MaxPatrol EDR версия 5.0

Руководство разработчика

Дата редакции документа: 03.10.2023

Содержание

1. Об этом документе

Руководство разработчика содержит справочную информацию и инструкции для специалистов, разрабатывающих модули агентов в MaxPatrol Endpoint Detection and Response (далее также — MaxPatrol EDR). Руководство не содержит инструкций по использованию основных функций продукта.

Комплект документации MaxPatrol EDR включает в себя следующие документы:

— Этот документ.

— Руководство администратора — содержит справочную информацию и инструкции по развертыванию, настройке и администрированию MaxPatrol EDR.

— Начало работы — содержит информацию и инструкции для первоначальной настройки MaxPatrol EDR.

2. О MaxPatrol EDR

MaxPatrol Endpoint Detection and Response — система на базе платформы MaxPatrol 10, предназначенная для защиты конечных устройств от киберугроз. Собирая и анализируя данные из множества систем, MaxPatrol EDR выявляет в IT-инфраструктуре организации сложные целевые атаки и автоматически реагирует на них. MaxPatrol EDR встроен в экосистему продуктов Positive Technologies и позволяет:

— отправлять подозрительные файлы на проверку в PT Sandbox и использовать полученные вердикты одновременно на всех конечных устройствах;

— запускать на конечных устройствах сканирование в режиме аудита и отправлять результаты в MaxPatrol VM;

— использовать для выявления и расследования кибератак данные и экспертизу из других продуктов.

При обнаружении угроз MaxPatrol EDR имеет возможность выполнить следующие автоматические действия:

— удалить файл;

— завершить один или несколько процессов;

— заблокировать сетевой трафик;

— запустить проверку файлов и процессов на основе YARA-правил;

— отправить файл на проверку в PT Sandbox;

— отправить данные о событиях ИБ на syslog-сервер.

Кроме того, администратор или оператор системы может в любой момент времени вручную запустить на конечном устройстве реагирование на угрозу.

3. Архитектура и алгоритм работы MaxPatrol EDR

MaxPatrol EDR состоит из серверной части и агентов, устанавливаемых на конечные устройства. Серверная часть MaxPatrol EDR состоит из двух программных компонентов — управляющего сервера и сервера агентов.

Управляющий сервер — основной компонент системы, который позволяет конфигурировать ее через веб-интерфейс. Сервер агентов — приложение для управления агентами и модулями, а также для взаимодействия с внешними системами (MaxPatrol SIEM, MaxPatrol VM, PT Sandbox, syslog-сервер).

Агент — приложение, которое устанавливается на конечное устройство для обеспечения работы модулей и связи с сервером агентов. Модуль — приложение, которое запускается на конечном устройстве для выполнения основных функций продукта.

Существуют две конфигурации MaxPatrol EDR — односерверная и многосерверная. Многосерверную конфигурацию вы можете использовать для распределения нагрузки при большом количестве конечных устройств в вашей организации. В этой конфигурации на основном сервере устанавливаются все компоненты, а на других только сервер агентов, СУБД PostgreSQL и объектное хранилище MinIO.

Алгоритм работы MaxPatrol EDR:

1. Сервер агентов передает на агенты модули и их конфигурацию.
2. Модули доставки и установки устанавливают и настраивают приложения на конечном устройстве, например Sysmon.
3. Модули сбора собирают данные о системных событиях, кэшируют их в памяти агента, передают в модули обнаружения и при необходимости на сервер агентов и в MaxPatrol SIEM.
4. Модули обнаружения анализируют файлы, процессы, собранные события, обнаруживают подозрительную активность на конечном устройстве — и регистрируют события ИБ.
5. Модули реагирования пресекают подозрительную и вредоносную активность, выполняя действия в соответствии с политикой или по команде пользователя.
6. Модули интеграции обеспечивают интеграцию с внешними системами.
7. Данные о событиях ИБ кэшируются в памяти агента, сервера агентов и пересылаются в базу данных MaxPatrol SIEM.
8. Агент передает метрики и данные трассировки (см. раздел 5) на сервер агентов.
9. Управляющий сервер получает обновления продукта и пакетов экспертизы с сервера обновлений.

4. Разработка модулей в MaxPatrol EDR

Далее приведена основная информация о модулях агента, а также даны инструкции по работе с ними.

В этом разделе

4.1. Структура модулей

Модуль агента — это приложение, которое запускается на агенте для выполнения основных функций продукта.

В MaxPatrol EDR есть пять типов модулей:

- **Модули доставки и установки.** Устанавливают и настраивают приложения и управляют конфигурацией ОС на конечном устройстве.
- **Модули сбора.** Собирают данные о событиях на конечном устройстве и передают их в модули обнаружения и в SIEM-системы.
- **Модули обнаружения.** Анализируют собранные события, обнаруживают подозрительную и вредоносную активность на конечном устройстве — и регистрируют события ИБ.
- **Модули реагирования.** Пресекают подозрительную и вредоносную активность на конечном устройстве, выполняя действия в соответствии с конфигурацией модулей обнаружения.
- **Модули интеграции.** Обеспечивают интеграцию с внешними системами.

Некоторые модули по своим функциям могут относиться к нескольким типам.

Вы можете разрабатывать свои модули в MaxPatrol EDR. При создании нового модуля вам доступны шаблоны для каждого типа модуля, которые определяют набор функций и методов в исходных файлах.

Модуль в MaxPatrol EDR состоит из четырех частей:

- **Базовая.** Содержит в файлах формата JSON (см. приложение А) общую информацию о модуле и конфигурацию основных объектов — переменных, событий, действий и параметров.
- **Агентская.** Содержит исполняемые файлы, которые выполняются на конечном устройстве, а также конфигурационные и дополнительные файлы.
- **Серверная.** Содержит исполняемые файлы, которые выполняются на сервере MaxPatrol EDR, а также конфигурационные и дополнительные файлы. Серверная часть модуля отвечает за обмен данными между веб-частью модуля и агентом.
- **Веб-часть.** Содержит исполняемые и конфигурационные файлы, а также файлы разметки, отвечающие за возможности ручного реагирования на угрозы из веб-интерфейса MaxPatrol EDR.

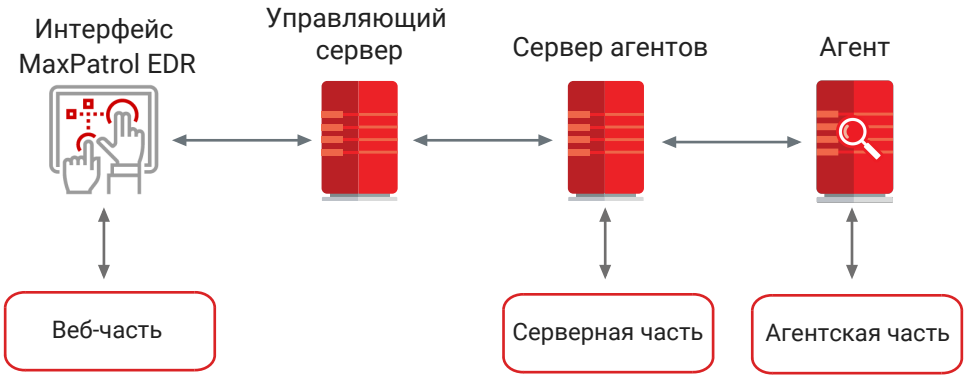

Рисунок 1. Структура модулей

4.2. Принципы разработки модулей

В этом разделе приведены основные принципы разработки модулей в MaxPatrol EDR и примеры исходного кода серверной и агентской части модуля.

