



Container Security версия 1.2

Руководство администратора

© Positive Technologies, 2026.

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: 25.08.2026

Содержание

1.	Об этом документе.....	5
1.1.	Условные обозначения.....	5
1.2.	Другие источники информации о PT CS.....	6
2.	О PT CS.....	7
3.	Архитектура PT CS.....	10
3.1.	Подсистема «Сервер».....	12
3.2.	Подсистема «Клиент сканирования».....	15
3.3.	Подсистема «Мониторинг среды выполнения контейнеров».....	15
4.	Лицензирование.....	17
5.	Развертывание PT CS.....	20
5.1.	Аппаратные и программные требования.....	22
5.2.	Быстрая установка.....	27
5.3.	Установка из дистрибутива.....	28
5.4.	Установка из онлайн-реестра.....	30
5.5.	Работа с Helm-чартом.....	32
5.6.	Публикация Helm-чарта и образов PT CS в реестре.....	47
5.7.	Настройка подключения к веб-интерфейсу PT CS.....	50
5.8.	Настройка проверки и мониторинга.....	52
5.8.1.	Настройка проверки через контроллер допуска.....	52
5.8.2.	Настройка мониторинга событий среды выполнения.....	54
5.9.	Развертывание в закрытом контуре.....	56
6.	Порядок первоначальной настройки PT CS.....	57
7.	Интерфейс PT CS.....	58
8.	О карте кластера.....	62
9.	Мониторинг и реагирование на события среды выполнения.....	65
9.1.	Управление источниками отслеживания событий среды выполнения.....	67
9.2.	Управление параметрами отслеживания и фильтрами событий.....	71
9.3.	Управление детекторами.....	73
10.	Управление реестрами образов.....	76
10.1.	Настройка реестров для подключения к PT CS.....	77
10.2.	Работа с реестрами образов.....	78
11.	Управление сервисами и шаблонами уведомлений.....	80
11.1.	Управление сервисами и шаблонами уведомлений по электронной почте.....	82
11.2.	Управление сервисами и шаблонами уведомлений с помощью механизма webhook.....	85
11.3.	Управление сервисами и шаблонами уведомлений по протоколу syslog.....	87
11.4.	Параметры шаблонов уведомлений.....	89
12.	Интеграция с продуктами Positive Technologies.....	110
12.1.	Интеграция PT CS с PT Sandbox и PT MultiScanner.....	110
12.2.	Интеграция PT CS с MaxPatrol SIEM.....	112
13.	Администрирование PT CS.....	114
13.1.	Состав компонентов ПО и его среды функционирования.....	114
13.2.	О ролях пользователей.....	115
13.3.	Управление дочерними кластерами.....	118

13.3.1.	Подключение дочернего кластера к PT CS	120
13.3.2.	Удаление дочернего кластера.....	126
13.4.	Управление учетными записями пользователей.....	127
13.4.1.	Управление локальными учетными записями.....	129
13.4.2.	Управление корпоративными учетными записями	129
13.5.	Мониторинг состояния PT CS.....	133
13.5.1.	Работа с дашбордами.....	133
13.5.2.	Просмотр записей в системном журнале.....	135
13.5.3.	Формат сообщений в системном журнале.....	136
14.	Настройка учетной записи пользователя.....	139
14.1.	Смена своего пароля	139
14.2.	Управление токенами доступа	139
15.	Обновление PT CS	142
16.	Настройка локального сервера обновлений.....	145
16.1.	Аппаратные и программные требования к локальному серверу обновлений.....	145
16.2.	Установка локального сервера обновлений	146
16.3.	Активация лицензии локального сервера обновлений	146
16.4.	Настройка локального сервера обновлений для работы через прокси-сервер.....	147
16.5.	Ручной перенос обновлений PT CS в закрытый сегмент сети	149
16.6.	Настройка автоматического переноса обновлений PT CS в закрытый сегмент сети	150
16.7.	Конфигурационный файл локального сервера обновлений	152
16.8.	Проверка и изменение параметров локального сервера обновлений	154
17.	Удаление PT CS.....	157
18.	Решение проблем.....	158
19.	О технической поддержке	163
Приложение А.	Список поддерживаемых баз данных угроз и уязвимостей	167
Приложение Б.	Список поддерживаемых платформ контейнеризации	168
Глоссарий.....		169

1. Об этом документе

Руководство администратора содержит справочную информацию и инструкции по развертыванию, настройке и администрированию Positive Technologies Container Security (далее также — PT CS). Руководство не содержит инструкций по использованию основных функций продукта.

Руководство адресовано специалистам, выполняющим установку, первоначальную настройку и администрирование PT CS.

Комплект документации PT CS включает в себя следующие документы:

- Этот документ.
- Руководство пользователя — содержит подробную информацию о сценариях работы с продуктом, о настройке функций продукта для решения конкретных задач.
- Справочное руководство по REST API — содержит информацию о доступных функциях сервиса REST API в PT CS.

В этом разделе

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

[Другие источники информации о PT CS \(см. раздел 1.2\)](#)

1.1. Условные обозначения

В документе приняты условные обозначения.

Таблица 1. Условные обозначения

Пример	Описание
Внимание! При выключении модуля снижается уровень защищенности сети	Предупреждения. Содержат информацию о действиях или событиях, которые могут иметь нежелательные последствия
Примечание. Вы можете создать дополнительные отчеты	Примечания. Содержат советы, описания важных частных случаев, дополнительную или справочную информацию, которая может быть полезна при работе с продуктом
▶ Чтобы открыть файл:	Начало инструкции выделено специальным значком
Нажмите OK	Названия элементов интерфейса (например, кнопок, полей, пунктов меню) выделены полужирным шрифтом
Выполните команду <code>Stop-Service</code>	Текст командной строки, примеры кода, прочие данные, которые нужно ввести с клавиатуры, выделены специальным шрифтом. Также выделены специальным шрифтом имена файлов и пути к файлам и папкам

Пример	Описание
Ctrl+Alt+Delete	Комбинация клавиш. Чтобы использовать комбинацию, клавиши нужно нажимать одновременно
<Название программы>	Переменные заключены в угловые скобки

1.2. Другие источники информации о PT CS

Вы можете найти дополнительную информацию о PT CS на справочном портале help.ptsecurity.com и на [Позитивном портале](#). Через Позитивный портал вы также можете обратиться [в техническую поддержку \(см. раздел 19\)](#).

2. О PT CS

Positive Technologies Container Security (далее также — PT CS) — система, предназначенная для обеспечения комплексной безопасности контейнеризированных инфраструктур.

PT CS обеспечивает безопасность артефактов контейнеризации (образов, контейнеров, оркестратора, репозитория образов, ОС узлов) на всех этапах жизненного цикла разработки ПО.

Вы можете использовать PT CS [для защиты нескольких кластеров \(см. раздел 13.3\)](#).

PT CS позволяет снизить риски, возникающие при использовании контейнеризации. К таким рискам относятся:

- Эксплуатация уязвимостей компонентов, конфигураций приложений и системных библиотек контейнеров.
- Нарушение прав доступа к компонентам оркестратора.
- Нарушение изоляции контейнеров при выполнении и запуск произвольного кода.
- Внедрение ВПО в образы.
- Нарушение целостности и достоверности образов.

PT CS обеспечивает выполнение следующих задач по управлению безопасностью в контейнерной инфраструктуре:

- Автоматизация управления уязвимостями в компонентах и ПО.
- Автоматизация обнаружения ВПО в образах.
- Автоматизация проверки достоверности и целостности образов.
- Автоматизация аудита конфигурации среды выполнения контейнеров согласно требованиям стандартов.
- Мониторинг и реагирование на инциденты информационной безопасности в среде выполнения контейнеров.

Автоматизация управления уязвимостями в компонентах и ПО

Злоумышленники могут использовать в своих целях устаревшие версии образов с уязвимостями или загружать образы с вложенными инструментами разведки в реестры организации. Если разработчики воспользуются такими образами, это может привести к реализации атаки. Кроме того, некоторые атаки могут быть направлены на снижение уровня защищенности сервисов путем изменения их конфигурации, например разрешения повышения привилегий контейнеров в среде выполнения (runtime).

PT CS позволяет снизить вероятность реализации таких атак благодаря перечисленным ниже возможностям:

- Интеграция в CI/CD для блокировки конфигураций контейнеров и образов, нарушающих требования стандартов, политик и лучших практик. Подробнее об интеграции в CI/CD см. в Руководстве пользователя.
- [Подключение реестров \(см. раздел 10\)](#) и регулярная проверка образов контейнеров для выявления устаревших или содержащих уязвимости образов. Подробнее о проверке образов см. в Руководстве пользователя.
- Интеграция с системами учета и управления дефектами благодаря возможности генерации отчетов о сканировании в машиночитаемых форматах, что позволяет напрямую выгружать результаты сканирования и ускорить процесс триажа уязвимостей. Подробнее о форматах отчетов о сканировании см. в Руководстве пользователя.
- Отправка уведомлений [с помощью механизма webhook \(см. раздел 11.2\)](#) для автоматизации реагирования на результаты сканирования, например для создания заявки на устранение уязвимости в системе управления проектами или в трекере задач.

Автоматизация управления ВПО в образах

Злоумышленники могут использовать ВПО, внедряя его в контейнерные образы. Внедрение ВПО в образы может привести к эксплуатации уязвимостей целевой системы, установлению постоянного доступа, краже конфиденциальных данных и реализации других рисков информационной безопасности. PT CS позволяет обнаруживать ВПО в образах, благодаря возможности интеграции с PT Sandbox и PT MultiScanner. Сканирование образов на ВПО может быть интегрировано в CI/CD-процесс.

Автоматизация проверки достоверности и целостности образов

Подпись образа гарантирует достоверность и целостность артефакта. PT CS позволяет автоматизировать проверку подписи образов. Проверка выполняется с помощью контроллера допуска. Если подпись отсутствует или недействительна система может заблокировать образ или отправить уведомление о событии в соответствии с настроенными правилами реагирования. Таким образом, проверка подписи образов устраняет риск внедрения поддельных или измененных образов еще до их развертывания в продуктивной среде.

Автоматизация аудита конфигурации среды выполнения контейнеров

При использовании контейнеров существуют риски, связанные с внесением несанкционированных изменений в конфигурацию среды выполнения для снижения уровня защиты или повышения привилегий для получения доступа к критически важным данным.

Администраторы из-за невнимательности или по другим причинам могут внести в конфигурацию изменения, которые позволят злоумышленникам реализовать техники из матрицы MITRE ATT&CK в среде контейнеризации (например, изменение прав или запуск сторонних процессов).

PT CS позволяет снизить риски нелегитимного доступа и несанкционированных изменений за счет использования собственной реализации механизма контроллера допуска (admission controller), который автоматически проверяет вносимые изменения на уязвимости и позволяет блокировать подозрительные действия, например доступ к терминалу контейнера. Подробнее о проверке через контроллер допуска см. в Руководстве пользователя.

Мониторинг и реагирование на события среды выполнения

Атаки, связанные с выполнением несанкционированного кода внутри контейнеров, могут быть направлены на нарушение изоляции контейнеров, а также на получение доступа к ОС узла и повышение привилегий на нем. Это может привести к нарушению работы всех сервисов или утечке критически важных данных; при этом злоумышленник останется незамеченным для средств защиты.

PT CS позволяет снизить риски таких атак [с помощью мониторинга событий и реагирования на инциденты информационной безопасности в контейнерах при их выполнении \(рантайме\)](#) (см. раздел 9).

Для мониторинга среды выполнения PT CS использует источники, реализованные с использованием технологии eBPF, и детекторы, реализованные с использованием технологии WebAssembly, которые позволяют в режиме реального времени обрабатывать десятки тысяч событий в секунду и выявлять аномалии в поведении контейнеров.

Система позволяет вам разрабатывать и добавлять источники и детекторы. Кроме того, вы можете изменять уже существующие источники.

Несмотря на то что классические инструменты не могут так эффективно обеспечивать безопасность контейнеризации, как специализированные продукты, часто должны использоваться общие подходы в управлении и учете инцидентов. Для этого реализована возможность интеграции PT CS с решениями SIEM и log management [по протоколу syslog](#) (см. раздел 11.3).

См. также

[Управление реестрами образов \(см. раздел 10\)](#)

[Управление сервисами и шаблонами уведомлений с помощью механизма webhook](#) (см. раздел 11.2)

[Управление сервисами и шаблонами уведомлений по протоколу syslog](#) (см. раздел 11.3)

3. Архитектура PT CS

PT CS состоит из следующих подсистем:

- Подсистема «Сервер» — сканирует образы контейнеров и оценивает уровень опасности образа по установленным правилам. Обеспечивает хранение событий сканирования образов и конфигураций, проверок действий в кластере, мониторинга событий среды выполнения, интеграцию с внешними системами уведомлений и реестрами образов контейнеров. Предоставляет доступ к API, позволяя автоматизировать выполнение стандартных операций, доступных в веб-интерфейсе продукта.
- Подсистема «Клиент сканирования» — компонент, развернутый вне кластера Kubernetes. Сканирует образы контейнеров, находящихся на внешних узлах, и блокирует сборки уязвимых контейнеров на сборщиках в CI-системах.
- Подсистема «Мониторинг среды выполнения контейнеров» — собирает события, связанные с активностью внутри контейнеров, анализирует их и ищет в них угрозы с помощью детекторов.

