MaxPatrol EPP версия 8.3

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: 01.11.2025

Содержание

1. Об этом документе

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

Комплект документации MaxPatrol EPP включает в себя следующие документы:

— Этот документ.

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

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

2. О MaxPatrol EPP

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

— отправлять данные о системных событиях и событиях ИБ в MaxPatrol 10;

— 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.

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

— удалить файл;

— завершить один или несколько процессов;

— заблокировать сетевой трафик;

— заблокировать учетную запись;

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

— отправить файл на проверку в PT Sandbox;

— запустить сканирование в режиме аудита и отправить результаты в MaxPatrol VM;

— заблокировать все сетевые соединения по IP-адресу;

— перенаправить DNS-запросы на IP-адрес;

— изолировать файл в зашифрованном хранилище.

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 EPP

MaxPatrol EPP состоит из серверной части и агентов, устанавливаемых на конечные устройства. Серверная часть MaxPatrol EPP 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 EPP — 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 EPP:

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 syslog-сервер или 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 и пересылаются на сервер агентов, в базу данных MaxPatrol SIEM или на syslog-сервер.
8. Агент передает метрики и данные трассировки (см. раздел 6) на сервер агентов.
9. Управляющий сервер получает обновления продукта и пакетов экспертизы с сервера обновлений.

Взаимодействие компонентов

При обычной установке управляющий сервер в системе один, а серверов агентов может быть несколько. При установке в отказоустойчивом кластере компонент `api` управляющего сервера может быть установлен на нескольких серверах. Компоненты Observability (см. раздел 6) для снижения сетевого трафика могут быть установлены на одних серверах с серверами агентов.

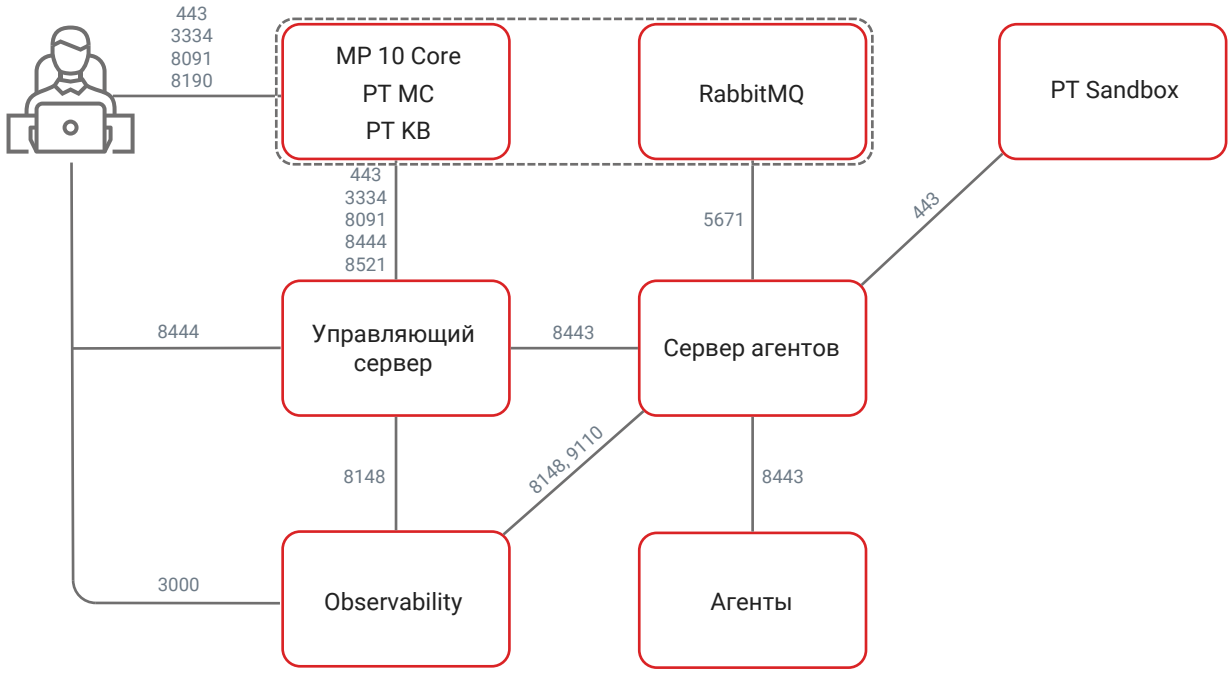

Рисунок 1. Взаимодействие компонентов MaxPatrol EPP

Для обеспечения сетевого взаимодействия компонентов MaxPatrol EPP должны быть доступны перечисленные ниже порты.