В этом разделе

Взаимодействие частей Lua-кода в модулях (см. раздел 4.2.1)

Получение и отправка сообщений (см. раздел 4.2.2)

Пример маршрутизации сообщений в серверной части модуля для связки агентской и веб-части (см. раздел 4.2.3)

Пример агентской части модуля реагирования (см. раздел 4.2.4)

Пример агентской части модуля обнаружения (см. раздел 4.2.5)

4.2.1. Взаимодействие частей Lua-кода в модулях

Серверная и агентская части модуля представляют собой скрипты на языке Lua с возможностью подключения скомпилированных библиотек на языках C и C++. Скрипты исполняются последовательно в среде с кооперативной многозадачностью. Исполнение серверной и агентской частей модуля всегда начинается с файла `main.lua`. Исходный код серверной и агентской частей выполняет две основные группы функций:

— обработка сообщений типа `action` от сервера и других модулей, выполнение действий и регистрация событий по их результатам;

— опрос сенсоров, вызов системных библиотек, получение комплексных ответов от динамических библиотек и отправка событий на сервер.

В соответствии с принципами кооперативной многозадачности модули должны получать управление и выполнять полезные действия. Пример исходного кода модуля с минимальным набором функций:

```
-- cmodule/main.lua
__api.await(-1)
```

Этот код при инициализации модуля сразу передает управление другим модулям на неограниченный срок (вызов функции `__api.await` с параметром `-1`).

Клиентская и серверная части модуля взаимодействуют с другими модулями и сервером MaxPatrol EDR с помощью двух компонентов:

— Система обратных вызовов (callbacks) позволяет получать структурированную информацию и реагировать на нее в коде модуля.

— Система коммуникации (networking) позволяет отправлять данные и команды между частями модуля и осуществлять взаимодействие с другими модулями.

Обратные вызовы (callbacks) регистрируются во время инициализации модуля с помощью функции `__api.set_cbs()`:

```
-- начало блока инициализации модуля
__api.set_cbs(...)
-- конец блока инициализации

-- передача управления вызывающему коду
__api.await(-1)

-- начало финального блока модуля
-- ...
-- конец финального блока
```

Внимание! Получение сообщений через систему обратных вызовов работает в рамках среды с кооперативной многозадачностью и по тем же принципам, что и остальные части модуля. Одним из вариантов возврата управления коду модуля является передача обратного вызова.

Если модуль должен выполнять какие-либо действия независимо от внешних сообщений, то для этого ему нужно периодически возобновлять работу и передавать управление вызывающему коду:

```
-- начало блока инициализации модуля
__api.set_cbs(...)
-- конец блока инициализации

while not __api.is_close() do
__api.await(100) -- минимальное количество миллисекунд до возврата управления модулю
do_own_work()
end
-- начало финального блока модуля
-- ...
-- конец финального блока
```

В этом примере функция `__api.is_close()` позволяет установить момент, в который управляющий код остановил работу модуля, а функция `__api.await()` передает управление вызывающему коду на заданное время.

4.2.2. Получение и отправка сообщений

Для получения модулем сообщений с обратной связью (callbacks) нужно зарегистрировать соответствующие функции. Для этого используется функция `__api.add_cbs` (add callbacks):

```
__api.add_cbs({
  data = function(src, data)
    __log.debugf("received data: %s", data)
    return true
  end,
  file = function(src, path, name)
    __log.debugf("received file '%s' stored in: %s", name, path)
    return true
  end,
  action = function(src, data, name)
    __log.debugf("received action '%s' with payload: %s", name, data)
    return true
  end,
  control = function(name, data)
    __log.debugf("received control msg '%s' with payload: %s", name, data)
    return true
  end,
})
```

После регистрации такого набора функций модуль может обрабатывать сообщения четырех типов:

— `action` — вызывает выполнение определенного действия;

— `data` — вызывает пересылку произвольного набора данных;

— `file` — вызывает отправку файлов (в том числе бинарных), которые автоматически сохраняются в файловой системе;

— `control` — вызывает обработку одного из управляющих событий: `quit`, `agent_connected`, `agent_disconnected`, `update_config`.

Пример отправки сообщения из модуля на сервер или другому модулю:

```
__api.send_data_to(dst, data)
__api.send_file_from_fs_to(dst, path, name)
__api.send_action_to(dst, data, name)
```

Пример регистрации события:

```
require("engine")
event_engine:push_event(name, data)
```

См. также

Пример маршрутизации сообщений в серверной части модуля для связки агентской и веб-части (см. раздел 4.2.3)

4.2.3. Пример маршрутизации сообщений в серверной части модуля для связки агентской и веб-части

В серверной части модуля обычно задается механизм взаимодействия агентской и веб-части модуля. В этом случае скрипт серверной части модуля выполняет маршрутизацию запросов действий от веб-части модуля к агентской части и возвращает результат действия обратно в веб-часть. Сначала выполняется регистрация обработчиков сообщений с последующей передачей управления вызывающему коду:

```
-- smodule/main.lua

__api.add_cbs({
  action = function(src, data, name)
    __log.infof("receive action '%s' from '%s' with data %s", name, src, data)
    return true
  end,
  data = function(src, data)
    __log.infof("receive data from '%s' with data %s", src, data)
    return true
  end,
  control = function(name, data)
    __log.debugf("receive control msg '%s' with data %s", name, data)
```

```
    return true
  end,
})

__log.infof("module '%s' was started", __config.ctx.name)
__api.await(-1)
__log.infof("module '%s' was stopped", __config.ctx.name)

return "success"
```

Пример исходного кода для передачи действий от веб-части модуля:

```
local action_data = cjson.decode(data)
action_data.retaddr = src

local id = get_agent_id_by_src(src, "any")
local dst = get_agent_src_by_id(id, "VXAgent")
if dst ~= "" then
  __api.send_action_to(dst, cjson.encode(action_data), name)
else
  __api.send_data_to(src, cjson.encode({status
= "error", error = "connection_error"}))
end
```

В этом примере сначала десериализуется объект данных:

```
action_data.retaddr = src
```

Далее заполняется адрес, на который будет отправлен ответ:

```
action_data.retaddr = src
```

Затем выполняется поиск идентификатора агентской части модуля, которой необходимо отправить пакет:

```
local id = get_agent_id_by_src(src, "any")
local dst = get_agent_src_by_id(id, "VXAgent")
```

Если получатель не найден, обратно отправляется ответ с ошибкой:

```
  __api.send_data_to(src, cjson.encode({status
= "error", error = "connection_error"}))
```

Если получатель найден, выполняется сериализация и отправка данных этому получателю:

```
  __api.send_action_to(dst, cjson.encode(action_data), name)
```

Аналогичный пример исходного кода передачи ответа от агентской части модуля:

```
local payload = cjson.decode(data)
local retaddr = payload.retaddr

local id = get_agent_id_by_src(src, "VXAgent")
if type(retaddr) == "string" and retaddr ~= "" and id ~= "" then
  payload.retaddr = nil
```

```
    __api.send_data_to(payload.retaddr, cjson.encode(payload))
end
```

4.2.4. Пример агентской части модуля реагирования

В этом примере модуль реагирования в ответ на сообщение типа `action` выполняет на конечном устройстве определенное действие:

```
local cjson   = require("cjson.safe")

__api.add_cbs({
    action = function(src, data, name)
        local action_data = cjson.decode(data) or {}
        local result = {
            retaddr=action_data.retaddr, agent_id=__aid, name=name,
            type="exec_test_response", status="", error=""
        }

        if name == "update_test_file" then
            filewrite = io.open("tempfile", "w")
            filewrite:write(data)
            filewrite:close()
            result.status = "success"
        else
            result.status = "error"
            result.error = string.format("action %s not registered", name)
        end

        __api.send_data_to(src, cjson.encode(result))
        return true
    end,
})

__api.await(-1)

return "success
```

Для десериализации полученного сообщения типа `action` и формирования шаблона ответа регистрируется функция:

```
__api.add_cbs({
    action = function(src, data, name)
        local action_data = cjson.decode(data) or {}
        local result = {
            retaddr=action_data.retaddr, agent_id=__aid, name=name,
            type="exec_test_response", status="", error=""
        }
```

После этого выполняется проверка действия. Если действие поддерживается модулем, то модуль выполняет его и дополняет ответ:

```
if name == "update_test_file" then
    filewrite = io.open("tempfile", "w")
    filewrite:write(data)
    filewrite:close()
    result.status = "success"
else
    result.status = "error"
    result.error = string.format("action %s not registered", name)
```

После выполнения действия отправляется ответ источнику изначального сообщения:

```
__api.send_data_to(src, cjson.encode(result))
```

