive Technologies — один из лидеров в области результативной кибербезопасности. Компания является ведущим разработчиком продуктов, решений и сервисов, позволяющих выявлять и предотвращать кибератаки до того, как они причинят неприемлемый ущерб бизнесу и целым отраслям экономики. Наши технологии используют около 3000 организаций по всему миру. Positive Technologies — первая и единственная компания из сферы кибербезопасности на Московской бирже (МОЕХ: POSI). Количество акционеров превышает 220 тысяч.