Примечание. Если какие-либо компоненты MaxPatrol EPP расположены на одном сервере, то обеспечивать внешнюю доступность портов при их взаимодействии необязательно. Например, при установке всех компонентов на один сервер открывать порты 8148, 8443, 9047, 9110 не требуется.

Примечание. В таблице приведены порты, используемые по умолчанию.

Таблица 1. Компоненты и порты взаимодействия

Источник	Получатель	Протокол	TCP-порт
Управляющий сервер	Сервер агентов	HTTPS	8443
Управляющий сервер	MP 10 Core	HTTPS	443, 3334, 8521

Источник	Получатель	Протокол	TCP-порт
Управляющий сервер	Компонент Observability	gRPC	8148
Управляющий сервер	Сервис пользовательской экспертизы (компонент `custom_expertise`)	HTTPS	9047 (при установке в отказоустойчивом кластере)
MP 10 Core	Управляющий сервер	HTTPS	8444
Сервер агентов	PT Sandbox	HTTPS	443
Сервер агентов	Сервер RabbitMQ	AMQP	5671
Сервер агентов	Компонент Observability	gRPC	8148
Сервер агентов	Компонент Observability	HTTPS	9110
Агент	Сервер агентов	WSS	8443
Рабочая станция пользователя	Управляющий сервер	HTTPS	8444
Рабочая станция пользователя	Управляющий сервер	SSH	22 (при необходимости для удаленного доступа по протоколу SSH)
Рабочая станция пользователя	Сервер агентов	SSH	22 (при необходимости для удаленного доступа по протоколу SSH)
Рабочая станция пользователя	MP 10 Core	HTTPS	443, 3334, 8091, 8190, 8444
Рабочая станция пользователя	Компонент `observability`	HTTPS	3000 (веб-интерфейс Grafana)
Внешние системы (взаимодействие через публичный API (см. раздел 5))	Управляющий сервер	HTTPS	8444
Сервер с ролью Deployer (если эта роль установлена отдельно от компонента MP 10 Core)	Управляющий сервер Сервер агентов	TCP	22

Источник	Получатель	Протокол	TCP-порт
	Компонент Observability		

См. также

Мониторинг состояния MaxPatrol EPP (см. раздел 6)

4. Разработка модулей в MaxPatrol EPP

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 EPP есть шесть типов модулей:

- **Системные модули.** Обеспечивают работу других модулей и агента.
- **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 EPP. При создании нового модуля вам доступны шаблоны для каждого типа модуля, которые определяют набор функций и методов в исходных файлах.

Модуль в MaxPatrol EPP 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 EPP, 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 EPP.

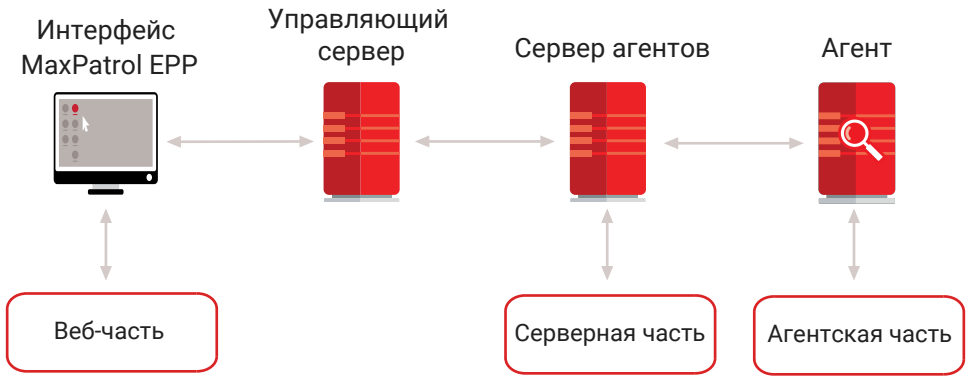

Рисунок 2. Структура модулей

4.2. Принципы разработки модулей

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

В этом разделе

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 EPP 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. Подпись кода модулей

Если при установке MaxPatrol EPP был активирован механизм проверки кода модулей, то для доставки разрабатываемых модулей на конечные устройства их код должен быть подписан. Для подписи кода вы можете использовать сертификаты, генерируемые автоматически при установке продукта, или собственные сертификаты. Подпись кода выполняется автоматически при создании модуля, при импорте и при каждом сохранении его конфигурации.