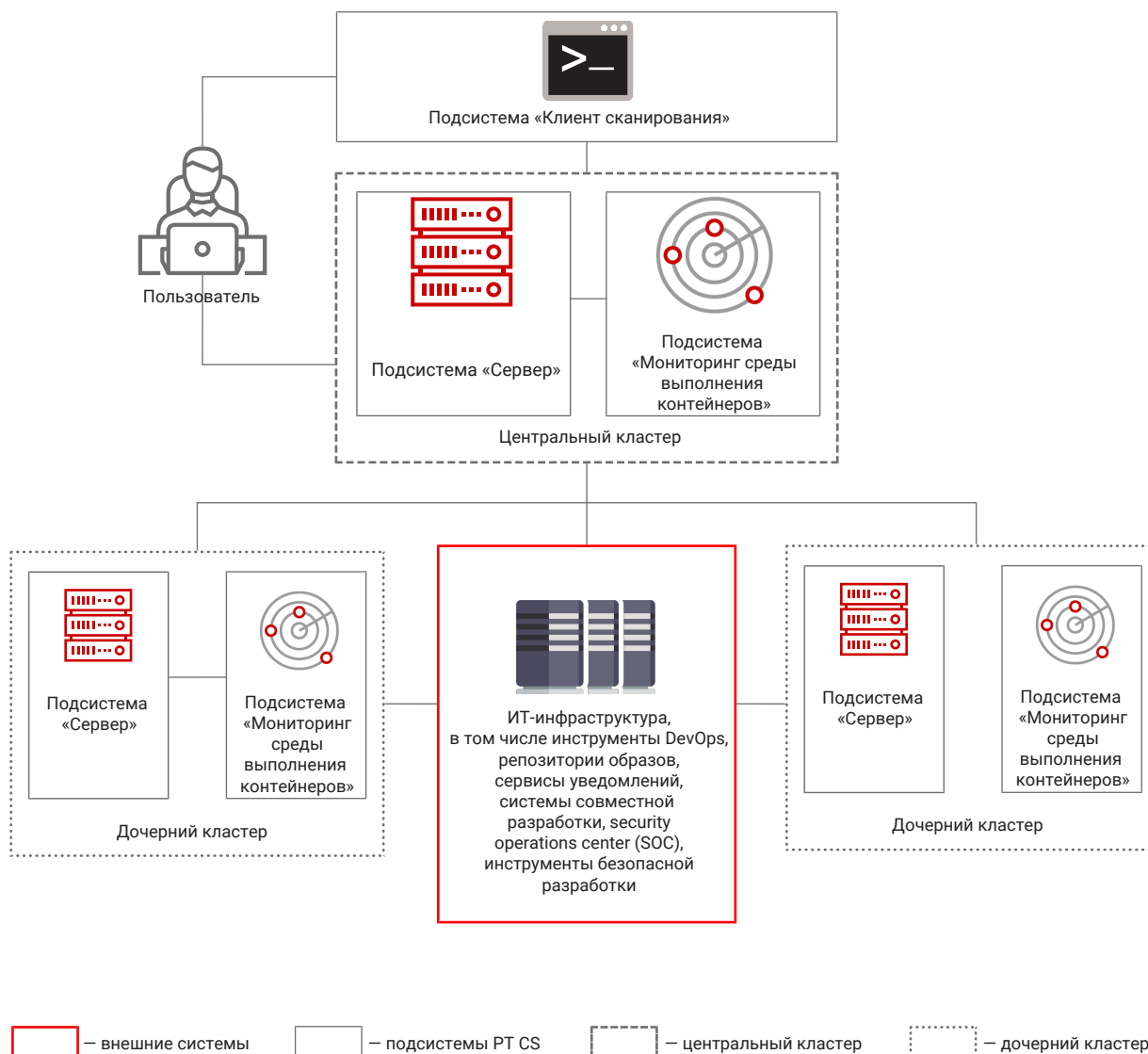


Рисунок 1. Взаимодействие компонентов PT CS

PT CS работает по следующему принципу:

- Подсистема «Клиент сканирования» передает в подсистему «Сервер» данные о компонентах сканируемого образа. Подсистема «Клиент сканирования» передает подсистеме «Сервер» логин и пароль учетной записи, от имени которой запускается сканирование, для аутентификации.
- Подсистема «Сервер» передает подсистеме «Клиент сканирования» результаты оценки образа и решение о блокировке сборки.
- Подсистема «Мониторинг среды выполнения контейнеров» отправляет результаты анализа событий в подсистему «Сервер».

PT CS собирает данные об уязвимостях в образах и событиях в контейнерах со всей ИТ-инфраструктуры, которая использует контейнеризацию, и отправляет данные об обнаруженных уязвимостях и угрозах в системы информационной безопасности.

В этом разделе

[Подсистема «Сервер» \(см. раздел 3.1\)](#)

[Подсистема «Клиент сканирования» \(см. раздел 3.2\)](#)

[Подсистема «Мониторинг среды выполнения контейнеров» \(см. раздел 3.3\)](#)

3.1. Подсистема «Сервер»

Подсистема «Сервер» состоит из набора компонентов, каждый из которых представляет собой микросервис и отвечает за определенную группу функций. Некоторые компоненты развертываются только в центральном кластере.

Информация о внутренних компонентах подсистемы приведена в таблице ниже.

Таблица 2. Внутренние компоненты подсистемы «Сервер»

Компонент	Описание	Устанавливается в дочерних кластерах при развертывании
Auth API	Выполняет следующие функции: — Управление учетными записями и ролями пользователей. — Аутентификация и авторизация пользователей. — Интеграция с внешними LDAP-серверами	Нет
Admission controller	Выполняет следующие функции: проверка запросов к API-серверу Kubernetes до того, как они будут сохранены в конфигурации кластера	Да
Image scanner	Выполняет следующие функции: — Сканирование образов на наличие уязвимостей и ВПО. — Формирование дерева зависимостей для сканируемого образа. — Поиск и сопоставление уязвимостей с компонентами образа. — Получение запроса на сканирование от утилиты ptcscli.	Да

Компонент	Описание	Устанавливается в дочерних кластерах при развертывании
	— Загрузка образа из реестра и его отправка на сканирование в PT MultiScanner. — Формирование и управление очередью запланированных задач на сканирование в соответствии с приоритетом. Приоритет устанавливается в зависимости от источника сканирования и выбора приоритета пользователем. — Отправка образа в компонент Policy enforcer для поиска соответствующих правил реагирования. — Отправка результата выполненного сканирования в компонент History API. — Отправка уведомления о событии в компонент Notifier	
IAC checker	Выполняет следующие функции: — Проверка конфигураций на уязвимости. — Поиск и сопоставление уязвимостей с компонентами конфигурации	Да
Reverse proxy	Выполняет следующие функции: обеспечение взаимодействия других компонентов PT CS по протоколу gRPC и предоставление графического пользовательского интерфейса	Да
Policy enforcer	Выполняет следующие функции: — Определение реакции (например, прервать CI/CD-процесс, отправить уведомление) по результатам сканирования. — Поиск всех правил, соответствующих текущей реакции, и предоставление рекомендаций о дальнейших действиях. — Предоставление интерфейса для управления правилами реагирования	Да
CS manager	Выполняет следующие функции: обновление БДУ и активация лицензии	Да
Cluster manager	Выполняет следующие функции: управление дочерними кластерами (регистрация, обновление информации и прочие операции)	Нет

Компонент	Описание	Устанавливается в дочерних кластерах при развертывании
History API	Выполняет следующие функции: сохранение, извлечение и передача в БД сведений о событиях системы. К таким сведениям относятся: — проведенные сканирования, их результаты; — компоненты, связанные с событиями; — выявленные уязвимости	Да
Notifier	Выполняет следующие функции: — Отправка уведомлений пользователям и внешним системам благодаря поддержке интеграции: • с сервисами электронной почты; • через webhook; • через syslog. — Хранение шаблонов уведомлений. — Рассылка уведомлений от системы	Да
Container registry integrator	Выполняет следующие функции: — Интеграция с реестрами контейнерных образов. — Авторизация в целевом реестре. — Получение списка образов. — Формирование перечня доступных образов в репозиториях. — Скачивание образов для последующего сканирования	Да
Public API	Выполняет следующие функции: предоставление программного интерфейса для автоматизации операций, доступных в веб-интерфейсе PT CS	Да
Kube manager	Выполняет следующие функции: — Подключение к API-серверу Kubernetes кластера, в котором запущен. — Мониторинг изменений состояния объектов кластера. — Удаление пода, событие в котором подпадает под правило с реакцией Блокировка (при необходимости).	Да

Компонент	Описание	Устанавливается в дочерних кластерах при развертывании
	— Проверка всех конфигураций кластера. — Отправка результата проверки в компонент History API	

Информация о внешних компонентах подсистемы приведена в таблице ниже.

Таблица 3. Внешние компоненты подсистемы «Сервер»

Компонент	Функции	Устанавливается в дочерних кластерах при развертывании
ClickHouse	Хранит данные подсистемы «Мониторинг среды выполнения контейнеров»	Да
PostgreSQL	Хранит конфигурации компонентов PT CS и данные о сканированиях	Да
RabbitMQ	Реализует очередь событий от компонента Runtime monitor подсистемы «Мониторинг среды выполнения контейнеров»	Да
Redis	Хранит очередь задач	Да

3.2. Подсистема «Клиент сканирования»

Подсистема «Клиент сканирования» представляет собой консольную утилиту `ptcs-cli`.

Утилита позволяет сканировать образы на наличие уязвимостей или ВПО и проверять манифесты Kubernetes. Утилита `ptcs-cli` позволяет запускать сканирование и проверку вручную или интегрировать их в CI/CD-процесс. Вы можете настроить блокировку CI/CD-процесса при обнаружении уязвимости или ВПО, создав правила реагирования с соответствующей реакцией.

3.3. Подсистема «Мониторинг среды выполнения контейнеров»

Информация о компонентах подсистемы приведена в таблице ниже.

Таблица 4. Компоненты подсистемы «Мониторинг среды выполнения контейнеров»

Компонент	Описание	Устанавливается в дочерних кластерах при развертывании
Runtime monitor	Выполняет следующие функции: — Сбор событий из запущенных контейнеров в соответствии с источниками, входящими в комплект поставки. — Настройка параметров отслеживания и фильтров событий. — Отправка собранных событий в очередь RabbitMQ подсистемы «Сервер». — Изменение и добавление источников отслеживания событий в режиме эксперта	Да
Event processor	Выполняет следующие функции: — Получение событий из очереди RabbitMQ подсистемы «Сервер». — Анализ событий и их проверка через цепочку детекторов. — Формирование результата и его передача компоненту Policy enforcer подсистемы «Сервер». — Управление детекторами (добавление и удаление)	Да

4. Лицензирование

Для работы с PT CS нужна действующая лицензия.

Лицензия определяет:

- количество сканеров, проверяющих образы;
- количество реестров, которые можно подключить для сканирования;
- количество узлов кластера, на которых будут запущены агенты мониторинга среды выполнения;
- количество кластеров, которые можно подключить для сканирования.

Активированная лицензия имеет ограниченный срок действия. По истечении этого срока добавление реестров, сканирование образов и конфигураций, проверка действий через контроллер допуска, установка агентов мониторинга среды выполнения и подключение дочерних кластеров становятся недоступны.

Лицензия может быть активирована только в экземпляре PT CS, установленном в центральном кластере. Дочерние кластеры получают информацию о лицензии из центрального.

Активация лицензии в веб-интерфейсе

После установки PT CS вам нужно активировать лицензию, чтобы получить доступ к функциям системы. Вы можете получить серийный номер лицензии, обратившись в Positive Technologies.

► Чтобы активировать лицензию:

1. Если требуется, настройте подключение к серверу лицензий через прокси-сервер, указав параметр `--proxy-url` при разворачивании PT CS.
2. Нажмите .

Примечание. Вы также можете нажать **Активируйте лицензию** в оповещении на любой странице.

3. Нажмите **Активировать лицензию**.
4. Введите серийный номер лицензии.
5. Нажмите **Активировать**.

После активации лицензии вам нужно дождаться, когда PT CS скачает последнюю версию баз уязвимостей с сервера обновлений update.ptsecurity.com.

Примечание. Автоматическое обновление баз уязвимостей может занимать длительное время, поэтому после активации лицензии рекомендуется запустить ручное обновление.

- ▶ Чтобы запустить ручное обновление баз уязвимостей,

выполните запрос:

```
curl -XPOST http://<Доменное имя или IP-адрес сервера PT CS>:<Порт>/api/v1/vulnkb/update
-H 'authorization: Bearer <Токен доступа (JWT)>'
```

- ▶ Чтобы получить токен доступа (JWT),

выполните команду:

```
curl --location https://<Доменное имя или IP-адрес сервера PT CS>:<Порт>/api/v1/signin --
form username="<Логин пользователя>" --form password="$(read -s -p 'password: ' p; echo
$p)" | yq .access_token
```

Замена лицензии в веб-интерфейсе

Замена лицензии может потребоваться в следующих случаях:

- Приобретена новая лицензия с обновленным сроком действия. Срок действия лицензии устанавливается при ее заказе. Если срок подходит к концу или истек, вы можете обратиться в техническую поддержку, чтобы продлить срок действия текущей лицензии или заказать новую.
- Одна и та же лицензия была активирована в нескольких экземплярах PT CS. Поскольку одна лицензия может использоваться только в одном экземпляре продукта, вам нужно активировать отдельную лицензию для каждого экземпляра.

Примечание. При нехватке лицензий вашей организации нужно их докупить.

- Необходимо изменить параметры лицензии.

- ▶ Чтобы заменить лицензию:

1. Нажмите .
2. Нажмите **Активировать новую лицензию**.
3. Введите серийный номер лицензии.
4. Нажмите **Активировать**.

Активация и замена лицензии с помощью командной строки

Помимо активации и замены лицензии в веб-интерфейсе, вы можете активировать или заменить лицензию с помощью командной строки.

- ▶ Чтобы активировать или заменить лицензию с помощью командной строки:

1. Получите токен доступа (JWT).
2. Выполните команду:

```
curl --location "https://<Доменное имя или IP-адрес сервера PT CS>/api/v1/license" --
header "Content-Type: application/json" \
--header "Accept: application/json" \
```

```
--header "Authorization: Bearer <Токен доступа (JWT)>" \
--data '{
  "serial_number": <Серийный номер лицензии>,
  "instance_id": <Уникальный идентификатор (UUID) экземпляра лицензии>
}'
```

Просмотр информации о лицензии дочернего кластера

► Чтобы просмотреть информацию о лицензии дочернего кластера:

1. Нажмите .
2. Вверху страницы справа выберите кластер, информацию о лицензии которого вы хотите просмотреть.

Отобразится информация о лицензии для выбранного кластера. Если информация о лицензии недоступна, то это означает, что [кластер не смог перейти в статус Установлен \(см. раздел 18\)](#).