4.2.5. Пример агентской части модуля обнаружения

В этом примере модуль обнаружения регистрирует события об изменении определенного файла в файловой системе:

```
require("engine")

local function read_file(file)
  local file = open(file, "rb")
  if not file then return nil end
  local content = file:read "*a"
  file:close()
  return content
end

local file_name="tempfile"
local prev_file_contents = read_file(file_name)

while not __api.is_close() do
  __api.await(10000)
  do_own_work()
  local cur_file_contents = read_file(file_name)
  if prev_file_contents ~= cur_file_contents then
    event_engine:push_event({ name="test_file_changed", data={}, actions={} })
    prev_file_contents = cur_file_contents
  end
end


return "success"
```

Во время инициализации модуля формируется начальное состояние и запоминается содержимое искомого файла:

```
local file_name="tempfile"
local prev_file_contents = read_file(file_name)
```

Затем модуля циклично раз в 10 секунд проверяет этот же файл и сравнивает его содержимое с сохраненной версией. Если модуль обнаруживает несоответствие, то регистрируется событие:

```
event_engine:push_event({ name="test_file_changed", data={}, actions={} })
```

Внимание! Этот пример не подходит для разработки реальных модулей. Модуль хранит в памяти файл, а также выполняет постоянный опрос файловой системы и чтение содержимого файла. Это может привести к чрезмерному использованию ресурсов на конечном устройстве.

4.3. Создание модуля

- Чтобы создать модуль:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите кнопку **Создать модуль**.

   Откроется окно **Новый модуль**.

3. В поле **Идентификатор** введите идентификатор модуля.
4. В раскрывающемся списке **Шаблон** выберите шаблон, на основе которого будет создан модуль.
5. В поле **Версия** введите номер первой версии модуля.
6. Выберите типы операционных систем, на которых будет работать модуль.

Модуль создан.

4.4. Разработка модуля

Для разработки серверной и агентской части модуля в MaxPatrol EDR используются язык Lua и компилятор LuaJIT. Более требовательные к ресурсам части модуля могут быть вынесены в библиотеки, написанные на языках C и C++. Веб-часть модуля основывается на фреймворке Vue.js, для ее разработки используются языки JavaScript, HTML и CSS.

Далее приведены основные инструкции по разработке модуля.

В этом разделе

4.4.1. Добавление переменной

Модули используют в своей работе переменные, с помощью которых они обмениваются данными. Например, модуль обнаружения с помощью переменной `filepath` в событии передает в модуль реагирования путь к опасному файлу, который нужно удалить. Для передачи модулем данных или выполнения им действий вам нужно добавить переменные.

> **Примечание.** В MaxPatrol EDR зафиксирован ряд переменных (см. приложение Б), которые используются модулями.

- Чтобы добавить переменную:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

     Откроется страница **Модули EDR**.

  2. Нажмите на название модуля.

     Откроется карточка модуля.

  3. Выберите вкладку **Переменные**.

  4. Нажмите кнопку **Добавить**.

  5. Если требуется, чтобы все события и действия модуля использовали эту переменную, установите флажок **Да**.

  6. В поле **Идентификатор** введите идентификатор переменной.

  7. В раскрывающемся списке **Тип данных** выберите тип данных переменной.

8. Если требуется, в поле **Дополнительные ключи** введите дополнительную конфигурацию переменной.

   Если в дополнительной конфигурации нужны параметры или сообщения, требующие локализации на разные языки, вводите соответствующие ключи в формате Domain.Feature.ControlType.Function. В этом случае вы сможете локализовать их на вкладке **Локализация**.

   Например:

   ```
   "ErrMsg": "BrowserModule.BlockByIpConfig.ErrorText.IncorrectFormat"
   ```

   Примечание. В MaxPatrol EDR интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

9. Добавьте переменную хотя бы в одно событие (см. раздел 4.4.2) или действие (см. раздел 4.4.3).

10. Нажмите кнопку **Сохранить**.

Переменная добавлена.

4.4.2. Добавление событий

Модуль может регистрировать события трех типов:

- **Атомарные.** Простые события.
- **Агрегированные.** Слияние нескольких одинаковых событий любого типа в одно результирующее.
- **Корреляционные.** Слияние цепочки событий любых типов в одно результирующее.

Далее приведены инструкции по добавлению событий всех типов.

В этом разделе

4.4.2.1. Добавление простого события

► Чтобы добавить простое событие:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. Выберите вкладку **События**.
4. В панели **Схема** нажмите кнопку **Добавить**.
5. В поле **Идентификатор** введите идентификатор события.
6. Если требуется, в раскрывающемся списке **Переменные** выберите переменные, которые будут передавать событие.
7. Если требуется, в поле **Дополнительные ключи событий** введите дополнительную конфигурацию события.

   Если в дополнительной конфигурации нужны параметры или сообщения, требующие локализации на разные языки, вводите соответствующие ключи в формате Domain.Feature.ControlType.Function. В этом случае вы сможете локализовать их на вкладке **Локализация**.

   Например:

   ```
   "ErrMsg": "BrowserModule.BlockByIpConfig.ErrorText.IncorrectFormat"
   ```

   Примечание. В MaxPatrol EDR интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.
8. Если требуется, в панели **По умолчанию** выберите действия, которые будут выполняться по умолчанию при регистрации события.
9. Нажмите кнопку **Сохранить**.

Событие добавлено.

4.4.2.2. Добавление агрегированного события

- Чтобы добавить агрегированное событие:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.
2. Нажмите на название модуля.

   Откроется карточка модуля.
3. Выберите вкладку **События**.
4. В панели **Схема** нажмите кнопку **Добавить**.
5. В поле **Идентификатор** введите идентификатор события.
6. В раскрывающемся списке **Тип события** выберите значение **aggregation**.
7. В панели **По умолчанию** настройте алгоритм агрегации событий.

   Агрегация начинается при регистрации нескольких одинаковых событий (блок параметров **Последовательность**), в которых совпадают значения указанных переменных (блок параметров **Группировка по переменным**). Агрегированное

событие будет зарегистрировано, если количество подходящих событий будет больше, чем значение параметра **min_count**. Агрегация завершается либо после регистрации максимального количества подходящих событий (параметр **Максимальное количество событий**), либо после истечения указанного времени (параметр **Время для сбора последовательности, сек**).

8. Если требуется, в панели **По умолчанию** выберите действия, которые будут выполняться по умолчанию при регистрации события.
9. Нажмите кнопку **Сохранить**.

Событие добавлено.

4.4.2.3. Добавление корреляционного события

- Чтобы добавить корреляционное событие:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.
2. Нажмите на название модуля.

   Откроется карточка модуля.
3. Выберите вкладку **События**.
4. В панели **Схема** нажмите кнопку **Добавить**.
5. В поле **Идентификатор** введите идентификатор события.
6. В раскрывающемся списке **Тип события** выберите значение **correlation**.
7. В панели **По умолчанию** настройте алгоритм корреляции событий.

   Для регистрации корреляционного события должна произойти цепочка событий (блок параметров **Последовательность**) в течение указанного времени (параметр **Время для сбора последовательности, сек**). В корреляции будут учитываться только те события, у которых будут совпадать значения указанных переменных (блок параметров **Группировка по переменным**).
8. Если требуется, в панели **По умолчанию** выберите действия, которые будут выполняться по умолчанию при регистрации события.
9. Нажмите кнопку **Сохранить**.

Событие добавлено.

4.4.3. Добавление действия

Если модуль будет выполнять действия, вам нужно добавить их.

► Чтобы добавить действие:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. Выберите вкладку **Действия**.

4. В панели **Схема** нажмите кнопку **Добавить**.

5. В поле **Идентификатор** введите идентификатор действия.

6. Если требуется, в раскрывающемся списке **Переменные** выберите переменные, которые будет использовать действие.

7. В поле **Приоритет** укажите приоритет действия в условных единицах от 1 до 100.

   Приоритет определяет порядок выполнения действий, если их назначено несколько на одно событие. Если у двух действий одинаковый приоритет, то они будут выполняться в случайном порядке.