4.4. Создание модуля

- Чтобы создать модуль:

1. В главном меню выберите **Модули**.
2. Нажмите **Разработка модулей**.
3. Нажмите **Создать модуль**.
4. Введите идентификатор модуля.
5. Выберите шаблон, на основе которого будет создан модуль.
6. Введите номер первой версии модуля.
7. Выберите типы операционных систем, на которых будет работать модуль.
8. Нажмите **Создать**.

4.5. Разработка модуля

Для разработки серверной и агентской части модуля в MaxPatrol EPP 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

4.5.1. 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 EPP зафиксирован ряд переменных (см. приложение Б), которые используются модулями.

- Чтобы добавить переменную:

  1. В главном меню выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Выберите вкладку **Переменные**.
  5. Нажмите **Добавить**.
  6. Если требуется, чтобы все события и действия модуля использовали эту переменную, установите флажок **Да**.
  7. Введите идентификатор переменной.
  8. Выберите тип данных переменной.
  9. 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 EPP интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

10. Добавьте переменную хотя бы в одно событие (см. раздел 4.5.2) или действие (см. раздел 4.5.3).
11. Нажмите **Сохранить**.

4.5.2. 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

4.5.2.1. Добавление простого события

- Чтобы добавить простое событие:

1. В главном меню выберите **Модули**.
2. Нажмите **Разработка модулей**.
3. Нажмите на название модуля.
4. Выберите вкладку **События**.
5. В панели **Схема** нажмите **Добавить**.
6. Введите идентификатор события.
7. Если требуется, выберите переменные, которые будут передавать событие.

8. 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 EPP интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

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

10. Нажмите **Сохранить**.

4.5.2.2. Добавление агрегированного события

- Чтобы добавить агрегированное событие:

  1. В главном меню выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Выберите вкладку **События**.
  5. В панели **Схема** нажмите **Добавить**.
  6. Введите идентификатор события.
  7. Выберите тип события **aggregation**.
  8. 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**).

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

  10. Нажмите **Сохранить**.

4.5.2.3. Добавление корреляционного события

- Чтобы добавить корреляционное событие:

  1. В главном меню выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Выберите вкладку **События**.
  5. В панели **Схема** нажмите **Добавить**.
  6. Введите идентификатор события.
  7. Выберите тип события **correlation**.
  8. 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**).
  9. Если требуется, в панели **По умолчанию** выберите действия, которые будут выполняться по умолчанию при регистрации события.
  10. Нажмите **Сохранить**.

4.5.3. Добавление действия

Если модуль будет выполнять действия, вам нужно добавить их.

- Чтобы добавить действие:

  1. В главном меню выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Выберите вкладку **Действия**.
  5. В панели **Схема** нажмите **Добавить**.
  6. Введите идентификатор действия.
  7. Если требуется, выберите переменные, которые будет использовать действие.
  8. 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.

9. 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 EPP интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

10. Нажмите **Сохранить**.

4.5.4. 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

Добавление обычного параметра (см. раздел 4.5.4.1)

Добавление защищенного параметра (см. раздел 4.5.4.2)

4.5.4.1. Добавление обычного параметра

- Чтобы добавить обычный параметр:

  1. В главном меню выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Выберите вкладку **Конфигурация**.
  5. Нажмите **Добавить**.
  6. Если требуется, чтобы параметр был обязательным, установите флажок **Обязательное**.

7. Введите идентификатор параметра.
8. 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 EPP 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 **Сохранить**.

4.5.4.2. Добавление защищенного параметра

- Чтобы добавить защищенный параметр:

1. В главном меню выберите **Модули**.
2. Нажмите **Разработка модулей**.
3. Нажмите на название модуля.
4. Выберите вкладку **Защищенная конфигурация**.
5. Нажмите **Добавить**.
6. Если требуется, чтобы параметр был обязательным, установите флажок **Обязательное**.
7. Если значение параметра не нужно передавать на агенты, установите флажок **Использовать только на сервере**.
8. Введите идентификатор параметра.
9. Выберите тип данных параметра.
10. 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 EPP интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

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

4.5.5. 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. Нажмите **Разработка модулей**.
2. В главном меню выберите **Модули**.
3. Нажмите на название модуля.
4. Выберите вкладку **События**.
5. Раскройте конфигурацию события, нажав .
6. В разделе **Ключи конфигурации событий** нажмите **Добавить**.
7. Если требуется, чтобы параметр был обязательным, установите флажок **Обязательное**.
8. Введите идентификатор параметра.
9. Выберите тип данных параметра.
10. 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 EPP интегрирован генератор форм ncform. При разработке модуля вы можете использовать документацию вендора и визуальную площадку.