Получение серийного номера лицензии из файла прошивки

► Чтобы получить серийный номер лицензии из файла прошивки:

1. Сохраните лицензию PT CS в файл `license-access-token.key`.

Примечание. Файл `license-access-token.key` закодирован по стандарту Base64.

2. Распознайте закодированный файл `license-access-token.key`, выполнив команду:


```
cat license-access-token.key | cut -d '.' -f2 | base64 -d | jq .
```

3. Из файла `license-access-token.key` получите серийный номер лицензии, выполнив команду:

```
cat license-access-token.key | cut -d '.' -f2 | base64 -d | jq -r .license_features.serial
```

5. Развертывание PT CS

PT CS может быть установлен для защиты одного кластера или для защиты мультикластерных систем. В этом разделе описано развертывание PT CS для защиты одного кластера или в центральном кластере мультикластерной системы. Для установки PT CS в дочерних кластерах мультикластерной системы вы можете воспользоваться [инструкцией по управлению дочерними кластерами \(см. раздел 13.3\)](#).

PT CS развертывается с помощью пакетного менеджера Helm для Kubernetes. Helm упрощает развертывание, управление приложениями и использует чарты, содержащие все необходимые ресурсы для запуска приложения в кластере.

Внимание! Если вы хотите развернуть PT CS для ознакомления или тестирования без настройки, вы можете воспользоваться [инструкцией по быстрой установке \(см. раздел 5.2\)](#). В этом случае конфигурационный файл Helm-чарта будет сгенерирован автоматически, все необходимые СУБД и сервисы будут развернуты автоматически, доступ к веб-интерфейсу будет организован через проброс порта.

В остальных случаях перед установкой рекомендуется:

- Убедиться, что сервер, на котором будет выполняться установка, соответствует [аппаратным и программным требованиям \(см. раздел 5.1\)](#).
- Обратиться к менеджеру компании Positive Technologies для получения лицензии.
- Внимательно ознакомиться с шагами по подготовке к установке, описанными ниже, и выбрать варианты, соответствующие требованиям вашей среды.

Шаг 1. Выбор способа установки

Вы можете установить PT CS:

- [Из дистрибутива \(см. раздел 5.3\)](#) — используя локальные Helm-чарт и образы из архива. Этот способ подходит для установки в изолированных средах, позволяет проверить подлинность дистрибутива. Потребуется Docker-реестр в инфраструктуре [для публикации и хранения образов контейнеров \(см. раздел 5.6\)](#) PT CS и больше ручных действий, чем при установке из онлайн-реестра.
- [Из онлайн-реестра \(см. раздел 5.4\)](#) — Helm-чарт и образы будут загружены с сервера обновлений update.ptsecurity.com во время установки. Этот способ требует наличия доступа к серверу update.ptsecurity.com, но обеспечивает быстрое начало развертывания и возможность установки самой актуальной версии PT CS.

Шаг 2. Выбор варианта развертывания СУБД и вспомогательных сервисов

Перед установкой PT CS в зависимости от требований к производительности и отказоустойчивости необходимо выбрать один из вариантов развертывания СУБД и вспомогательных сервисов:

- **Автоматическое развертывание** — СУБД и сервисы устанавливаются вместе с PT CS. Этот вариант подходит для тестовых сред и небольших нагрузок, дополнительных настроек не требуется.
- **Использование внешних СУБД и сервисов** — при выборе этого варианта вам потребуется указать данные для подключения к ним в конфигурационном файле Helm-чарта.

Шаг 3. Выбор настройки подключения к веб-интерфейсу

Вам необходимо выбрать [вариант настройки подключения к веб-интерфейсу \(см. раздел 5.7\)](#) PT CS. Настроить подключение можно с помощью:

- **Проброса порта** — можно настроить после установки PT CS.
- **Сервиса NodePort** — подключение через статический порт на каждом узле кластера, параметры подключения указываются при установке.
- **Ingress-контроллера** — требует предварительной установки Ingress-контроллера, параметры задаются до установки в конфигурации.

Шаг 4. Выбор варианта заполнения конфигурационного файла Helm-чарта

Все параметры установки PT CS задаются [в конфигурационном файле Helm-чарта \(см. раздел 5.5\)](#). Вам необходимо выбрать вариант заполнения конфигурационного файла. Этот файл можно:

- **Сгенерировать автоматически с помощью скрипта**. Такой вариант позволяет создать конфигурационный файл быстрее. Для многих параметров автоматически устанавливаются значения по умолчанию.
- **Заполнить вручную**. Такой вариант требует больше времени и погружения в технические детали.

Шаг 5. Выбор способа создания сертификатов для компонентов

При развертывании PT CS сертификаты для безопасного взаимодействия между компонентами генерируются автоматически и сохраняются в конфигурационный файл Helm-чарта. Если автоматическая генерация нежелательна, вы можете создать сертификаты заранее и вручную добавить их в конфигурационный файл.

Внимание! Параметры установки, заданные в конфигурационном файле Helm-чарта, нельзя изменить [без процедуры обновления \(см. раздел 15\)](#) (`helm upgrade`). Рекомендуется переходить к установке только после того, как вы убедитесь, что все необходимые решения приняты.

В этом разделе

[Аппаратные и программные требования \(см. раздел 5.1\)](#)

[Быстрая установка \(см. раздел 5.2\)](#)

[Установка из дистрибутива \(см. раздел 5.3\)](#)

[Установка из онлайн-реестра \(см. раздел 5.4\)](#)

[Работа с Helm-чартом \(см. раздел 5.5\)](#)

[Публикация Helm-чарта и образов PT CS в реестре \(см. раздел 5.6\)](#)

[Настройка подключения к веб-интерфейсу PT CS \(см. раздел 5.7\)](#)

[Настройка проверки и мониторинга \(см. раздел 5.8\)](#)

[Развертывание в закрытом контуре \(см. раздел 5.9\)](#)

5.1. Аппаратные и программные требования

Аппаратные требования основаны на результатах нагрузочного тестирования. Для тестирования использовался сервер с количеством подов по умолчанию, указанным в Helm-чарте. Требования могут меняться в зависимости от следующих факторов:

- количества одновременных сканирований образов;
- количества генерируемых событий от источников событий среды выполнения;
- количества пространств имен (namespaces), подов и нагрузки подов.

Каждый под с контейнерами сторонних приложений неизбежно, но непрогнозируемо увеличивает количество событий. Это ведет к повышению аппаратных требований. Для получения точных требований, основанных на параметрах вашей среды, обратитесь [в службу технической поддержки \(см. раздел 19\)](#) PT CS.

Аппаратные требования к серверу развертывания

Аппаратные требования к серверу развертывания PT CS приведены в таблице ниже.

Таблица 5. Аппаратные требования к серверу развертывания PT CS

Параметр	Минимальные требования
Центральный процессор	4 ядра
Память (ОЗУ)	8 ГБ

Параметр	Минимальные требования
Жесткий диск	20 ГБ

Вам необходимо обеспечить подключение к API кластера Kubernetes и реестру с образами PT CS.

Аппаратные требования к серверу с центральным кластером Kubernetes

Аппаратные требования к серверу с центральным кластером Kubernetes приведены в таблице ниже.

Внимание! Указанные требования необходимы для работы PT CS и не учитывают нагрузку, которая уже есть на сервере с центральным кластером Kubernetes. Для выполнения сканирования больших образов или при использовании нескольких реплик компонента Image scanner может потребоваться больше ресурсов.

Таблица 6. Аппаратные требования к серверу с центральным кластером Kubernetes

Параметр	Минимальные требования	Рекомендованные требования
Центральный процессор	8 ядер	10 ядер
Память (ОЗУ)	16 ГБ	32 ГБ
Жесткий диск	70 ГБ	130–180 ГБ

В зависимости от способа установки вам необходимо обеспечить сетевой доступ к серверу с реестром образов PT CS или к серверу обновлений update.ptsecurity.com.

Если вы используете СУБД PostgreSQL, RabbitMQ и ClickHouse из установочного дистрибутива PT CS, необходимо выделить дополнительные свободные ресурсы в центральном кластере Kubernetes, руководствуясь требованиями из таблицы ниже.

Таблица 7. Аппаратные требования к дополнительным ресурсам в центральном кластере Kubernetes при использовании PostgreSQL, RabbitMQ и ClickHouse из установочного дистрибутива PT CS

Параметр	Минимальные требования
Центральный процессор	8 ядер
Память (ОЗУ)	16 ГБ

Минимальный размер жесткого диска рассчитывается по разным формулам в зависимости от сохраняемых данных.

Средний размер события Tetragon равен 13 КБ. Вы можете самостоятельно рассчитать минимальный размер жесткого диска для сервера с ClickHouse в зависимости от среднего количества событий в секунду за рассчитываемый период R и количества дней хранения событий в доступе d по формуле $SSD = 13 \times R \times 86\,400 \times d$.

При расчете размера жесткого диска вы получите его размер в КБ. Для перевода в ГБ разделите значение на 10^6 .

Минимальный размер жесткого диска для сервера с PostgreSQL зависит:

- от размера события сканирования образа (зависит от количества компонентов и уязвимостей), s . В среднем размер события составляет 500 КБ, для образов с большим числом компонентов рекомендуется увеличить это значение на 25–30%;
- количества образов, которые нужно сканировать в рамках одного запуска периодического сканирования, l ;
- количества сканирований за период хранения, N_d . Например, если выполняется одно сканирование по расписанию в неделю, то за год будет выполнено 52 сканирования, за месяц — 4. Если сканирование выполняется дважды в день, то за месяц будет выполнено 62 сканирования.

Вы можете самостоятельно рассчитать минимальный размер жесткого диска для сервера с PostgreSQL по формуле $SSD = s \times l \times N_d$.

При расчете размера жесткого диска вы получите его размер в КБ. Для перевода в ГБ разделите значение на 10^6 .

Минимальный размер жесткого диска для сервера, хранящего события проверки через контроллер допуска и сканирования конфигураций, зависит:

- от количества запросов сервера API Kubernetes к компоненту Admission controller в секунду, N_r . Количество запросов можно определить [с помощью Grafana \(см. раздел 13.5\)](#);
- размера события сканирования, S_s . В одном событии сканирования, как правило, содержится информация об одном компоненте (за исключением IaC), поэтому можно считать, что средний размер события составляет 50–60 КБ;
- количества дней хранения событий, d ;

Вы можете самостоятельно рассчитать минимальный размер жесткого диска для сервера для хранения событий проверки через контроллер допуска и сканирования конфигураций по формуле $SSD = N_r \times S_s \times 86\,400 \times d$.

При расчете размера жесткого диска вы получите его размер в КБ. Для перевода в ГБ разделите значение на 10^6 .

Если вы используете PostgreSQL, RabbitMQ, ClickHouse и Redis (а также сертифицированные ФСТЭК России Postgres Pro, Arenadata QuickMarts T4) не из установочного дистрибутива PT CS, необходимо обеспечить подключение кластера к серверам с соответствующими СУБД.

Аппаратные требования к серверам с дочерними кластерами Kubernetes

Аппаратные требования к серверам с дочерними кластерами Kubernetes аналогичны требованиям к серверу с центральным кластером за исключением требований к сетевому доступу. Серверам с дочерними кластерами Kubernetes дополнительно необходим сетевой доступ к серверу с центральным кластером.

Аппаратные требования к серверу с реестром образов PT CS

Для сервера с реестром образов PT CS необходимо не менее 3 ГБ свободного дискового пространства.

Аппаратные требования к серверу с СУБД

Если не указаны дополнительные параметры, СУБД разворачивается в кластере Kubernetes. При разворачивании PT CS вы можете указать параметры для подключения к уже существующим СУБД. На серверах с внешними СУБД PostgreSQL и ClickHouse вам необходимо разрешить подключение по протоколу SSL. К серверам предъявляются требования, приведенные в таблице ниже.

Таблица 8. Аппаратные требования к серверу с СУБД

Параметр	Минимальное значение
Центральный процессор	8 ядер
Память (ОЗУ)	16 ГБ
Жесткий диск	500 ГБ
Твердотельный накопитель (SSD)	
Пропускная способность	Чтение 313 МБ/с, запись 156 МБ/с
IOPS	Чтение 10 000 операций в секунду, запись 5000 операций в секунду

Поддерживаемые ОС

Все компоненты PT CS поддерживают установку на 64-разрядные версии следующих ОС:

- Debian;
- CentOS;
- Ubuntu;
- Astra Linux.

Примечание. Тестирование выполнялось на следующих версиях ОС: Debian 11 (версия ядра 5.10), Debian 12 (версия ядра 6.1), Ubuntu 20.04 (версия ядра 5.4), Ubuntu 22.04 (версия ядра 5.15), Astra Linux 1.8 (версия ядра 6.6), CentOS 10 Stream (версия ядра 6.12).

Ядро ОС должно удовлетворять следующим требованиям:

- версия ядра ОС поддерживает [Tetragon](#);

Примечание. Рекомендуется использовать ОС с версией ядра 5.4 или выше. Для работы детектора **Использование Unix-сокета** (CS_RT_UNIX_SOCKET_USE) необходимо использовать ОС с версией ядра 5.11 или выше.

- в ядре включена функция BTF;
- в ядре включены модули eBPF и cgroup. С подробной информацией о модулях вы можете ознакомиться в официальной документации [Tetragon](#);
- в ядре отключены или переведены в разрешающий режим модули безопасности Linux (LSM), контролирующие запуск eBPF-программ:
 - для модуля lockdown установлен режим none или integrity;
 - для модуля Yama установлен режим 0 или 1;
 - модули AppArmor, SELinux, Parsec отключены для процессов [Tetragon](#) или к модулям ядра настроен точный доступ.

По вопросам поддержки ОС обращайтесь в [службу технической поддержки](#) (см. раздел 19).

Программные требования к серверу развертывания

Пользователю, от имени которого выполняются команды запуска скриптов, необходимы права на исполнение файлов в каталоге `bin/` распакованного архива с дистрибутивом PT CS.

На сервере развертывания необходимо разместить файл конфигурации `kubeconfig` для подключения к Kubernetes и установить менеджер пакетов Helm и утилиту `kubectl`.