8. Если требуется, в поле **Дополнительные ключи действий** введите дополнительную конфигурацию действия.

   Если в дополнительной конфигурации нужны параметры или сообщения, требующие локализации на разные языки, вводите соответствующие ключи в формате Domain.Feature.ControlType.Function. В этом случае вы сможете локализовать их на вкладке **Локализация**.

   Например:

   ```
   "ErrMsg": "BrowserModule.BlockByIpConfig.ErrorText.IncorrectFormat"
   ```

   Примечание. В MaxPatrol EDR интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

9. Нажмите кнопку **Сохранить**.

Действие добавлено.

4.4.4. Добавление параметров модуля

Вы можете добавлять параметры в конфигурацию модуля. Таким параметром может быть, например, адрес сервера, на который модуль будет отправлять данные. Значения параметров модуля задаются отдельно в каждой политике при настройке модуля.

Параметры могут быть обычными и защищенными. Вы можете использовать защищенные параметры для данных, компрометация которых может привести к ущербу для организации. Значения таких параметров передаются на агенты в зашифрованном виде. Просматривать и изменять защищенные параметры могут только пользователи с соответствующими правами.

В этом разделе

4.4.4.1. Добавление обычного параметра

- Чтобы добавить обычный параметр:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. Выберите вкладку **Конфигурация**.
4. Нажмите кнопку **Добавить**.
5. Если требуется, чтобы параметр был обязательным, установите флажок **Обязательное**.
6. В поле **Идентификатор** введите идентификатор параметра.
7. В раскрывающемся списке **Тип данных** выберите тип данных параметра.
8. Если требуется, в поле **Дополнительные ключи** введите дополнительную конфигурацию параметра.

   Если в дополнительной конфигурации нужны параметры или сообщения, требующие локализации на разные языки, вводите соответствующие ключи в формате Domain.Feature.ControlType.Function. В этом случае вы сможете локализовать их на вкладке **Локализация**.

   Например:

   ```
   "ErrMsg": "BrowserModule.BlockByIpConfig.ErrorText.IncorrectFormat"
   ```

   Примечание. В MaxPatrol EDR интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

9. Если требуется, в панели **По умолчанию** задайте значение параметра по умолчанию.
10. Нажмите кнопку **Сохранить**.

Параметр добавлен.

4.4.4.2. Добавление защищенного параметра

- Чтобы добавить защищенный параметр:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. Выберите вкладку **Защищенная конфигурация**.
4. Нажмите кнопку **Добавить**.
5. Если требуется, чтобы параметр был обязательным, установите флажок **Обязательное**.
6. Если значение параметра не нужно передавать на агенты, установите флажок **Использовать только на сервере**.
7. В поле **Идентификатор** введите идентификатор параметра.
8. В раскрывающемся списке **Тип данных** выберите тип данных параметра.
9. Если требуется, в поле **Дополнительные ключи** введите дополнительную конфигурацию параметра.

   Если в дополнительной конфигурации нужны параметры или сообщения, требующие локализации на разные языки, вводите соответствующие ключи в формате Domain.Feature.ControlType.Function. В этом случае вы сможете локализовать их на вкладке **Локализация**.

   Например:

   ```
   "ErrMsg": "BrowserModule.BlockByIpConfig.ErrorText.IncorrectFormat"
   ```

   Примечание. В MaxPatrol EDR интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

10. Если требуется, в панели **По умолчанию** задайте значение параметра по умолчанию.
11. Нажмите кнопку **Сохранить**.

Параметр добавлен.

4.4.5. Добавление параметра события или действия

Вы можете добавлять параметры для событий и действий. Параметры событий и действий могут быть полезны, если в политике нужно учитывать условия, при которых они регистрируются или выполняются. Например, для события «Скачан подозрительный файл» можно с помощью добавленного параметра указать определенные приложения, для которых будет выполняться проверка. Значения параметров событий и действий задаются отдельно в каждой политике при настройке модуля.

Далее приведена инструкцию по добавлению параметра в событие. Вы можете добавить параметр в действие таким же способом.

- Чтобы добавить параметр:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. Выберите вкладку **События**.

4. Раскройте конфигурацию события, нажав .

5. В разделе **Ключи конфигурации событий** нажмите кнопку **Добавить**.

6. Если требуется, чтобы параметр был обязательным, установите флажок **Обязательное**.

7. В поле **Идентификатор** введите идентификатор параметра.

8. В раскрывающемся списке **Тип данных** выберите тип данных параметра.

9. Если требуется, в поле **Дополнительные ключи** введите дополнительную конфигурацию параметра.

   Если в дополнительной конфигурации нужны параметры или сообщения, требующие локализации на разные языки, вводите соответствующие ключи в формате Domain.Feature.ControlType.Function. В этом случае вы сможете локализовать их на вкладке **Локализация**.

   Например:

   ```
   "ErrMsg": "BrowserModule.BlockByIpConfig.ErrorText.IncorrectFormat"
   ```

   Примечание. В MaxPatrol EDR интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

10. Если требуется, в панели **По умолчанию** задайте значение параметра по умолчанию.

11. Нажмите кнопку **Сохранить**.

Параметр добавлен.

4.4.6. Объявление зависимостей

Модуль может иметь зависимость от версии агента. Это означает, что он может быть установлен только на те агенты, версии которых не ниже указанной. Также у модуля могут быть зависимости от других модулей, если он будет получать или отправлять им данные. В этих случаях вам нужно объявить зависимости модуля.

- Чтобы объявить зависимости:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

     Откроется страница **Модули EDR**.

  2. Нажмите на название модуля.

     Откроется карточка модуля.

3. Выберите вкладку **Зависимости**.
4. Если корректная работа модуля зависит от версии агента, в раскрывающемся списке **Версия агента** выберите версию агента, начиная с которой модуль будет поддерживаться.
5. Если модуль будет получать данные от другого модуля, в блоке параметров **Получение данных** добавьте этот модуль.

   Примечание. При необходимости вы можете выбрать версию добавленного модуля, начиная с которой зависимость будет актуальна. Вы также можете добавить модуль, которого пока нет в репозитории, введя его название.
6. Если модуль будет отправлять данные в другой модуль, в блоке параметров **Отправка данных** добавьте этот модуль.
7. Нажмите кнопку **Сохранить**.

Зависимости объявлены.

4.4.7. Работа с файлами

В MaxPatrol EDR вы можете работать с файлами модуля, в том числе используя встроенный редактор кода. Файлы хранятся на сервере MaxPatrol EDR в каталогах `<Идентификатор модуля>/<Версия>/<Структурная часть>`.

В карточке модуля на вкладке **Файлы** интерфейс разбит на три раздела, соответствующих структурным частям модуля — агентской, серверной и веб-части.

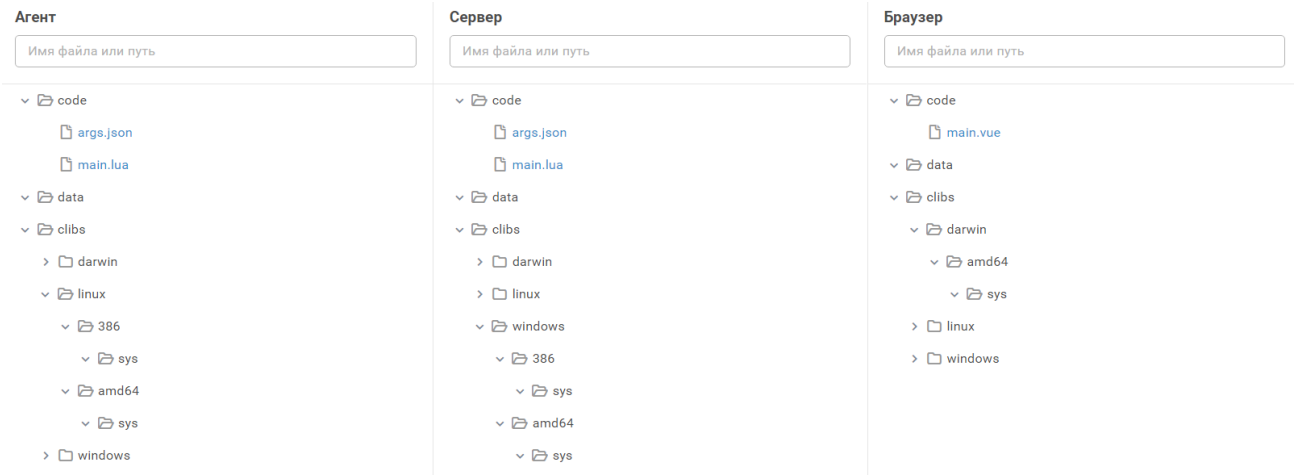

Рисунок 2. Вкладка **Файлы**

Агентская и серверная части состоят из трех основных каталогов с файлами:

— **code.** Каталог для хранения платформонезависимого Lua-кода.