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

4.5.6. 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 выберите **Модули**.
2. Нажмите **Разработка модулей**.
3. Нажмите на название модуля.
4. Выберите вкладку **Зависимости**.
5. Если корректная работа модуля зависит от версии агента, выберите версию агента, начиная с которой модуль будет поддерживаться.
6. 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.
7. Если модуль будет отправлять данные в другой модуль, в блоке параметров **Отправка данных** добавьте этот модуль.
8. Нажмите **Сохранить**.

4.5.7. Работа с файлами

В MaxPatrol EPP вы можете работать с файлами модуля, в том числе используя встроенный редактор кода. Файлы хранятся на сервере MaxPatrol EPP 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.

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

4.5.7.1. Создание файла

► Чтобы создать файл:

1. В главном меню выберите **Модули**.
2. Нажмите **Разработка модулей**.
3. Нажмите на название модуля.
4. Выберите вкладку **Файлы**.
5. Напротив каталога, в котором вы хотите создать файл, нажмите ⋮ и в раскрывшемся меню выберите пункт **Создать**.
6. Введите путь и имя файла.
7. Нажмите **Создать**.

4.5.7.2. Редактирование файла

► Чтобы отредактировать файл:

1. В главном меню выберите **Модули**.
2. Нажмите **Разработка модулей**.

3. Нажмите на название модуля.
4. Выберите вкладку **Файлы**.
5. Нажмите на имя файла.
6. Внесите изменения в файл.
7. Нажмите **Сохранить**.

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

4.5.7.3. Загрузка файла на сервер

- Чтобы загрузить файл на сервер:

1. В главном меню выберите **Модули**.
2. Нажмите **Разработка модулей**.
3. Нажмите на название модуля.
4. Выберите вкладку **Файлы**.
5. Напротив каталога, в который вы хотите загрузить файл, нажмите и в раскрывшемся меню выберите пункт **Загрузить**.
6. Выберите файл.
7. Введите путь к каталогу, в который вы хотите загрузить файл.
8. Нажмите **Загрузить**.

4.5.7.4. Скачивание файла

- Чтобы скачать файл:

1. В главном меню выберите **Модули**.
2. Нажмите **Разработка модулей**.
3. Нажмите на название модуля.
4. Выберите вкладку **Файлы**.
5. Напротив файла, который вы хотите скачать, нажмите и в раскрывшемся меню выберите пункт **Скачать**.

4.5.7.5. 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 4.5.7.4) и загрузки файла на сервер (см. раздел 4.5.7.3).

- Чтобы переместить файл в другой каталог:

  1. В главном меню выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Выберите вкладку **Файлы**.
  5. Напротив файла, который вы хотите переместить, нажмите ⋮ и в раскрывшемся меню выберите пункт **Переместить**.
  6. Введите новый путь к файлу.
  7. Нажмите **Переместить**.

4.5.7.6. 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 выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Выберите вкладку **Файлы**.
  5. Напротив файла, который вы хотите удалить, нажмите ⋮ и в раскрывшемся меню выберите пункт **Удалить**.
  6. Нажмите **Удалить**.

4.5.8. 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 выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Выберите вкладку **Локализация**.
  5. Раскройте списки параметров необходимых объектов и локализуйте их.
  6. Нажмите **Сохранить**.

4.5.9. Добавление истории изменений версии модуля

До релиза версии модуля вы можете заполнить журнал ее изменений.

- Чтобы добавить историю изменений:

  1. В главном меню выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Выберите вкладку **История изменений**.
  5. Раскройте список с текущей версией модуля и добавьте историю ее изменений.
  6. Нажмите **Сохранить**.

4.5.10. 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.

4.6. 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 4.7). Если черновик использовался в политиках, то модуль будет автоматически обновлен в них после релиза.

- Чтобы выпустить релиз версии модуля:

  1. В главном меню выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Нажмите .
  5. В блоках параметров **Русский язык** и **Английский язык** заполните журнал изменений модуля на двух языках.
  6. Нажмите **Выпустить релиз**.