Внимание! Вам необходимо выбирать версии Helm и `kubectl`, совместимые с используемой версией Kubernetes. Информацию о совместимости вы можете прочесть в [документации Helm](#) и [kubectl](#). Использование несовместимых версий может привести к непредвиденным последствиям.

Программные требования к серверам с центральным и дочерними кластерами Kubernetes

Необходимо использовать Kubernetes версии 1.20 или выше, RabbitMQ версии 4.0 или выше и следующие СУБД:

- Redis версии 7.0.11 или выше;
- PostgreSQL версии 14 или выше или Postgres Pro (сертификат ФСТЭК России № 3637);
- ClickHouse версии 20 или выше или Arenadata QuickMarts T4 (сертификат ФСТЭК России № 4823).

Если вы используете СУБД PostgreSQL, ClickHouse и Redis и сервис RabbitMQ из установочного дистрибутива PT CS, то при развертывании необходимо указать классы хранилищ для них.

Необходимо предварительно развернуть внешние базы данных и сервис RabbitMQ версии 3.12.6 или выше и при установке PT CS указать параметры для подключения к ним.

Программные требования к серверу с реестром образов PT CS

Необходимо использовать реестры, которые поддерживают Docker Registry API v2. Если реестры (например, GitLab) не поддерживают длинные имена образов, то при развертывании PT CS вам необходимо указать параметр `--short-image-name`.

См. также

[Мониторинг состояния PT CS \(см. раздел 13.5\)](#)

[О технической поддержке \(см. раздел 19\)](#)

5.2. Быстрая установка

Вы можете воспользоваться инструкцией по быстрой установке для тестирования PT CS. Для быстрой установки будет использован конфигурационный файл Helm-чарта с параметрами по умолчанию. В этом случае необходимые СУБД и сервисы будут установлены одновременно с PT CS. Если необходимо учитывать особенности имеющейся инфраструктуры, вы можете [заполнить конфигурационный файл Helm-чарта \(см. раздел 5.5\)](#) до начала установки.

► Чтобы установить PT CS:

1. Запустите [скрипт для публикации образов \(см. раздел 5.6\)](#) PT CS в реестре.
2. Выполните команду:

```
DATABASE_PASSWORD='<Пароль>' REDIS_PASSWORD='<Пароль>' RABBITMQ_PASSWORD='<Пароль>'
CLICKHOUSE_PASSWORD='<Пароль>' ADMINISTRATOR_PASSWORD='<Пароль>' ./generate_values.sh --
registry <Адрес сервера с реестром образов PT CS> -a 'ptcs' --storage-class '<Класс
хранилища данных PT CS>' --ingress-class <Имя класса Ingress> --own-ptcs-url <Адрес, по
которому будет доступен устанавливаемый экземпляр PT CS>
```

Например:

```
DATABASE_PASSWORD='P@ssw0rd' REDIS_PASSWORD='P@ssw0rd' RABBITMQ_PASSWORD='P@ssw0rd'
CLICKHOUSE_PASSWORD='P@ssw0rd' ADMINISTRATOR_PASSWORD='P@ssw0rd' ./generate_values.sh --
registry <registry.example.com> -a 'ptcs' --storage-class 'nfs-client' --ingress-class
nginx --own-ptcs-url https://ptcs.example.com
```

Теперь вы можете перейти [к первоначальной настройке](#) (см. раздел 6).

См. также

[Порядок первоначальной настройки PT CS](#) (см. раздел 6)

5.3. Установка из дистрибутива

Состав дистрибутива PT CS

Основные компоненты дистрибутива PT CS приведены в таблице ниже.

Таблица 9. Основные компоненты дистрибутива PT CS

Компонент	Описание
Каталог <code>bin</code>	Содержит служебные утилиты, в частности утилиту для генерации сертификатов и ключей для работы модулей PT CS и утилиту <code>skoreo</code> для переноса образов в другие реестры
Каталог <code>Documentation</code>	Содержит Руководство администратора, Руководство пользователя и Справочник по API
<code>generate_values.sh</code>	Скрипт для заполнения файла конфигурационного файла Helm-чарта (см. раздел 5.5)
<code>ptcs_images.tgz</code>	Архив с образами
<code>ptcs-<Номер версии PT CS>.tgz</code>	Архив с Helm-чартом. В архиве находятся следующие каталоги: — <code>charts</code> — с файлами для работы Helm; — <code>dashboards</code> — с дашбордами в формате JSON для загрузки в Grafana; — <code>templates</code> — с шаблонами конфигурационных файлов в формате YAML

Распаковка архива с дистрибутивом PT CS

Перед установкой PT CS вам необходимо распаковать архив с дистрибутивом.

► Чтобы распаковать архив с дистрибутивом PT CS:

1. Скопируйте архив с дистрибутивом PT CS в любой каталог.

2. Перейдите в каталог с архивом, выполнив команду:

```
cd <Имя каталога>
```

3. Распакуйте архив в один из каталогов, выполнив команду:

```
tar xvf ptcs-<Номер версии PT CS>.tar -C <Имя каталога>/
```

Например:

```
tar xvf ptcs-v0.7.0.tar -C ptcs-installer/
```

Архив с дистрибутивом распакован. Теперь вы можете перейти к установке PT CS.

Установка PT CS

Перед установкой рекомендуется ознакомиться [с инструкциями по работе с Helm-чартом \(см. раздел 5.5\)](#).

Вы можете настроить доступ к веб-интерфейсу PT CS с помощью проброса порта после установки. Если вы планируете использовать для доступа к веб-интерфейсу PT CS Ingress-контроллер или сервис Noderport, то вам нужно указать [параметры для развертывания доступа \(см. раздел 5.7\)](#) в команде установки PT CS. Кроме того, вы можете запустить команду установки PT CS, используя [дополнительные параметры \(см. раздел 5.5\)](#).

В команде установки вам нужно указать имя релиза, которое впоследствии нужно будет использовать [при обновлении \(см. раздел 15\)](#) PT CS. Вы можете указать произвольное имя релиза, но для упрощения дальнейшей работы рекомендуется использовать `ptcs`.

► Чтобы установить PT CS с помощью дистрибутива:

1. Перейдите в каталог с распакованным архивом дистрибутива PT CS, выполнив команду:

```
cd <Имя каталога с распакованным архивом>/
```

Например:

```
cd ptcs-installer/
```

2. Запустите [скрипт для публикации образов \(см. раздел 5.6\)](#) PT CS в реестре.

3. Если к центральному кластеру планируется подключать дочерние, запустите [скрипт для публикации Helm-чарта \(см. раздел 5.6\)](#) в реестре.

4. Выполните одно из следующих действий:

- Заполните конфигурационный файл Helm-чарта вручную.
- [Заполните конфигурационный файл Helm-чарта с помощью скрипта generate_values.sh \(см. раздел 5.5\)](#).

При успешном завершении работы скрипта `generate_values.sh` будет создан конфигурационный файл Helm-чарта (`values.yaml`), а на экран будет выведена команда для установки с заполненными параметрами, которую вы можете скопировать, а затем — выполнить.

5. Выполните одно из следующих действий:

- Если вы заполнили конфигурационный файл Helm-чарта вручную, выполните команду для установки нового пакета PT CS:

```
helm upgrade --install <Имя релиза> <Путь к архиву с Helm-чартом> --version <Номер версии> --namespace <Пространство имен> --create-namespace --values <Имя конфигурационного файла Helm-чарта>.yaml
```
- Если вы заполнили конфигурационный файл Helm-чарта с помощью скрипта `generate_values.sh`, скопируйте и выполните полученную команду установки.

Теперь вы можете перейти [к первоначальной настройке \(см. раздел 6\)](#).

См. также

[Обновление PT CS \(см. раздел 15\)](#)

[Аппаратные и программные требования \(см. раздел 5.1\)](#)

[Настройка подключения к веб-интерфейсу PT CS \(см. раздел 5.7\)](#)

5.4. Установка из онлайн-реестра

Получение логина и пароля из файла лицензии для подключения к серверу обновлений

Перед установкой PT CS вам необходимо получить логин и пароль из файла лицензии для подключения к серверу обновлений.

► Чтобы получить логин и пароль из файла лицензии для подключения к серверу обновлений:

1. В каталоге, из которого будет выполняться установка, сохраните лицензию PT CS в файл `license-access-token.key`.

Примечание. Файл `license-access-token.key` закодирован по стандарту Base64.

2. Распознайте закодированный файл `license-access-token.key`, выполнив команду:

```
cat license-access-token.key | cut -d '.' -f2 | base64 -d | jq .
```
3. Из файла `license-access-token.key` получите `InstanceAccessToken`, выполнив команду:

```
JWT=$(cat license-access-token.key | cut -d '.' -f2 | base64 -d | jq -r .instance_access_token_jwt)
```

Создана переменная `JWT`, которая будет использоваться при установке PT CS.

Установка PT CS из онлайн-реестра

Вы можете настроить доступ к веб-интерфейсу PT CS с помощью проброса порта после установки. Если вы планируете использовать для доступа к веб-интерфейсу PT CS Ingress-контроллер или сервис Noderport, то вам нужно указать [параметры для развертывания доступа \(см. раздел 5.7\)](#) в команде установки PT CS. Кроме того, вы можете запустить команду установки PT CS, используя [дополнительные параметры \(см. раздел 5.5\)](#).

В команде установки вам нужно указать имя релиза, которое впоследствии нужно будет использовать [при обновлении \(см. раздел 15\)](#) PT CS. Вы можете указать произвольное имя релиза, но для упрощения дальнейшей работы рекомендуется использовать `ptcs`.

► Чтобы установить PT CS:

1. Получите список доступных версий PT CS и их состав, выполнив команду:

```
curl -s https://update.ptsecurity.com/api/v7/products/PTCS/versions/ -H "Authorization: InstanceAccessToken $JWT" | jq .
```

Примечание. Номер версии PT CS отображается в значении параметра `helm.ptcs` в секции `properties`.

2. Получите логин пользователя для авторизации в реестре с Helm-чартом, выполнив команду:

```
LOGIN=$(echo $JWT | cut -d '.' -f 1-2)
```

3. Получите пароль для авторизации в реестре с Helm-чартом, выполнив команду:

```
PASSWORD=$(echo $JWT | cut -d '.' -f 3)
```

4. Выполните одно из следующих действий:

- Если используется Helm версии 3.18.0 и выше, авторизуйтесь в репозитории Helm, выполнив команду:

```
docker login update.ptsecurity.com/ptcs --username $LOGIN --password $PASSWORD
```

Примечание. Helm версии 3.18.0 и выше не поддерживают тип авторизации, используемый `update.ptsecurity.com`, поэтому выполнение команды `helm registry login` приведет к ошибке.

- Если используется Helm версии 3.17.3 и ниже, авторизуйтесь в репозитории Helm, выполнив команду:

```
helm registry login update.ptsecurity.com/ptcs --username $LOGIN --password $PASSWORD
```

5. Экспортируйте параметры Helm-чарта в конфигурационный файл <Название конфигурационного файла Helm-чарта>.yaml, выполнив команду:

```
helm show values oci://update.ptsecurity.com/ptcs/ptcs --version <Номер версии PT CS> > <Имя конфигурационного файла Helm-чарта>.yaml
```

Например:

```
helm show values oci://update.ptsecurity.com/ptcs/ptcs --version v0.0.0-pre.2 > values.yaml
```

Примечание. Для конфигурационного файла Helm-чарта рекомендуется использовать название `values`.

6. [Заполните конфигурационный файл Helm-чарта вручную \(см. раздел 5.5\).](#)

7. Выполните команду для установки нового пакета PT CS:

```
helm upgrade --install <Имя релиза> oci://update.ptsecurity.com/ptcs/ --version <Номер версии> --namespace <Пространство имен> --create-namespace --values <Название конфигурационного файла Helm-чарта>.yaml
```

Например:

```
helm upgrade --install ptcs oci://update.ptsecurity.com/ptcs/ptcs --version v0.0.0-pre.2 --namespace ptcs --create-namespace --values ./values.yaml
```

Примечание. Рекомендуется использовать имя релиза `ptcs`. Если вы не планируете использовать сгенерированные ранее значения параметров `encryption` и `publicAccessTokenSalt`, то в файле `values.yaml` вы можете задать для них значение `INIT-DO-NOT-USE`. В таком случае значения параметров будут сгенерированы при запуске команды.

Теперь вы можете перейти [к первоначальной настройке \(см. раздел 6\)](#).

См. также

[Обновление PT CS \(см. раздел 15\)](#)

[Аппаратные и программные требования \(см. раздел 5.1\)](#)

[Настройка подключения к веб-интерфейсу PT CS \(см. раздел 5.7\)](#)

[Работа с Helm-чартом \(см. раздел 5.5\)](#)

5.5. Работа с Helm-чартом

Параметры, с которыми будет установлен PT CS, указываются в конфигурационном файле Helm-чарта.

Для экземпляра PT CS, устанавливаемого в центральном кластере, вы можете заполнить конфигурационный файл вручную или с использованием скрипта `generate_values.sh`. Скрипт создан для упрощения создания конфигурационного файла Helm-чарта с самыми популярными параметрами, и его запуск необязателен.

Примечание. Конфигурационный файл Helm-чарта для установки PT CS в дочерних кластерах будет сгенерирован после указания параметров [для подключения дочерних кластеров \(см. раздел 13.3.1\)](#).

Названия и описания всех доступных для использования в конфигурационном файле параметров приведены в файлах `values.yaml` и `README.md` из архива `ptcs-<Номер версии PT CS>.tgz` установочного дистрибутива. Для конфигурационного файла рекомендуется использовать название `values.yaml`, но при необходимости вы можете использовать другое.

Ниже приведены инструкции по работе с конфигурационным файлом Helm-чарта, параметры, поддерживаемые скриптом `generate_values.sh` и параметры скрипта публикации Helm-чарта в реестр.