— **data.** Каталог для хранения произвольных данных модуля.

— **clibs.** Каталог со строгой иерархией вложенных каталогов для хранения скомпилированных динамических библиотек.

При создании модуля в каталоги `code` автоматически добавляются обязательные файлы `main.lua`, `args.json` и `main.vue`.

Далее приведены инструкции по работе с файлами.

В этом разделе

Создание файла (см. раздел 4.4.7.1)

Редактирование файла (см. раздел 4.4.7.2)

Загрузка файла на сервер (см. раздел 4.4.7.3)

Скачивание файла (см. раздел 4.4.7.4)

Перемещение файла (см. раздел 4.4.7.5)

Удаление файла (см. раздел 4.4.7.6)

4.4.7.1. Создание файла

► Чтобы создать файл:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. Выберите вкладку **Файлы**.

4. Напротив каталога, в котором вы хотите создать файл, нажмите ⋮ и в раскрывшемся меню выберите пункт **Создать**.

   Откроется окно **Создание файла**.

5. Введите путь и имя файла.

6. Нажмите кнопку **Создать**.

Файл создан.

4.4.7.2. Редактирование файла

- Чтобы отредактировать файл:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

     Откроется страница **Модули EDR**.

  2. Нажмите на название модуля.

     Откроется карточка модуля.

  3. Выберите вкладку **Файлы**.

  4. Нажмите на имя файла.

     Откроется редактор кода.

  5. Внесите изменения в файл.

  6. Нажмите кнопку **Сохранить**.

  Изменения сохранены.

Кроме того, вы можете сбросить изменения и вернуться к версии файла, которая сохранена на сервере, нажав .

4.4.7.3. Загрузка файла на сервер

- Чтобы загрузить файл на сервер:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

     Откроется страница **Модули EDR**.

  2. Нажмите на название модуля.

     Откроется карточка модуля.

  3. Выберите вкладку **Файлы**.

  4. Напротив каталога, в который вы хотите загрузить файл, нажмите и в раскрывшемся меню выберите пункт **Загрузить**.

  5. Выберите файл.

     Откроется окно **Загрузка файла**.

  6. Введите путь к каталогу, в который вы хотите загрузить файл.

  7. Нажмите кнопку **Загрузить**.

  Файл загружен на сервер.

4.4.7.4. Скачивание файла

- Чтобы скачать файл:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

     Откроется страница **Модули EDR**.

  2. Нажмите на название модуля.

     Откроется карточка модуля.

  3. Выберите вкладку **Файлы**.

  4. Напротив файла, который вы хотите скачать, нажмите ⋮ и в раскрывшемся меню выберите пункт **Скачать**.

  Файл сохранен на вашем компьютере.

4.4.7.5. Перемещение файла

Вы можете перемещать файлы в другие каталоги в рамках основного каталога. Переместить файл в другой основной каталог или в другую структурную часть модуля вы можете с помощью операций скачивания (см. раздел 4.4.7.4) и загрузки файла на сервер (см. раздел 4.4.7.3).

- Чтобы переместить файл в другой каталог:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

     Откроется страница **Модули EDR**.

  2. Нажмите на название модуля.

     Откроется карточка модуля.

  3. Выберите вкладку **Файлы**.

  4. Напротив файла, который вы хотите переместить, нажмите ⋮ и в раскрывшемся меню выберите пункт **Переместить**.

     Откроется окно **Перемещение объекта**.

  5. Введите новый путь к файлу.

  6. Нажмите кнопку **Переместить**.

  Файл перемещен.

4.4.7.6. Удаление файла

Вы можете удалить созданные и загруженные файлы. Удалить файлы `main.lua`, `args.json` и `main.vue` невозможно.

► Чтобы удалить файл:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. Выберите вкладку **Файлы**.

4. Напротив файла, который вы хотите удалить, нажмите ⋮ и в раскрывшемся меню выберите пункт **Удалить**.

   Откроется окно **Удаление объекта**.

5. Нажмите кнопку **Удалить**.

Файл удален.

4.4.8. Локализация модуля

Вы можете локализовать на русский и английский языки название и описание модуля, все события, действия, переменные, метки и параметры конфигурации.

► Чтобы локализовать модуль:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. Выберите вкладку **Локализация**.

4. Раскройте списки параметров необходимых объектов и локализуйте их.

5. Нажмите кнопку **Сохранить**.

Модуль локализован.

4.4.9. Добавление истории изменений версии модуля

До релиза версии модуля вы можете заполнить журнал ее изменений.

► Чтобы добавить историю изменений:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

Откроется карточка модуля.

3. Выберите вкладку **История изменений**.
4. Раскройте список с текущей версией модуля и добавьте историю ее изменений.
5. Нажмите кнопку **Сохранить**.

История изменений добавлена.

4.4.10. Просмотр пользовательского интерфейса модуля

Вы можете просмотреть, как интерфейс модуля будет отображаться для операторов и администраторов MaxPatrol EDR.

- Чтобы просмотреть пользовательский интерфейс модуля:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

     Откроется страница **Модули EDR**.
  2. Нажмите на название модуля.

     Откроется карточка модуля.
  3. Нажмите кнопку **Режим пользователя**.

  Отобразился пользовательский интерфейс модуля.

- Чтобы вернуться к разработке модуля,

  повторно нажмите кнопку **Режим пользователя**.

4.4.11. Проверка работы модуля

- Чтобы проверить работу модуля:

  1. Установите агент.
  2. Авторизуйте агент.
  3. Создайте политику.
  4. Добавьте разрабатываемый модуль в политику.
  5. Назначьте политику на группу агентов, в которую входит ваш агент.
  6. Если для работы разрабатываемого модуля необходимы другие модули, назначьте политики с ними на группу агентов, в которую входит ваш агент.
  7. Просмотрите события модуля.

  Примечание. Подробные инструкции см. в документе Руководство администратора.

4.5. Выпуск релиза версии модуля

По умолчанию созданная версия модуля имеет статус «черновик». Вы можете использовать черновик в политиках для отладки работы модуля на агентах. После окончания разработки вы можете выпустить релиз и начать разработку новой версии модуля (см. раздел 4.6). Если черновик использовался в политиках, то модуль будет автоматически обновлен в них после релиза.

- Чтобы выпустить релиз версии модуля:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

     Откроется страница **Модули EDR**.

  2. Нажмите на название модуля.

     Откроется карточка модуля.

  3. Нажмите .

     Откроется окно **Релиз модуля <Название модуля>**.

  4. В блоках параметров **Русский язык** и **Английский язык** заполните журнал изменений модуля на двух языках.

  5. Нажмите кнопку **Выпустить релиз**.

  Релиз модуля выпущен.

4.6. Создание новой версии модуля

После релиза версии модуля (см. раздел 4.5) вы не можете ее редактировать. Для дальнейшей разработки модуля вам нужно создать его новую версию.

- Чтобы создать новую версию модуля:

  1. В главном меню в разделе **EDR** выберите пункт **Модули**.

     Откроется страница **Модули EDR**.

  2. Нажмите на название модуля.

     Откроется карточка модуля.

  3. Нажмите .

     Откроется окно **Новый черновик модуля <Название модуля>**.

  4. В поле **Версия** введите номер новой версии модуля.