4.7. Создание новой версии модуля

После релиза версии модуля (см. раздел 4.6) вы не можете ее редактировать. Для дальнейшей разработки модуля вам нужно создать его новую версию.

- Чтобы создать новую версию модуля:

  1. В главном меню выберите **Модули**.
  2. Нажмите **Разработка модулей**.
  3. Нажмите на название модуля.
  4. Нажмите .
  5. Введите номер новой версии модуля.
  6. В блоках параметров **Русский язык** и **Английский язык** заполните журнал изменений модуля на двух языках.
  7. Нажмите **Создать черновик**.

См. также

Выпуск релиза версии модуля (см. раздел 4.6)

4.8. 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 выберите **Модули**.
2. Нажмите **Разработка модулей**.
3. Нажмите на название модуля.
4. Нажмите .

4.9. 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 выберите **Модули**.
2. Нажмите **Разработка модулей**.
3. Нажмите на название модуля.
4. Если вы хотите экспортировать только определенную версию модуля, выберите ее в раскрывающемся списке **<Номер версии>**.
5. Выполните одно из следующих действий:
   - Если у модуля есть только одна версия, нажмите **Экспортировать**.
   - Если у модуля есть несколько версий и вы хотите экспортировать только открытую, нажмите **Экспортировать** и выберите пункт **Только версию <Номер версии>**.
   - Если у модуля есть несколько версий и вы хотите экспортировать их все, нажмите **Экспортировать** и выберите пункт **Все версии**.

5. Публичный API

С помощью публичного API MaxPatrol EPP вы можете запускать реагирование на конечных устройствах. Для получения схемы API вам необходимо обратиться в службу технической поддержки Positive Technologies.

Корневой URL API:

```
https://<Адрес MaxPatrol EPP>:8444/api/edr/external/v1
```

Например:

```
https://edr.example:8444/api/edr/external/v1
```

Внимание! Перед выполнением запросов необходимо пройти авторизацию и получить токен доступа в PT MC. Подробные инструкции по этим операциям приведены в документации к системе MaxPatrol 10.

Примечание. Для выполнения запросов у полученного токена доступа должны быть соответствующие привилегии.

6. Мониторинг состояния MaxPatrol EPP

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

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

— 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 EPP в Grafana

В этом разделе

Просмотр записей в системном журнале (см. раздел 6.1)

Работа с дашбордами (см. раздел 6.2)

Построение графика метрики (см. раздел 6.3)

6.1. 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 производителя.

  2. В панели слева нажмите  .
  3. В раскрывающемся списке сверху выберите источник данных **Loki**.
  4. Нажмите **Log browser**.
  5. В блоке параметров **Select labels to search in** выберите метки, по которым нужно искать записи в журнале.
  6. В блоке параметров **Find values for the selected labels** укажите значения выбранных меток.

     Например, для поиска записей об ошибках в работе сервера агентов вы можете выбрать идентификатор сервера и уровень записи **ERROR** с помощью меток **server_id** и **level**.
  7. Нажмите **Show logs**.

6.2. Работа с дашбордами

Дашборд в Grafana — это страница с графиками, диаграммами и прочей статистической информацией о работе той или иной IT-системы.

При установке MaxPatrol EPP 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 производителя.

  2. В левом верхнем углу нажмите **General / Home**.
  3. Выберите дашборд.

6.3. 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 производителя.

2. В панели слева нажмите  .
3. В раскрывающемся списке сверху выберите источник данных **VictoriaMetrics**.
4. В поле **Metrics** введите метрику или выберите ее в списке.
5. Нажмите **Run query**.

7. О технической поддержке

Базовая техническая поддержка доступна для всех владельцев действующих лицензий на MaxPatrol EPP 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 поставляет 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 порталу 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 2. 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.

Приложение А. Файлы базовой конфигурации

Файлы базовой конфигурации расположены в папке `config`.

Таблица 3. 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 EPP зафиксирован ряд переменных, которые используются модулями. Кроме того, некоторые модули в качестве переменных используют поля событий из MaxPatrol SIEM.

Таблица 4. Переменные модулей MaxPatrol EPP

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 документации MaxPatrol SIEM.

Глоссарий

агент

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

группа агентов

Один или несколько агентов, 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 — 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. Positive Technologies — первая и единственная компания из сферы кибербезопасности на Московской бирже (MOEX: POSI). Количество акционеров превышает 220 тысяч.