Просмотр документации Helm-чарта

- ▶ Чтобы просмотреть документацию Helm-чарта PT CS,

выполните команду:

```
helm show readme <Путь к архиву с Helm-чартом PT CS> --version <Версия Helm-чарта PT CS>
```

Генерация конфигурационного файла со всеми переменными

- ▶ Чтобы сгенерировать конфигурационный файл Helm-чарта со всеми переменными Helm-чарта,

выполните команду:

```
helm show values <Путь к архиву с Helm-чартом PT CS> --version <Версия Helm-чарта PT CS> > values.yaml
```

Заполнение конфигурационного файла вручную

- ▶ Чтобы заполнить конфигурационный файл Helm-чарта вручную:

1. Перейдите в каталог с распакованным архивом дистрибутива PT CS, выполнив команду:
`cd <Имя каталога с распакованным архивом>/`

Например:

```
cd ptcs-installer/
```

2. Заполните конфигурационный файл Helm-чарта вручную.

Примечание. Если вы используете название файла с параметрами Helm-чарта, отличное от `values.yaml`, то при выполнении следующих шагов вам нужно указывать это название вместо `values.yaml`.

3. Если для хранения образов PT CS, вы используете реестр образов GitLab, то добавьте в файл `values.yaml` секцию:

```
runtime-monitor:
  tetragon:
    image:
      repository: tetragon
```

4. Если для хранения образов PT CS, вы используете реестр образов GitLab и разворачиваете СУБД PostgreSQL, ClickHouse, Redis или брокер сообщений RabbitMQ из установочного дистрибутива PT CS, то добавьте в файл `values.yaml` секцию:

```
image:
  repository: <Название СУБД или брокера сообщений>
```

Например:

```
image:
  repository: postgres
image:
  repository: rabbitmq
```

```
image:
  repository: clickhouse-server
image:
  repository: redis
```

Заполнение конфигурационного файла с помощью скрипта

- Чтобы заполнить конфигурационный файл Helm-чарта с помощью скрипта `generate_values.sh`:

1. Перейдите в каталог с распакованным архивом дистрибутива PT CS, выполнив команду:

```
cd <Имя каталога с распакованным архивом>/
```

Например:

```
cd ptcs-installer/
```

2. Запустите скрипт `generate_values.sh` с обязательными параметрами `--registry`, `--administration-user` и `--own-cs-url`:

```
./generate_values.sh --registry <Доменное имя или IP-адрес реестра с образами PT CS> --
administrator-user <Логин пользователя> --own-cs-url <Адрес для доступа к устанавливаемому
экземпляру PT CS>
```

Примечание. Если в установочном скрипте вы не указываете параметр `--storage-class`, то будет выбран `StorageClass` кластера, заданный по умолчанию.

Примечание. Если вы подключаете реестр с аутентификацией и указали в команде `./generate_values.sh` параметр `--registry-user`, но не задали в переменной `IMAGE_REGISTRY_PASSWORD` пароль, то он будет запрошен в интерактивном режиме.

О сертификатах для защищенного взаимодействия между модулями PT CS

При разворачивании PT CS с помощью Helm сертификаты для безопасного взаимодействия между модулями генерируются автоматически. По умолчанию они сохраняются в конфигурационном файле Helm-чарта. Вы можете использовать отдельный файл для хранения сертификатов, указав его имя с помощью параметра `--secret-values` при запуске скрипта `generate_values.sh`.

Для использования заранее подготовленных сертификатов их нужно добавить напрямую в файл `values.yaml`. В сертификатах вам нужно указать Subject Alternative Names (SAN) для всех сервисов PT CS: `ptcs.example.com`, `localhost`, `admission-controller`, `admission-controller.ptcs.svc`, `admission-controller.ptcs.svc.cluster.local`, `auth-api`, `cluster-manager`, `container-registry-integrator`, `event-processor`, `history-api`, `notifier`, `policy-enforcer`, `ptcs-manager`, `reverse-proxy`, `reverse-proxy.ptcs.svc.cluster.local`, `runtime-monitor`, `vuln-scanner`, `grafana`, `grafana.ptcs.svc.cluster.local`.

Параметры скрипта для генерации конфигурационного файла Helm-чарта

В таблице ниже приведены допустимые параметры скрипта `generate_values.sh` для генерации конфигурационного файла Helm-чарта.

Таблица 10. Параметры скрипта для генерации конфигурационного файла Helm-чарта

Параметр	Обязательный	Описание	Значение по умолчанию
<code>--help</code>	Нет	Выводит на экран описание всех допустимых параметров установочного скрипта	—
<code>--administrator-user</code>	Да	Задаёт логин пользователя для подключения к веб-интерфейсу PT CS. Пароль вы можете задать через переменную <code>ADMINISTRATOR_PASSWORD</code> . Если пароль не указан через переменную, то он будет запрошен в процессе установки PT CS	—
<code>--admission-ignored-namespaces</code>	Нет	Задаёт пространства имен, действия в которых не будут проверяться через контроллер допуска (см. раздел 5.8.1) . Введенное значение перезаписывает предыдущее. Например, если ранее в список были добавлены пространства имен <code>kube-system</code> , <code>kube-node-lease</code> и вы хотите добавить <code>kube-public</code> , то вам нужно задать все три пространства имен. Если вы укажете только <code>kube-public</code> , то <code>kube-system</code> , <code>kube-node-lease</code> будут исключены из игнорируемых	—
<code>--admission-webhook-block-on-failure</code>	Нет	Переопределяет значение параметра <code>FailurePolicy</code> для ресурса <code>ValidatingWebhookConfiguration</code> . Значение параметра <code>FailurePolicy</code> изменится на <code>fail</code> , если задано значение <code>true</code> , или на <code>ignore</code> , если задано значение <code>false</code>	<code>false</code>

Параметр	Обязательный	Описание	Значение по умолчанию
<code>--storage-class</code>	Нет	Задаёт название существующего класса хранилища (StorageClass), который будет использоваться для хранения данных PT CS. Вы можете переопределить класс хранилища для конкретного сервиса с помощью параметров <code>--database-storage-class</code>, <code>--redis-storage-class</code>, <code>--clickhouse-storage-class</code>, <code>--rabbitmq-storage-class</code>, <code>--loki-storage-class</code>	—
<code>--ca</code>	Нет	Задаёт путь к корневому сертификату, который будет использоваться для проверки подлинности сторон при установке TLS(SSL)-соединения	—
<code>--cert-path</code>	Нет	Задаёт путь к сертификату, который будет использоваться для установки безопасного соединения между сервисами PT CS	—
<code>--cert-path-key</code>	Нет	Задаёт путь к закрытому ключу, связанному с сертификатом, указанным с помощью параметра <code>--cert-path</code>	—
<code>--central-cs-url</code>	Нет	Адрес центрального кластера PT CS в формате <Протокол>://<Адрес>[:<Порт>]. В качестве протокола вы можете указать значение <code>http</code> или <code>https</code>. Если этот параметр не задан, будет использовано значение параметра <code>--own-cs-url</code>	—
<code>--central-cs-disable-tls-check-cert</code>	Нет	Позволяет отключить проверку сертификатов для центрального кластера PT CS при использовании соединения по протоколу TLS	—
<code>--clickhouse</code>	Нет	Задаёт адрес для подключения к внешней СУБД ClickHouse (СУБД, развернутой не из дистрибутива PT CS) в формате <Адрес>:<Порт>.	—

Параметр	Обязательный	Описание	Значение по умолчанию
		Если параметр не передан в процессе установки PT CS, то СУБД ClickHouse будет развернута из дистрибутива PT CS в том же пространстве имен	
<code>--clickhouse-storage-class</code>	Нет	Задаёт название существующего класса хранения данных (StorageClass), который будет использоваться для хранения данных СУБД ClickHouse, установленной из дистрибутива PT CS Вы можете использовать этот параметр, если не задан параметр <code>--clickhouse</code>	—
<code>--clickhouse-database-name</code>	Нет	Имя БД ClickHouse в центральном кластере. Примечание. Рекомендуется использовать значение <code>ptcs</code>	—
<code>--clickhouse-user</code>	Нет	Позволяет указать логин пользователя для подключения к внешней СУБД ClickHouse в формате <code><Адрес> : <Порт></code> . Пароль для подключения вы можете указать через переменную <code>CLICKHOUSE_PASSWORD</code> . Если пароль не указан через переменную, то он будет запрошен в процессе установки PT CS	<code>ptcs</code>
<code>--clickhouse-disable-tls</code>	Нет	Позволяет не использовать протокол TLS при подключении к внешней СУБД ClickHouse	—
<code>--clickhouse-disable-tls-check-cert</code>	Нет	Если используется внешняя СУБД ClickHouse и использование TLS-соединения не отключено, но не задан путь к сертификату или сертификат не добавлен в доверенные на уровне инфраструктуры, то подключиться к СУБД ClickHouse будет невозможно. Этот параметр позволяет отключить проверку сертификата. Вы можете не	—

Параметр	Обязательный	Описание	Значение по умолчанию
		отключать проверку сертификата, а задать путь к сертификату с помощью параметра <code>--clickhouse-ca-path</code>	
<code>--clickhouse-ca-path</code>	Нет	Задаёт путь к сертификату, используемому при подключении к внешней СУБД ClickHouse. Сертификат будет добавлен в доверенные для сервисов PT CS. Если этот параметр не задан и используется внешняя СУБД ClickHouse, то невозможно будет проверить достоверность сертификата ClickHouse, следовательно подключения к СУБД Redis не будет и PT CS работать не сможет. Рекомендуется задать путь к сертификату с помощью параметра <code>--clickhouse-ca-path</code> или отключить проверку сертификата, указав параметр <code>--clickhouse-disable-tls-check-cert</code>	—
<code>--clickhouse-ephemeral</code>	Нет	Отключает персистентность для СУБД ClickHouse, развернутой из дистрибутива PT CS. Вы можете использовать этот параметр, если не задан параметр <code>--clickhouse</code>	—
<code>--database</code>	Нет	Задаёт адрес для подключения к внешней СУБД PostgreSQL (СУБД, развернутой не из дистрибутива PT CS) в формате <code><Адрес> : <Порт></code>. Если параметр не передан в процессе установки PT CS, то СУБД PostgreSQL будет развернута из дистрибутива PT CS	—
<code>--database-storage-class</code>	Нет	Задаёт название существующего класса хранения данных (StorageClass), который будет ис-	—

Параметр	Обязательный	Описание	Значение по умолчанию
		пользоваться для хранения данных СУБД PostgreSQL, установленной из дистрибутива PT CS Вы можете использовать этот параметр, если не задан параметр <code>--database</code>	
<code>--database-user</code>	Нет	Позволяет указать логин пользователя для подключения к внешней СУБД PostgreSQL в формате <code><Адрес> : <Порт></code>. Пароль для подключения вы можете указать через переменную <code>DATABASE_PASSWORD</code>. Если пароль не указан через переменную, то он будет запрошен в процессе установки PT CS	<code>ptcs</code>
<code>--database-name</code>	Нет	Имя БД PostgreSQL в центральном кластере. Примечание. Рекомендуется использовать значение <code>ptcs</code>	—
<code>--database-ca-path</code>	Нет	Позволяет задать путь к сертификату, используемому при подключении к внешней СУБД PostgreSQL. Сертификат будет добавлен в доверенные для сервисов PT CS. Если этот параметр не задан и используется внешняя СУБД PostgreSQL, то невозможно будет проверить достоверность сертификата PostgreSQL, следовательно подключения к СУБД PostgreSQL не будет и PT CS работать не сможет. Рекомендуется задать путь к сертификату с помощью параметра <code>--database-ca-path</code> или отключить проверку сертификата, указав параметр <code>--database-disable-tls-check-cert</code>	—
<code>--database-disable-tls</code>	Нет	Позволяет не использовать протокол TLS при подключении к внешней СУБД PostgreSQL	—

Параметр	Обязательный	Описание	Значение по умолчанию
<code>--database-disable-tls-check-cert</code>	Нет	Если используется внешняя СУБД PostgreSQL и использование TLS-соединения не отключено, но не задан путь к сертификату или сертификат не добавлен в доверенные на уровне инфраструктуры, то подключиться к СУБД PostgreSQL будет невозможно. Этот параметр позволяет отключить проверку сертификата. Вы можете не отключать проверку сертификата, а задать путь к сертификату с помощью параметра <code>--database-ca-path</code>	—
<code>--database-ephemeral</code>	Нет	Отключает персистентность для СУБД PostgreSQL, развернутой из дистрибутива PT CS. Вы можете использовать этот параметр, если не задан параметр <code>--database</code>	—
<code>--deploy-ingress</code>	Нет	Развертывает манифест Ingress по умолчанию для доступа к веб-версии PT CS	1
<code>--deploy-nodeport</code>	Нет	Позволяет развернуть сервис NodePort для доступа к веб-интерфейсу PT CS	false
<code>--nodeport</code>	Нет	Задаёт порт для подключения по протоколу HTTP или gRPC. Возможные значения: от 30000 до 32767	—
<code>--dev</code>	Нет	Устанавливает PT CS в режиме разработки (в одиночных репликах и с отладочным режимом журналирования)	0
<code>--enable-logger</code>	Нет	Включает сохранение событий PT CS в системе Grafana Loki, устанавливаемой при развертывании PT CS. Возможные значения: <code>true</code> (включить) и <code>false</code> (отключить)	—
<code>--grafana</code>	Нет	Задаёт адрес для подключения к серверу с внешней системой Grafana (системой, развернутой не из дистрибутива PT CS) <Протокол>://	—