5. В блоках параметров **Русский язык** и **Английский язык** заполните журнал изменений модуля на двух языках.
6. Нажмите кнопку **Создать черновик**.

Новая версия модуля создана.

См. также

Выпуск релиза версии модуля (см. раздел 4.5)

4.7. Синхронизация версии модуля

После внесения изменений в версию модуля, которая уже установлена на агентах, вам нужно синхронизировать ее со всеми политиками.

► Чтобы синхронизировать версию модуля:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.
2. Нажмите на название модуля.

   Откроется карточка модуля.
3. Нажмите  .

Версия модуля синхронизирована.

4.8. Экспорт модуля

Вы можете экспортировать с сервера одну или сразу все версии модуля. Это может быть полезно, если вы хотите сохранить резервную копию модуля на вашем компьютере, передать файлы модуля другим разработчикам или вести разработку модуля в привычной для вас среде.

► Чтобы экспортировать модуль:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.
2. Нажмите на название модуля.

   Откроется карточка модуля.
3. Если вы хотите экспортировать только определенную версию модуля, выберите ее в раскрывающемся списке **<Номер версии>**.
4. Выполните одно из следующих действий:
   - Если у модуля есть только одна версия, нажмите кнопку **Экспортировать**.

- Если у модуля есть несколько версий и вы хотите экспортировать только открытую, нажмите кнопку **Экспортировать** и в раскрывшемся меню выберите пункт **Только версию <Номер версии>**.
- Если у модуля есть несколько версий и вы хотите экспортировать их все, нажмите кнопку **Экспортировать** и в раскрывшемся меню выберите пункт **Все версии**.

Архив с файлами модуля сохранен на вашем компьютере.

4.9. Импорт модуля

Вы можете импортировать модуль на сервер MaxPatrol EDR из ZIP-архива. Архив может содержать несколько разных модулей и несколько версий каждого модуля. За одну операцию вы можете загрузить только один модуль, при этом возможен импорт сразу всех версий этого модуля. Если загружаемая версия модуля уже есть на сервере, то импорт будет возможен только при условии перезаписи ее файлов и конфигурации. Размер архива не должен превышать 100 МБ.

► Чтобы импортировать модуль:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.
2. Нажмите кнопку **Импортировать**.
3. Выберите архив с файлами модуля.

   Откроется окно **Импорт модуля**.
4. В раскрывающемся списке **Название модуля** выберите модуль, который вы хотите импортировать на сервер MaxPatrol EDR.
5. В раскрывающемся списке **Версия модуля** выберите версию модуля, которую вы хотите импортировать.
6. Если вы хотите перезаписать на сервере файлы и конфигурации загружаемых версий модуля, установите флажок **Перезаписать модуль**.

   Если вы не установите этот флажок и на сервере уже есть хотя бы одна загружаемая версия модуля, то не импортируется ни одна из версий.
7. Нажмите кнопку **Импортировать**.

Модуль импортирован.

4.10. Удаление версии модуля

Вы можете удалить любую версию модуля из репозитория. Если на агентах установлен модуль этой версии, то он продолжит работу. При этом добавить эту версию в другие политики будет невозможно.

► Чтобы удалить версию модуля:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. В раскрывающемся списке **<Номер версии>** выберите версию модуля, которую вы хотите удалить.

4. Нажмите кнопку  и в раскрывшемся меню выберите пункт **Только версию <Номер версии>**.

5. Подтвердите удаление.

Версия модуля удалена.

4.11. Удаление модуля

Вы можете полностью удалить модуль из репозитория. В этом случае он перестанет работать на всех агентах, на которых был установлен.

► Чтобы удалить модуль:

1. В главном меню в разделе **EDR** выберите пункт **Модули**.

   Откроется страница **Модули EDR**.

2. Нажмите на название модуля.

   Откроется карточка модуля.

3. Выполните одно из следующих действий:

   - Если у модуля есть только одна версия, нажмите  .
   - Если у модуля есть несколько версий, нажмите кнопку  и в раскрывшемся меню выберите пункт **Модуль целиком**.

4. Подтвердите удаление.

Модуль удален.

5. Мониторинг состояния MaxPatrol EDR

Вы можете отслеживать работу сервера MaxPatrol EDR, агентов, модулей и внутренних компонентов, анализируя специальные метрики и данные трассировки. Для мониторинга состояния MaxPatrol EDR вместе с продуктом устанавливаются следующие сервисы:

— OpenTelemetry — для передачи данных трассировки с агента на сервер MaxPatrol EDR;

— Jaeger — для работы с данными трассировки;

— Elasticsearch — для хранения данных трассировки;

— VictoriaMetrics — для хранения метрик;

— Grafana — для визуализации, мониторинга и анализа метрик и данных трассировки;

— Grafana Loki — для хранения и просмотра журналов.

Рисунок 3. Мониторинг MaxPatrol EDR в Grafana

В этом разделе

Просмотр записей в системном журнале (см. раздел 5.1)

Поиск и просмотр данных об ошибках агента (см. раздел 5.2)

Поиск и просмотр данных об ошибках сервера агентов (см. раздел 5.3)

Поиск и просмотр данных об ошибках модуля (см. раздел 5.4)

Работа с дашбордами (см. раздел 5.5)

Построение графика метрики (см. раздел 5.6)

5.1. Просмотр записей в системном журнале

Вы можете просмотреть записи о работе системы с помощью сервиса Grafana Loki.

- Чтобы просмотреть записи в системном журнале:

  1. Войдите в веб-интерфейс Grafana.

     Примечание. Подробные инструкции по работе с Grafana приведены на сайте вендора.

  2. В панели слева нажмите  .

     Откроется страница **Explore**.

  3. В раскрывающемся списке сверху выберите источник данных **Loki**.

  4. Выберите режим редактирования запроса **Code**.

  5. Нажмите кнопку **Log browser**.

  6. В блоке параметров **Select labels to search in** выберите метки, по которым нужно искать записи в журнале.

  7. В блоке параметров **Find values for the selected labels** укажите значения выбранных меток.

     Например, для поиска записей об ошибках в работе какого-либо модуля на агенте вы можете выбрать идентификатор агента, модуль и уровень записи **ERROR** с помощью меток **agent_id**, **component** и **level**.

  8. Нажмите кнопку **Show logs**.

     Отобразятся записи из системного журнала, подходящие под условия запроса.

5.2. Поиск и просмотр данных об ошибках агента

- Чтобы просмотреть данные об ошибках агента:

  1. Войдите в веб-интерфейс Grafana.

     Примечание. Подробные инструкции по работе с Grafana приведены на сайте вендора.

  2. В панели слева нажмите  .

     Откроется страница **Explore**.

  3. В раскрывающемся списке сверху выберите источник данных **Jaeger**.

  4. Выберите тип запроса **Search**.

5. В раскрывающемся списке **Service** выберите компонент **vxagent**.
6. Выполните одно из следующих действий:

   Если вы хотите отобразить ошибки всех агентов, в поле **Tags** введите `error=true`.

   Если вы хотите отобразить ошибки конкретного агента, в поле **Tags** введите `error=true agent_id="<Идентификатор агента>"`.
7. Нажмите кнопку **Run query**.

   Отобразится таблица данных трассировки.
8. Выберите запись о трассировке.

   Справа откроется панель с данными об ошибке.

5.3. Поиск и просмотр данных об ошибках сервера агентов

- Чтобы просмотреть данные об ошибках сервера агентов:

1. Войдите в веб-интерфейс Grafana.

   Примечание. Подробные инструкции по работе с Grafana приведены на сайте вендора.
2. В панели слева нажмите  .

   Откроется страница **Explore**.
3. В раскрывающемся списке сверху выберите источник данных **Jaeger**.
4. Выберите тип запроса **Search**.
5. В раскрывающемся списке **Service** выберите компонент **vxserver**.
6. В поле **Tags** введите `error=true`.
7. Нажмите кнопку **Run query**.

   Отобразится таблица данных трассировки.
8. Выберите запись о трассировке.

   Справа откроется панель с данными об ошибке.

5.4. Поиск и просмотр данных об ошибках модуля

- Чтобы просмотреть данные об ошибках модуля:

1. Войдите в веб-интерфейс Grafana.

Примечание. Подробные инструкции по работе с Grafana приведены на сайте вендора.

2. В панели слева нажмите  .

   Откроется страница **Explore**.

3. В раскрывающемся списке сверху выберите источник данных **Jaeger**.
4. Выберите тип запроса **Search**.
5. В раскрывающемся списке **Service** выберите компонент **vxagent**.
6. Выполните одно из следующих действий:

   Если вы хотите отобразить ошибки модуля со всех агентов, в поле **Tags** введите `error=true log.module="<Идентификатор модуля>"`.

   Если вы хотите отобразить ошибки модуля с конкретного агента, в поле **Tags** введите `error=true log.module="<Идентификатор модуля>" agent_id="<Идентификатор агента>"`.

7. Нажмите кнопку **Run query**.

   Отобразится таблица данных трассировки.

8. Выберите запись о трассировке.

   Справа откроется панель с данными об ошибке.

5.5. Работа с дашбордами

Дашборд в Grafana — это страница, на которой отображаются панели с графиками, диаграммами и прочей статистической информацией о работе продукта и операционной системы.

При установке MaxPatrol EDR в Grafana добавляются несколько стандартных дашбордов для мониторинга состояния продукта.

- Чтобы открыть дашборд:

  1. Войдите в веб-интерфейс Grafana.

     Примечание. Подробные инструкции по работе с Grafana приведены на сайте вендора.

  2. В левом верхнем углу нажмите кнопку **General / Home**.

     Откроется страница со списком папок и дашбордов по компонентам продукта.

  3. Выберите дашборд.

     Откроется страница с дашбордом.

5.6. Построение графика метрики

Вы можете анализировать метрики в Grafana на графиках.

- Чтобы построить график метрики:

  1. Войдите в веб-интерфейс Grafana.

     Примечание. Подробные инструкции по работе с Grafana приведены на сайте вендора.

  2. В панели слева нажмите  .

     Откроется страница **Explore**.

  3. В раскрывающемся списке сверху выберите источник данных **VictoriaMetrics**.

  4. В поле **Metrics browser** введите метрику или выберите ее в списке.

  5. Нажмите кнопку **Run query**.

     Отобразится график метрики.

6. Обращение в службу технической поддержки

Техническая поддержка продукта включает в себя следующие услуги:

- решение вопросов эксплуатации продукта, помощь в использовании его функциональных возможностей;
- диагностику сбоев продукта, включая поиск причины сбоя и информирование клиента о найденных проблемах;
- разрешение проблем с продуктом, предоставление решений или возможностей обойти проблему с сохранением всей необходимой производительности;
- устранение ошибок в продукте (в рамках выпуска обновлений к продукту).

Вы можете получать техническую поддержку на портале.

Этот раздел содержит информацию о способах и условиях получения технической поддержки.

В этом разделе

6.1. Техническая поддержка на портале

Портал предоставляет вам возможность создавать запросы на техническую поддержку.

Вы можете создать учетную запись на портале, используя адреса электронной почты, расположенные на официальном домене вашей организации. Вы также можете указывать другие адреса электронной почты для учетной записи в качестве дополнительных. Для оперативной связи укажите в профиле учетной записи название вашей организации и контактный телефон.

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.

Техническая поддержка на портале предоставляется на русском и английском языках.

6.2. Время работы службы технической поддержки

На портале технической поддержки вы можете круглосуточно создавать и обновлять запросы, читать новости продуктов и пользоваться базой знаний.

6.3. Как служба технической поддержки работает с запросами

При получении вашего запроса специалист службы технической поддержки классифицирует его (присваивает запросу тип и уровень значимости) и выполняет дальнейшие шаги по выполнению запроса.

В этом разделе

6.3.1. Предоставление информации для технической поддержки

При обращении за технической поддержкой по первому требованию специалиста Positive Technologies нужно предоставить:

- номер лицензии на использование продукта;
- файлы журналов и другие наборы диагностических данных, хранящихся в продукте;
- снимки экрана;
- результаты выполнения рекомендаций специалиста технической поддержки;
- каналы для удаленного доступа к продукту (по взаимному согласованию оптимального канала диагностики).

Positive Technologies не несет обязательств по оказанию технической поддержки в случае отказа предоставить указанную выше информацию.

Если информация по обращению не предоставлена в течение значительного времени (от двух недель с момента последней активности), специалист технической поддержки имеет право считать ваше обращение неактуальным и, уведомив вас, закрыть запрос.

6.3.2. Типы запросов

Специалист технической поддержки относит ваш запрос к одному из следующих типов.

Вопросы по установке, повторной установке и предстартовой настройке продукта

Подразумевается помощь в подготовке продукта к работе, ответы на вопросы на данном этапе эксплуатации продукта. Техническая поддержка по этим вопросам доступна в течение 30 дней с момента активации продукта.

Вопросы по администрированию и настройке продукта

Включают в себя вопросы, возникающие в процессе эксплуатации продукта, рекомендации по оптимизации и настройке параметров продукта.

Восстановление работоспособности продукта

В случае критического сбоя и потери доступа к основной функциональности продукта специалист Positive Technologies оказывает помощь в восстановлении работоспособности продукта. Восстановление заключается либо в помощи по установке продукта заново с потенциальной потерей накопленных до сбоя данных, либо в откате продукта на доступную резервную копию (резервное копирование должно быть настроено заблаговременно). Positive Technologies не несет ответственность за потерю данных в случае неверно настроенного резервного копирования.

Обновление продукта

Positive Technologies поставляет пакеты обновления в течение срока обновления, указанного в лицензии на продукт.

Positive Technologies не несет ответственности за проблемы, возникшие при нарушении регламентированного процесса обновления.

Устранение дефектов продукта

Если по результатам диагностики обнаружен дефект продукта, Positive Technologies обязуется предпринять разумные усилия по предоставлению обходного решения (если возможно), а также включить исправление дефекта в ближайшие возможные обновления продукта.

6.3.3. Время реакции и приоритизация запросов

Время реакции на запрос рассчитывается с момента получения запроса до первичного ответа специалиста технической поддержки с уведомлением о взятии запроса в работу.

Время обработки запроса рассчитывается с момента отправки уведомления о взятии вашего запроса в работу до предоставления описания дальнейших шагов по устранению проблемы либо классификации вопроса, указанного в запросе, как дефекта ПО и передачи запроса ответственным лицам для исправления дефекта.

Время реакции и время обработки зависят от указанного вами уровня значимости запроса (см. таблицу 1).

Специалист службы технической поддержки оставляет за собой право переопределять уровень значимости запроса по приведенным ниже критериям. Указанные сроки являются целевыми и подразумевают стремление и разумные усилия исполнителя для их соблюдения, но возможны отклонения от данных сроков по объективным причинам.

Таблица 1. Время реакции на запрос и время его обработки

Уровень значимости запроса	**Критерии значимости запроса**	**Время реакции на запрос**	**Время обработки запроса**
Критический	Аварийные сбои, полностью препятствующие штатной работе продукта (исключая первоначальную установку) либо оказывающие критическое влияние на бизнес	До 4 часов	Не ограничено
Высокий	Сбои, затрагивающие часть функциональности продукта и проявляющиеся в любых условиях эксплуатации либо оказывающие значительное влияние на бизнес	До 24 часов	Не ограничено
Обычный	Сбои, проявляющиеся в специфических условиях эксплуатации продукта либо не оказывающие значительного влияния на бизнес	До 24 часов	Не ограничено
Низкий	Вопросы информационного характера либо сбои, не влияющие на эксплуатацию продукта	До 24 часов	Не ограничено

Указанные часы относятся только к рабочему времени специалистов технической поддержки (времени обработки запроса).

6.3.4. Выполнение работ по запросу

По мере выполнения работ по вашему запросу специалист технической поддержки сообщает вам:

- о диагностике проблемы и ее результатах;
- о поиске решения или возможности обойти причины возникновения проблемы;
- о планировании и выпуске обновления продукта (если требуется для устранения проблемы).