Параметр	Обязательный	Описание	Значение по умолчанию
		<Адрес> : <Порт>. В качестве протокола вы можете указать значение <code>http</code> или <code>https</code>	
<code>--grafana-user</code>	Нет	Позволяет указать логин пользователя для подключения к серверу с Grafana	—
<code>--enable-monitoring</code>	Нет	Включает сбор метрик для сервисов PT CS с помощью Prometheus. Возможные значения: <code>true</code> (включить) и <code>false</code> (отключить)	—
<code>--external-prometheus</code>	Нет	Отключает развертывание Prometheus из дистрибутива PT CS. Возможные значения: <code>true</code> (отключить) и <code>false</code> (включить)	—
<code>--history-retention-interval</code>	Нет	Задаёт длительность хранения истории, указывается в часах	8760h
<code>--ingress-class</code>	Нет	Задаёт класс установленного Ingress-контроллера для доступа к веб-интерфейсу PT CS	—
<code>--ingress-hostname</code>	Нет	Задаёт имя хоста для Ingress-контроллера	—
<code>--ingress-cert</code>	Нет	Задаёт путь к сертификату для подключения к веб-интерфейсу PT CS через Ingress-контроллер и тем самым обеспечивает возможность использовать уже имеющийся в инфраструктуре организации сертификат. Примечание. Значения параметра <code>Common Name (CN)</code> для корневого и используемого сертификатов должны различаться. Параметр может быть использован только при настройке доступа к веб-интерфейсу через Ingress-контроллер	—
<code>--ingress-cert-key</code>	Нет	Задаёт путь к закрытому ключу сертификата для подключения к веб-интерфейсу PT CS	—

Параметр	Обязательный	Описание	Значение по умолчанию
<code>--loki</code>	Нет	Задаёт адрес для подключения к Grafana Loki в формате <code><Адрес> : <Порт></code>	—
<code>--loki-ephemeral</code>	Нет	Отключает персистентность для Grafana Loki	—
<code>--loki-storage-class</code>	Нет	Задаёт класс хранилища для данных Grafana Loki. Если не задан, то будет использоваться класс хранилища, указанный в значении параметра <code>--storage-class</code>	—
<code>--loki-disable-tls</code>	Нет	Позволяет отключить использование протокола TLS при подключении к сервису Grafana Loki. Возможные значения <code>true</code> (отключить) и <code>false</code> (включить)	—
<code>--loki-disable-tls-check-cert</code>	Нет	Если использование TLS-соединения с Grafana Loki не отключено, но не задан путь к сертификату или сертификат не добавлен в доверенные на уровне инфраструктуры, то подключиться к Grafana Loki будет невозможно. Этот параметр позволяет отключить проверку сертификата. Вы можете не отключать проверку сертификата, а задать путь к сертификату с помощью параметра <code>--loki-ca-path</code>	—
<code>--loki-ca-path</code>	Нет	Позволяет задать путь к сертификату, используемому при подключении к сервису Grafana Loki. Сертификат будет добавлен в доверенные для сервисов PT CS. Если этот параметр не задан, то невозможно будет проверить достоверность сертификата Grafana Loki, следовательно, подключения к Grafana Loki не будет.	—

Параметр	Обязательный	Описание	Значение по умолчанию
		Рекомендуется задать путь к сертификату с помощью параметра <code>--loki-ca-path</code> или отключить проверку сертификата, указав параметр <code>--loki-disable-tls-check-cert</code>	
<code>--loki-tenant-id</code>	Нет	Идентификатор (tenant ID) для мультитенантного доступа к GrafanaLoki	—
<code>--own-cs-url</code>	Да	Задаёт адрес, по которому будет доступен экземпляр PT CS, установленный в дочернем кластере в формате <code><Протокол>://<Адрес>[:<Порт>]</code> . В качестве протокола вы можете указать значение <code>http</code> или <code>https</code>	—
<code>--prometheus-ephemeral</code>	Нет	Отключает персистентность для сервиса Prometheus, развернутого из дистрибутива PT CS. Возможные значения: <code>true</code> (включить) и <code>false</code> (отключить). Вы можете использовать этот параметр, если не задан параметр <code>--external-prometheus</code>	—
<code>--prometheus-storage-class</code>	Нет	Задаёт название существующего класса хранения данных (StorageClass), который будет использоваться для хранения данных сервиса Prometheus, установленного из дистрибутива PT CS. Вы можете использовать этот параметр, если не задан параметр <code>--external-prometheus</code>	—
<code>--rabbitmq</code>	Нет	Задаёт адрес для подключения к внешнему брокеру сообщений RabbitMQ (брокеру, развернутому не из дистрибутива PT CS) в формате <code><Адрес>:<Порт></code> . Если параметр не передан в процессе установки PT CS, то брокер сообщений RabbitMQ будет развернут из дистрибутива PT CS в том же пространстве имен	—

Параметр	Обязательный	Описание	Значение по умолчанию
<code>--rabbitmq-storage-class</code>	Нет	Задаёт название существующего класса хранения данных (StorageClass), который будет использоваться для хранения данных брокера сообщений RabbitMQ, установленного из дистрибутива PT CS Вы можете использовать этот параметр, если не задан параметр <code>--rabbitmq</code>	—
<code>--rabbitmq-user</code>	Нет	Позволяет указать логин пользователя для подключения к внешнему брокеру сообщений RabbitMQ в формате <Адрес> : <Порт>. Пароль для подключения вы можете указать через переменную <code>RABBITMQ_PASSWORD</code>. Если пароль не указан через переменную, то он будет запрошен в процессе установки PT CS	<code>ptcs</code>
<code>--rabbitmq-ephemeral</code>	Нет	Отключает персистентность для менеджера очередей RabbitMQ, развернутого из дистрибутива PT CS. Возможные значения <code>true</code> (включить) и <code>false</code> (отключить). Вы можете использовать этот параметр, если не задан параметр <code>--rabbitmq</code>	—
<code>--redis</code>	Нет	Задаёт адрес для подключения к внешней СУБД Redis (СУБД, развернутой не из дистрибутива PT CS) в формате <Адрес> : <Порт>. Если параметр не передан в процессе установки PT CS, то СУБД Redis будет развернута из дистрибутива PT CS в том же пространстве имен	—
<code>--redis-storage-class</code>	Нет	Задаёт название существующего класса хранения данных (StorageClass), который будет использоваться для хранения данных СУБД Redis, установленной из дистрибутива PT CS	—

Параметр	Обязательный	Описание	Значение по умолчанию
		Вы можете использовать этот параметр, если не задан параметр <code>--redis</code>	
<code>--redis-user</code>	Нет	Позволяет указать логин пользователя для подключения к внешней СУБД Redis в формате <Адрес>: <Порт>. Пароль для подключения вы можете указать через переменную <code>REDIS_PASSWORD</code> . Если пароль не указан через переменную, то он будет запрошен в процессе установки PT CS	<code>ptcs</code>
<code>--redis-disable-tls</code>	Нет	Позволяет отключить использование протокола TLS при подключении к внешней СУБД Redis. Возможные значения <code>true</code> (отключить) и <code>false</code> (включить)	—
<code>--redis-disable-tls-check-cert</code>	Нет	Если используется внешняя СУБД Redis и использование TLS-соединения не отключено, но не задан путь к сертификату или сертификат не добавлен в доверенные на уровне инфраструктуры, то подключиться к СУБД Redis будет невозможно. Этот параметр позволяет отключить проверку сертификата. Вы можете не отключать проверку сертификата, а задать путь к сертификату с помощью параметра <code>--redis-ca-path</code>	—
<code>--redis-ca-path</code>	Нет	Позволяет задать путь к сертификату, используемому при подключении к внешней СУБД Redis. Сертификат будет добавлен в доверенные для сервисов PT CS. Если этот параметр не задан и используется внешняя СУБД Redis, то невозможно будет проверить достоверность сертификата Redis, следовательно подключения к СУБД Redis не будет и PT CS работать не сможет.	—

Параметр	Обязательный	Описание	Значение по умолчанию
		Рекомендуется задать путь к сертификату с помощью параметра <code>--redis-ca-path</code> или отключить проверку сертификата, указав параметр <code>--redis-disable-tls-check-cert</code>	
<code>--redis-persistent</code>	Нет	Включает персистентность для СУБД Redis, развернутой из дистрибутива PT CS. Возможные значения <code>true</code> (включить) и <code>false</code> (отключить). Вы можете использовать этот параметр, если не задан параметр <code>--redis</code>	—
<code>--registry</code>	Да	Задаёт доменное имя сервера с реестром образов PT CS или путь к серверу в формате <code><Адрес> : <Порт></code>	—
<code>--registry-user</code>	Нет	Задаёт логин пользователя для подключения к реестру образов PT CS. Пароль вы можете задать через переменную <code>IMAGE_REGISTRY_PASSWORD</code> . Если пароль не указан через переменную, то он будет запрошен в процессе установки PT CS	—
<code>--proxy-url</code>	Нет	Задаёт адрес прокси-сервера для работы компонентов PT CS в формате <code><Протокол> : // [<Логин> : <Пароль> @] <Адрес> : [<Порт>]</code> . В качестве протокола вы можете указать значение <code>http</code> или <code>https</code> . Примечание. Использование прокси-сервера позволяет получить доступ к серверу обновлений, базам данных угроз и уязвимостей, активировать лицензию (см. раздел 4) PT CS и реализовать отправку уведомлений с помощью механизма webhook (см. раздел 11) при установке PT CS в изолированном от интернета сегменте сети	—

Параметр	Обязательный	Описание	Значение по умолчанию
<code>--secret-values</code>	Нет	Включает сохранение генерируемых сертификатов для взаимодействия сервисов PT CS в отдельном файле	—
<code>--short-image-names</code>	Нет	Позволяет использовать сокращенные имена образов. Возможные значения <code>true</code> (включить) и <code>false</code> (отключить)	—
<code>--syslog</code>	Нет	Задаёт адрес syslog-сервера, на который будут отправляться события системы PT CS, в формате <code><Протокол>://<Адрес>[:<Порт>]</code> . В качестве протокола вы можете указать значение <code>udp</code> или <code>tcp</code> . Обрабатывается, если указан параметр <code>--enable-logger</code>	—
<code>--values</code>	Нет	Задаёт имя файла, в который будут записаны значения для конфигурирования Helm-чарта	<code>values.yaml</code>

См. также

[Лицензирование \(см. раздел 4\)](#)

[Управление сервисами и шаблонами уведомлений \(см. раздел 11\)](#)

5.6. Публикация Helm-чарта и образов PT CS в реестре

Если вы планируете устанавливать PT CS в дочерних кластерах и не хотите размещать Helm-чарт в каждом кластере отдельно, вы можете опубликовать его в реестре с помощью скрипта `push_chart.sh`. Вы также можете опубликовать образы PT CS с помощью скрипта `push_images.sh`.

Публикация Helm-чарта

► Чтобы опубликовать Helm-чарт в реестре,

запустите скрипт:

```
./push_chart.sh --registry <Путь к реестру, в котором будет опубликован Helm-чарт> -u <Логин для доступа к реестру>
```

Примечание. Вы можете задать пароль для доступа к реестру, указав переменную `IMAGE_REGISTRY_PASSWORD`.

Например:

```
IMAGE_REGISTRY_PASSWORD='<Пароль>' ./push_chart.sh <Путь к реестру, в котором будет опубликован Helm-чарт> -u <Логин для доступа к реестру>
```

Параметры скрипта публикации Helm-чарта в реестре `push_chart.sh` приведены в таблице ниже.

Таблица 11. Параметры скрипта `push_chart.sh`

Параметр	Обязательный	Описание	Значение по умолчанию
<code>-h, --help</code>	Нет	Выводит на экран описание всех допустимых параметров скрипта	—
<code>-r, --registry</code>	Да	Указывает доменное имя сервера с реестром с Helm-чартом или путь к серверу в формате <code><Адрес>: <Порт></code>	—
<code>-u, --registry-user</code>	Нет	Указывает имя пользователя для подключения к реестру. Пароль вы можете задать через переменную <code>IMAGE_REGISTRY_PASSWORD</code> . Если пароль не указан через переменную, то он будет запрошен в процессе работы скрипта <code>push_chart.sh</code>	—
<code>-i, --registry-insecure</code>	Нет	Отключает проверку сертификатов при использовании безопасного соединения. Может быть использован, если Helm-чарт нужно опубликовать в реестре, сертификат которого не является доверенным	—
<code>--registry-plain-http</code>	Нет	Позволяет опубликовать Helm-чарт в реестре, который не поддерживает соединение по протоколу HTTPS	—

Публикация образов

- Чтобы опубликовать образы PT CS в реестре,

запустите скрипт:

```
./push_images.sh --registry <Доменное имя или IP-адрес реестра с образами PT CS> --registry-user <Логин для доступа к реестру>
```

Примечание. Если для хранения образов PT CS используется реестр GitLab, то при запуске скрипта вам нужно указать параметр `--short-image-names`.

Примечание. Вы можете изменить число потоков загрузки образов, указав параметр `-P`. Число потоков по умолчанию — 4.

В таблице ниже приведены допустимые параметры скрипта для публикации образов в реестре `push_images.sh`.

Таблица 12. Параметры скрипта `push_images.sh`

Параметр	Обязательный	Описание	Значение по умолчанию
<code>-h, --help</code>	Нет	Выводит на экран описание всех допустимых параметров скрипта	Не используется
<code>-r, --registry</code>	Да	Задаёт доменное имя сервера с реестром с образами PT CS или путь к серверу в формате <code><Адрес> : <Порт></code>	—
<code>-u, --registry-user</code>	Нет	Задаёт логин для подключения к реестру с образами PT CS. Пароль можно задать через переменную <code>IMAGE_REGISTRY_PASSWORD</code> или ввести во время выполнения установочного скрипта. Примечание. Чтобы избежать экранирования символов, используйте одинарные кавычки для ввода пароля	—
<code>-i, --registry-insecure-login</code>	Нет	Отключает проверку сертификата для защищенного соединения по протоколу TLS	—
<code>--short-image-names</code>	Нет	Если указан этот параметр, то будет использоваться сокращенное имя образа. Напри-	—

Параметр	Обязательный	Описание	Значение по умолчанию
		мер, <code>redis:7.0.1</code> вместо <code>docker.io/bitnami/redis:7.0.1</code> . Можно использовать для реестров, не поддерживающих вложенные каталоги	
<code>-P, --parallel</code>	Нет	Количество потоков, отправляющих образы в реестр	4

5.7. Настройка подключения к веб-интерфейсу PT CS

Вы можете настроить доступ к веб-интерфейсу PT CS с помощью следующих способов:

- с помощью проброса порта;
- с помощью Ingress-контроллера. PT CS поддерживает два типа Ingress-контроллеров: `nginx` и `HAProxy`. Для использования других Ingress-контроллеров потребуется ручная настройка дополнительных параметров;
- с помощью сервиса `NodePort`.