Если по итогам обработки запроса необходимо внести изменения в продукт, Positive Technologies включает работы по исправлению в ближайшее возможное плановое обновление продукта (в зависимости от сложности изменений).

Работы по запросу считаются выполненными, если:

- предоставлено решение или возможность обойти проблему, не влияющая на производительность и критически важную функцию продукта;
- диагностирован дефект продукта, собрана техническая информация о дефекте и условиях его воспроизведения; исправление дефекта запланировано к выходу в рамках планового обновления продукта;
- проблема вызвана программными продуктами или оборудованием сторонних производителей, не подпадающих под гарантийные обязательства по продукту;
- проблема классифицирована как неподдерживаемая.

Приложение А. Файлы базовой конфигурации

Файлы базовой конфигурации расположены в папке `config`.

Таблица 2. Файлы базовой конфигурации

Файл	Описание
`info.json`	Общая информация о модуле
`config_schema.json`	Параметры модуля и их конфигурация
`default_config.json`	Значения параметров модуля по умолчанию
`current_config.json`	На этапе разработки модуля соответствует файлу `default_config.json`. При добавлении модуля в политику перезаписывается конфигурацией модуля в политике
`event_config_schema.json`	События и их конфигурация
`default_event_config.json`	Значения параметров событий по умолчанию, а также действия по умолчанию, которые назначены на событие
`current_event_config.json`	На этапе разработки модуля соответствует файлу `default_event_config.json`. При добавлении модуля в политику перезаписывается конфигурацией событий в политике
`action_config_schema.json`	Действия и их конфигурация
`default_action_config.json`	Значения параметров действий по умолчанию
`current_action_config.json`	На этапе разработки модуля соответствует файлу `default_action_config.json`. При добавлении модуля в политику перезаписывается конфигурацией действий в политике
`fields_schema.json`	Переменные и их конфигурация
`static_dependencies.json`	Указанные зависимости модуля
`dynamic_dependencies.json`	На этапе разработки модуля файл пуст. После добавления модуля в политику заполняется данными о зависимостях. Зависимости могут появиться, например, при назначении действий на события модуля
`locale.json`	Локализация модуля
`changelog.json`	Журнал изменений модуля

Приложение Б. Переменные модулей

В MaxPatrol EDR зафиксирован ряд переменных, которые используются модулями. Кроме того, некоторые модули в качестве переменных используют поля событий из MaxPatrol SIEM.

Таблица 3. Переменные модулей MaxPatrol EDR

Идентификатор	Тип данных	Описание
`credential`	String	Учетная запись
`malware_class`	String	Класс вредоносного ПО, которое было обнаружено при проверке в PT Sandbox
`object.account.type`	String	Тип учетной записи
`object.account.valid`	Bool	Присутствие учетной записи в списке локальных учетных записей, зарегистрированных в ОС
`object.is_malware`	Bool	Результат проверки файла модулем «Проверка файлов в PT Sandbox»: `true` — обнаружено вредоносное ПО, `false` — вредоносное ПО не обнаружено
`object.md5_hash`	String	Хеш-сумма файла-объекта (MD5)
`object.sha256_hash`	String	Хеш-сумма файла-объекта (SHA-256)
`object.size`	Integer	Размер файла-объекта
`rule_name`	String	Имя правила в модуле «YARA-сканер»
`rule_precision`	Number	Точность правила в модуле «YARA-сканер»
`rule_type`	String	Тип правила в модуле «YARA-сканер»
`rules`	String	Список правил, которые использовались при проверке модулем «YARA-сканер»
`subject.fullpath`	String	Путь к файлу-субъекту
`threat`	String	Информация об обнаруженном в PT Sandbox вредоносном ПО
`version`	String	Версия компонента

Переменные из таксономии MaxPatrol SIEM: `action`, `alert.context`, `alert.key`, `body`, `category.generic`, `category.high`, `category.low`, `chain_id`, `correlation_name`, `correlation_type`, `dst.asset`, `dst.fqdn`, `dst.host`, `dst.hostname`, `dst.ip`, `dst.mac`, `dst.port`, `event_src.asset`, `event_src.category`, `event_src.fqdn`, `event_src.host`, `event_src.hostname`, `event_src.ip`, `event_src.rule`, `event_src.subsys`, `event_src.title`, `event_src.vendor`, `importance`, `incident.aggregation.key`, `incident.aggregation.time_window`, `incident.aggregation.timeout`,

incident.attacking_addresses, incident.category, incident.severity, labels, object, object.account.domain, object.account.id, object.account.name, object.account.session_id, object.account.type, object.account.valid, object.fullpath, object.hash, object.id, object.name, object.new_value, object.path, object.process.cmdline, object.process.cwd, object.process.fullpath, object.process.guid, object.process.hash, object.process.id, object.process.meta, object.process.name, object.process.original_name, object.process.parent.cmdline, object.process.parent.fullpath, object.process.parent.guid, object.process.parent.id, object.process.parent.name, object.process.parent.path, object.process.path, object.process.version, object.property, object.query, object.state, object.storage.fullpath, object.storage.id, object.storage.name, object.storage.path, object.type, object.value, object.version, reason, src.asset, src.fqdn, src.host, src.hostname, src.ip, src.mac, src.port, status, subject, subject.account.domain, subject.account.id, subject.account.name, subject.account.privileges, subject.account.session_id, subject.name, subject.process.cmdline, subject.process.fullpath, subject.process.guid, subject.process.id, subject.process.meta, subject.process.name, subject.process.original_name, subject.process.parent.cmdline, subject.process.parent.fullpath, subject.process.parent.id, subject.process.parent.name, subject.process.parent.path, subject.process.path, uuid.

Примечание. Подробная информация об этих переменных приведена в разделе «Схема полей событий» в Руководстве разработчика из комплекта поставки MaxPatrol SIEM.

Глоссарий

агент EDR

Приложение, которое устанавливается на конечном устройстве для обеспечения работы модулей и связи с сервером агентов.

группа агентов EDR

Один или несколько агентов EDR, объединенных по определенному принципу для назначения им одних и тех же политик.

действие модуля

Операция, которую модуль выполняет на конечном устройстве. Запуск операции может выполняться по команде пользователя или автоматически при регистрации того или иного события ИБ.

зависимость

Условие, которое должно выполняться для корректной работы модуля агента.

конечное устройство

Оборудование, имеющее ценность для организации и подлежащее защите от киберугроз.

модуль агента

Приложение, которое запускается на агенте для выполнения основных функций продукта. Есть пять типов модулей: модули доставки и установки, сбора, интеграции, обнаружения, реагирования.

модуль доставки и установки

Модуль агента, который устанавливает и настраивает приложения, а также управляет конфигурацией ОС на конечном устройстве.

модуль обнаружения

Модуль агента, который анализирует собранные события, обнаруживает подозрительную и вредоносную активность на конечном устройстве — и регистрирует события ИБ.

модуль реагирования

Модуль агента, который пресекает подозрительную и вредоносную активность на конечном устройстве, выполняя действия в соответствии с политикой модуля обнаружения.

модуль сбора

Модуль агента, который собирает данные о событиях на конечном устройстве и передает их в модули обнаружения и в SIEM-системы.

поведенческий анализ

Технология выявления атак и киберугроз, основанная на анализе поведения файлов, приложений и пользователя в информационной системе.

политика конфигурации модулей агентов

Механизм управления поставкой модулей агентов в заданной конфигурации на конечные устройства. Политика состоит из перечня модулей и описания их конфигурации, который назначается группе агентов.

приоритет действия

Условная величина, которая определяет порядок выполнения действий при регистрации того или иного события ИБ.

сервер агентов

Серверное приложение, предназначенное для управления агентами и модулями.

управляющий сервер

Серверное приложение, предназначенное для управления конфигурацией системы.

positive
technologies

Positive Technologies — лидер рынка 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 более 3300 организаций по всему миру, в том числе 80% компаний из рейтинга «Эксперт-400». Positive Technologies — первая и единственная компания из сферы кибербезопасности на Московской бирже (MOEX: POSI), у нее более 170 тысяч акционеров.