Вы можете настроить доступ к веб-интерфейсу с помощью проброса порта после установки PT CS. Если вы планируете использовать для доступа к веб-интерфейсу Ingress-контроллер или `Nodeport`, то вам необходимо указать параметры подключения при установке PT CS.

Настройка доступа к веб-интерфейсу с помощью проброса порта

- Чтобы настроить доступ к веб-интерфейсу с помощью проброса порта,

выполните команду:

```
kubectl -n cs port-forward svc/reverse-proxy 9000:9000
```

Отобразится информация о порте, через который можно получить доступ к веб-интерфейсу PT CS.

Настройка доступа к веб-интерфейсу с помощью Ingress-контроллера

► Чтобы настроить доступ к веб-интерфейсу с помощью Ingress-контроллера:

1. Предварительно установите Ingress-контроллер.
2. При запуске скрипта для генерации конфигурационного файла Helm-чарта укажите имя установленного Ingress-контроллера [в значении параметра \(см. раздел 5.5\)](#) `--ingress-class`.
3. Если требуется, укажите параметры Ingress. Для этого укажите значение `0` [для параметра \(см. раздел 5.5\)](#) `--deploy-ingress` и создайте объект с типом Ingress с собственными аннотациями.

Примечание. Для настройки доступа к веб-интерфейсу с помощью Ingress-контроллера вы можете использовать сертификат, имеющийся в инфраструктуре, указав параметры `--ingress-cert` и `--ingress-cert-key`.

Пример манифеста Ingress с параметрами для nginx:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ptcs-reverse-proxy-ingress
  labels:
    app.kubernetes.io/name: reverse-proxy
    app.kubernetes.io/instance: ptcs
  annotations:
    # nginx default configs
    # https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/
    annotations/
    nginx.ingress.kubernetes.io/proxy-body-size: "100m"
    nginx.ingress.kubernetes.io/proxy-connect-timeout: "1800"
    nginx.ingress.kubernetes.io/proxy-read-timeout: "1800"
    nginx.ingress.kubernetes.io/proxy-send-timeout: "1800"
spec:
  rules:
    - http:
        paths:
          - pathType: Prefix
            path: "/"
            backend:
              service:
                name: reverse-proxy
                port:
                  number: 9000
```

Примечание. В примере выше приведены пограничные значения размера тела запроса и тайм-аута проксируемых соединений.

Настройка доступа к веб-интерфейсу с помощью сервиса NodePort

- ▶ Чтобы настроить доступ к веб-интерфейсу с помощью сервиса NodePort:
 1. В процессе установки PT CS укажите `--deploy-nodeport` для сервиса NodePort.
 2. Если требуется, укажите конкретный порт для доступа к сервису NodePort, задав параметр `--nodeport`.

5.8. Настройка проверки и мониторинга

В этом разделе описана настройка проверки через контроллер допуска и настройка мониторинга событий среды выполнения.

В этом разделе

[Настройка проверки через контроллер допуска \(см. раздел 5.8.1\)](#)

[Настройка мониторинга событий среды выполнения \(см. раздел 5.8.2\)](#)

5.8.1. Настройка проверки через контроллер допуска

Отключение проверки действий в отдельных пространствах имен

Вы можете настроить PT CS таким образом, чтобы действия, выполняемые в отдельных пространствах имен, не проверялись, добавив эти пространства имен в игнорируемые. В список игнорируемых рекомендуется добавить системные пространства имен `kube-system`, `kube-node-lease`, `kube-public`. Пространство имен, в котором разворачивается PT CS (как правило — `ptcs`), добавлено в игнорируемые по умолчанию, и его нельзя исключить.

- ▶ Чтобы отключить проверку действий в отдельных пространствах имен, выполните одно из следующих действий:
 - Если PT CS не установлен, запустите установочный скрипт `generate_values.sh`, задав в значении параметра `--admission-ignored-namespaces` список игнорируемых пространств имен.

Внимание! Введенное значение перезаписывает предыдущее. Например, если ранее в список были добавлены пространства имен `kube-system`, `kube-node-lease` и вы хотите добавить `kube-public`, то вам нужно задать все три пространства имен. Если вы укажете только `kube-public`, то `kube-system`, `kube-node-lease` будут исключены из игнорируемых.

- Если PT CS уже развернут в кластере, добавьте пространство имен в секцию `namespaceSelector.matchExpression.values` объекта `ValidatingWebhookConfiguration` с именем `admission-controller`.

Примечание. При установке PT CS на платформу Deckhouse Kubernetes Platform вы можете получить список пространств имен, которые нужно добавить в исключения, выполнив команду - `kubectl get ns | egrep 'd8-*|kube'`.

Отключение блокировки действий

Вы можете настроить PT CS таким образом, чтобы действия при проверке через контроллер допуска не блокировались.

- ▶ Чтобы отключить блокировку действий,

выполните одно из следующих действий:

- Если PT CS не установлен, во время установки задайте параметру `admission-controller.webhookBlockOnFailure` значение `false`.
- Если PT CS уже развернут в кластере, задайте параметру `WEBHOOK_BLOCK_ON_FAILURE` значение `false`, отредактировав конфигурационную карту (ConfigMap) `admission-controller-config`.

Примечание. Подробную информацию о конфигурационной карте (ConfigMap) вы можете получить [в документации Kubernetes](#).

Настройка сохранения событий

Вы можете настроить, какие из событий, генерируемых при проверке через контроллер допуска, сохранять и отображать на вкладке **События** страницы **Контроллер допуска**.

При установке PT CS [в файле values.yaml \(см. раздел 5.5\)](#) вы можете задать параметру `admission-controller.historyControl` одно из значений:

- `All` — событие сохранится вне зависимости от обнаружения угроз и уязвимостей или действительности подписи.
- `Vulns` — событие сохранится только в том случае, если в нем будет обнаружена хотя бы одна уязвимость.
- `Threats` — событие сохранится только в том случае, если в нем будет найдена хотя бы одна угроза.
- `ImageSignatures` — событие сохранится только в том случае, если подпись образа отсутствует или недействительна.
- Значение, отличающееся от перечисленных выше, будет распознано как `None` — не сохранять события.

- Чтобы настроить сохранение событий,

выполните одно из следующих действий:

- Если PT CS не установлен, то при установке в файле `values.yaml` задайте параметру `admission-controller.historyControl` необходимое значение.

Например:

```
admission-controller:
  historyControl: "None"
```

- Если PT CS уже развернут в кластере, то задайте параметру `HISTORY_CONTROL` необходимое значение, отредактировав конфигурационную карту (ConfigMap) `admission-controller-config`.

Примечание. Подробную информацию о конфигурационной карте (ConfigMap) вы можете прочесть [в документации Kubernetes](#).

- Если вы хотите изменить параметры сохранения событий с помощью переменной окружения `HISTORY_CONTROL`, выполните команду:

```
kubectl set env -n <Пространство имен> deploy admission-controller
HISTORY_CONTROL=<Значение>
```

Например:

```
kubectl set env -n ptcs deploy admission-controller HISTORY_CONTROL=WithThreats
```

5.8.2. Настройка мониторинга событий среды выполнения

Для настройки параметров модулей PT CS используются манифесты. Манифест — это конфигурационный файл в формате YAML.

Ограничение сообщений в очереди RabbitMQ

Большое количество сообщений в очереди RabbitMQ может привести к значительному увеличению потребления памяти. Вы можете настроить количество отправляемых сообщений с помощью параметров `RABBIT_RUNTIME_EVENTS_QUEUE_PREFETCH_COUNT` в манифесте компонента Event processor и `RABBIT_QUEUE_PREFETCH_COUNT` в манифесте компонента History api. Значения параметров по умолчанию — 100.

- Чтобы настроить количество отправляемых сообщений:

1. Откройте в режиме редактирования манифест компонента Event processor.
2. Задайте значение параметра `RABBIT_RUNTIME_EVENTS_QUEUE_PREFETCH_COUNT`.
3. Сохраните изменения и выйдите из режима редактирования.
4. Откройте в режиме редактирования манифест компонента History api.
5. Задайте значение параметра `RABBIT_QUEUE_PREFETCH_COUNT`.

6. Сохраните изменения и выйдите из режима редактирования.
7. Последовательно перезапустите компоненты Event processor и History api, выполнив для каждого из них команду:


```
kubectl rollout restart deployment/<Имя сервиса модуля> -n <Пространство имен, в котором установлен PT CS>
```

Настройка размера буфера компонента Runtime monitor

Использование буфера позволяет продолжать обрабатывать события среды выполнения при недоступности компонента RabbitMQ. Вы можете настроить размер буфера компонента Runtime monitor, указав параметр `TETRAGON_EVENTS_BUFFER` в его манифесте и параметр `CONFIG_UPDATE_INTERVAL` в его конфигурационной карте (ConfigMap).

- Чтобы настроить размер буфера компонента Runtime monitor:
 1. Откройте в режиме редактирования манифест компонента Runtime monitor.
 2. Задайте значение параметра `TETRAGON_EVENTS_BUFFER`.
 3. Сохраните изменения и выйдите из режима редактирования.
 4. Откройте в режиме редактирования конфигурационную карту (ConfigMap) `runtime-monitor-config`.
 5. Задайте значение параметра `CONFIG_UPDATE_INTERVAL`.
 6. Перезапустите компонент Runtime monitor, выполнив команду:


```
kubectl rollout restart daemonset/<Имя сервиса> -n <Пространство имен, в котором установлен PT CS>
```

Смена сетевых портов для компонента Runtime monitor

Компонент Runtime monitor работает в режиме `hostNetwork`, то есть использует сетевое пространство имен хоста. Если порт на хосте уже занят, то сервис не будет отслеживать события среды выполнения. Вы можете задать порты для работы сервиса в конфигурационном файле Helm-чарта `values.yaml` перед установкой PT CS. Вы также можете задать порт для работы контейнера `tetragon` компонента Runtime monitor, если порт по умолчанию на хосте уже занят.

- Чтобы задать сетевой порт для компонента Runtime monitor:

1. Откройте в режиме редактирования файл `values.yaml`.
2. Задайте значения портов в секции `runtime monitor`.

Например:

```
runtime-monitor:
  containerPorts:
    http: 9000
    grpc: 8000
    gops: 7000
```

3. Если требуется, задайте порт для работы контейнера tetragon.

Например:

```
runtime-monitor:
  containerPorts:
    http: 9000
    grpc: 8000
    gops: 7000
  tetragon:
    grpc:
      address: "localhost:54321"
```

4. Сохраните изменения и выйдите из режима редактирования.

```
kubectl rollout restart daemonset/Имя сервиса -n <Пространство имен, в котором
установлен PT CS>
```

5.9. Развертывание в закрытом контуре

Вы можете развернуть PT CS в закрытом контуре без доступа к интернету с использованием локального сервера обновлений.

- Чтобы развернуть PT CS с использованием локального сервера обновлений:

1. [Установите локальный сервер обновлений \(см. раздел 16.2\).](#)
2. Откройте на сервере HTTP-порт, используемый на локальном сервере обновлений.

Примечание. Как правило, используется порт 8553. Но [он может быть изменен \(см. раздел 16.8\)](#) при установке и настройке локального сервера обновлений.

3. [Активируйте лицензию \(см. раздел 16.3\)](#) локального сервера обновлений.
4. Установите PT CS.

Примечание. При запуске скрипта для генерации конфигурационного файла Helm-чарта `generate_values.sh` установите переменную окружения `FLUS_URL='<Адрес локального сервера обновлений>:<Порт>'`.

```
FLUS_URL=<Адрес локального сервера обновлений>:<Порт> ./generate_values.sh --registry
<Доменное имя или IP-адрес реестра с образами PT CS> --administrator-user <Логин
пользователя> --own-ptcs-url <Адрес для доступа к устанавливаемому экземпляру PT CS>
```

5. [Активируйте лицензию \(см. раздел 4\)](#) PT CS в веб-интерфейсе.

6. Порядок первоначальной настройки PT CS

Возможности по настройке зависят от роли пользователя.

После установки PT CS и активации лицензии вам нужно настроить объекты в системе для получения и обработки информации об уязвимостях, найденных в ходе автоматического или ручного сканирования, в следующем порядке:

1. [Настроить реестры \(см. раздел 10.1\)](#) для подключения к PT CS.
2. Если необходимо, [подключить реестры \(см. раздел 10\)](#) для просмотра информации об образах и запуска ручного сканирования.
3. Подключить [сервисы уведомлений \(см. раздел 11\)](#) и настроить для них шаблоны.
4. Создать правила реагирования для определения автоматических реакций при обнаружении угроз, уязвимостей, ВПО, недействительной подписи в ходе сканирования или проверки. Подробнее о правилах реагирования см. в Руководстве пользователя.
5. При необходимости настроить интеграцию с CI/CD. Подробнее об интеграции в CI/CD см. в Руководстве пользователя.

7. Интерфейс PT CS

Работа с PT CS выполняется через веб-интерфейс. Для доступа пользователей к страницам веб-интерфейса и основным функциям PT CS используется главное меню. В главном меню расположены разделы для перехода к страницам веб-интерфейса и кнопки для управления учетной записью и выхода из нее.

Разделы для перехода к страницам веб-интерфейса:

- **Сводка** — страница для просмотра сводной информации о сканировании объектов в PT CS.
- **Соответствие стандартам** — страница для запуска и просмотра результатов проверки всех конфигураций кластера на соответствие стандартам информационной безопасности.
- **Карта кластера** — страница для построения [карты кластера \(см. раздел 8\)](#).
- **Кластеры** — страница [для управления дочерними кластерами \(см. раздел 13.3\)](#).
- **Образы** — страница [для управления реестрами \(см. раздел 10\)](#) и работы с образами. Подробнее о работе с образами см. в Руководстве пользователя.
- **Сканирования образов** — страница для просмотра результатов сканирования образов на уязвимости и ВПО. Подробнее о сканировании образов см. в Руководстве пользователя.
- **Проверка конфигураций** — страница для просмотра результатов проверки конфигураций на уязвимости. Подробнее о проверке конфигураций см. в Руководстве пользователя.
- **Контроллер допуска** — страница для управления проверками подписи образов и просмотра событий проверки действий через контроллер допуска и проверки подписи образов. Подробнее о проверке через контроллер допуска см. в Руководстве пользователя.
- **Среда выполнения** — страница [для управления параметрами отслеживания и записи событий, детекторами \(см. раздел 9\)](#) и просмотра событий среды выполнения. Подробнее о мониторинге и реагировании на события среды выполнения см. в Руководстве пользователя.
- **Пользователи** — страница [для управления локальными учетными записями пользователей, подключением к LDAP-серверу и корпоративными учетными записями \(см. раздел 13.4\)](#).
- **Правила** — страница для управления правилами реагирования. Подробнее об управлении правилами реагирования см. в Руководстве пользователя.
- **Сервисы уведомлений** — страница [для управления сервисами и шаблонами уведомлений \(см. раздел 11\)](#).

В нижней части главного меню расположены кнопки  для управления учетной записью и  для выхода из нее.

При нажатии кнопки  открывается меню настройки учетной записи со следующими пунктами:

-  — для перехода на страницу **Параметры системы**;
-  — для перехода на страницу **Токены доступа**;
-  — для смены пароля;
-  — для скачивания утилиты командной строки `ptcs-cli`;
-  — для скачивания руководства администратора и справочника по API в формате PDF.

Страница Сводка

На странице **Сводка** отображаются блоки с информацией о сканировании объектов в PT CS. Вверху страницы справа вы можете выбрать кластер, для которого будет отображаться информация, и период, за который будет отображаться информация: последние 24 часа, последние 7 дней или последний месяц.

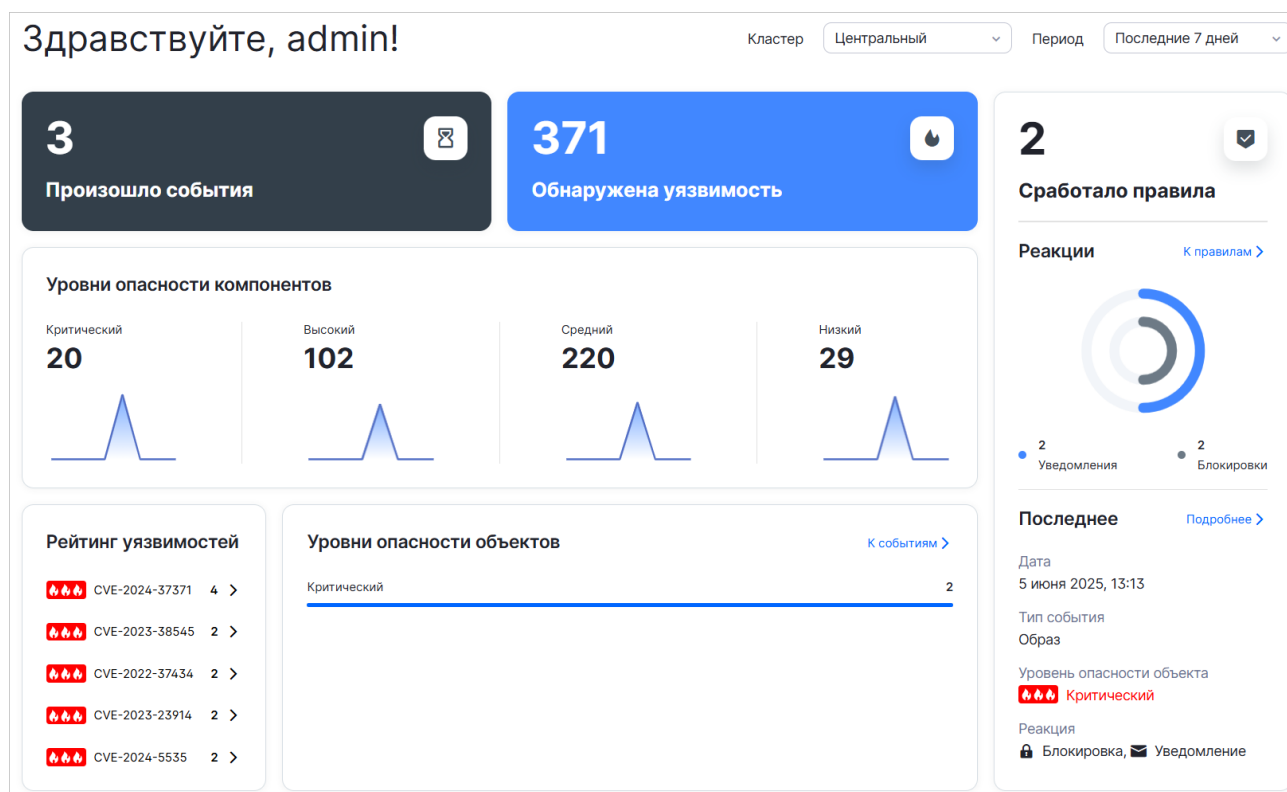


Рисунок 2. Страница **Сводка**

В рабочей области страницы находятся следующие элементы:

- В верхней части отображается количество произошедших событий и обнаруженных уязвимостей за выбранный период.
- В центральной части находятся диаграммы уровней опасности компонентов в просканированных объектах.
- В нижней части находятся:
 - рейтинг распространенности обнаруженных уязвимостей в порядке убывания. При нажатии на строку с уязвимостью открывается страница **Сканирования** с событиями сканирования, в ходе которых была обнаружена уязвимость;
 - количество просканированных объектов с определенным уровнем опасности. По нажатию **К событиям** открывается страница **Сканирования**.
- В правой части показана информация о правилах реагирования:
 - в верхней части блока отображается количество правил, сработавших за выбранный период;
 - в центральной части блока отображается диаграмма реакций, выполненных по результатам сканирования: количество заблокированных объектов и объектов, по которым были отправлены уведомления. По нажатию **К правилам** открывается страница **Правила**;
 - в нижней части блока отображаются сведения о последнем событии с самым высоким уровнем опасности. По нажатию **Подробнее** открывается страница с результатами сканирования объекта.

Страница Параметры системы

Вы можете открыть страницу **Параметры системы**, нажав  в главном меню.

Параметры системы

Кластер

Центральный

Лицензия

XXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX

Действительна до 31 декабря 2030, 23:59

Активировать новую лицензию



Общая информация

Версия системы
v1.1.0

Версия баз уязвимостей
4.20251218.002

Сканеры	100
Количество сканеров, проверяющих образы	
Артефактории	200
Количество артефакториев, которые можно подключить для сканирования	
Узлы кластера	100
Количество узлов кластера, на которых будут запущены агенты мониторинга рантайма	
Кластеры	10
Количество кластеров, которые можно подключить	

© Positive Technologies, 2026

[Лицензионное соглашение](#)

Рисунок 3. Страница **Параметры системы**

На этой странице вы можете:

- Узнать общую информацию о PT CS в центральном или дочернем кластере (например, версию системы и версию баз уязвимостей).
- [Активировать или заменить лицензию \(см. раздел 4\)](#) в центральном кластере.
- Просмотреть информацию об активированной лицензии в центральном или дочерних кластерах.

8. О карте кластера

PT CS предоставляет возможность просмотреть состав и оценить состояние кластера, в котором он установлен. Страница **Карта кластера** открывается при выборе в главном меню раздела **Карта кластера**. На карте отображается информация о нодах, пространствах имен (неймспейсах), подах, их взаимосвязи, контейнерах и событиях среды выполнения в контейнерах.

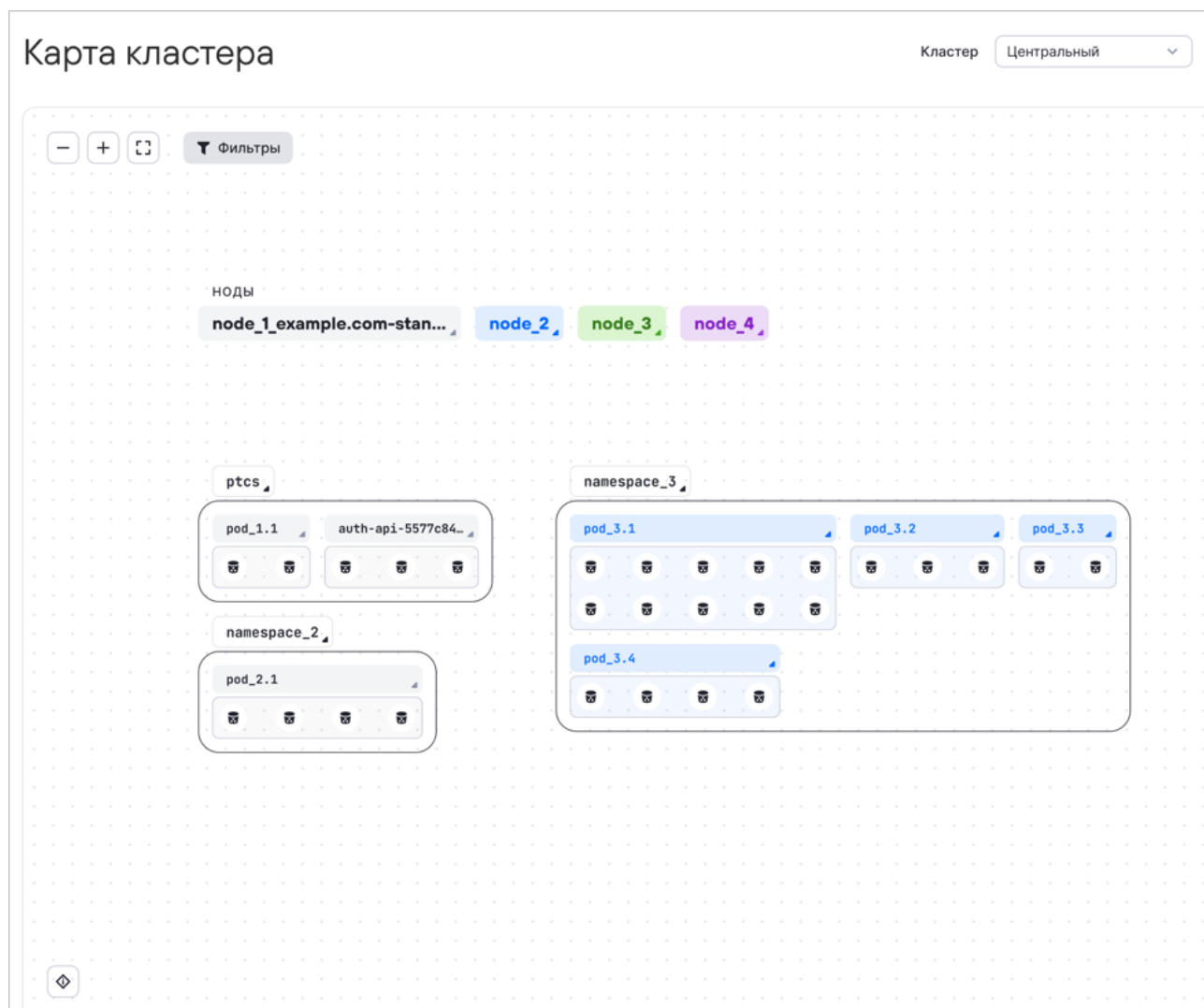


Рисунок 4. Страница **Карта кластера**

В правой верхней части страницы отображается раскрывающийся список **Кластер**, позволяющий выбрать кластер для работы. Если к PT CS не подключены дочерние кластеры или дочерние кластеры еще не находятся в статусе **Установлен**, то для выбора доступно только значение **Центральный**.

Вы можете управлять масштабом карты колесом мыши или с помощью кнопок  и . При нажатии кнопки  карта возвращается в исходное состояние для удобства просмотра. По кнопке **Фильтры** можно настроить фильтрацию объектов на карте по нодам, подам, неймспейсам и контейнерам.

В левой нижней части страницы отображается кнопка , нажав которую вы можете открыть легенду с описанием использованных цветов и символов.

Каждая нода на карте визуализируется в уникальном цвете, позволяющем различать ее и связанные с ней поды среди остальных элементов кластера. Поды группируются в неймспейсы. В одном неймспейсе могут быть сгруппированы поды из разных нод.

Вы можете просмотреть подробную информацию о конкретной ноде, конкретном неймспейсе или поде, нажав на них. Подробная информация отображается в боковой панели, которая открывается справа.

Информация о нодах

В боковой панели для ноды отображаются вкладки **Структура** и **Конфигурация**. На вкладке **Структура** отображается сгруппированная по неймспейсам информация о подах и количестве запущенных в поде контейнеров. При нажатии на неймспейс или под выполняется переход к нему на карте кластера и открывается боковая панель с подробной информацией.

Информация о неймспейсе

В боковой панели для неймспейса отображаются:

- нода;
- список подов, относящихся к этому неймспейсу;
- количество контейнеров, запущенных в каждом поде.

При нажатии на под выполняется переход к нему на карте кластера и открывается боковая панель с подробной информацией.

Информация о поде

В боковой панели для пода отображаются:

- Вкладка **Структура**, которая содержит информацию о состоянии пода, связанной ноде, неймспейсе и список запущенных контейнеров. Под может находиться в состоянии **Работает** или **В ожидании** (под создан, но еще не запущен). Нажав на контейнер, вы можете просмотреть информацию о событиях среды выполнения в этом контейнере.
- Вкладка **Сводка** с информацией о событиях среды выполнения для всех контейнеров пода.
- Вкладка **Конфигурация** с вкладками **Проверка на уязвимости** (для просмотра информации об обнаруженных уязвимостях и их уровне опасности) и **Код** (для просмотра манифеста YAML).

Информация о контейнерах и событиях среды выполнения

В боковой панели для каждого контейнера отображается общее количество произошедших событий и таблица с подробной информацией об обнаруженных в событиях угрозах. Таблица содержит:

- идентификатор детектора, при проверке через который обнаружены угрозы;
- уровень опасности угрозы;
- количество событий, в которых обнаружены угрозы при проверке.

Вы можете перейти на страницу просмотра событий среды выполнения, нажав **К событиям среды выполнения**. Подробнее о просмотре событий среды выполнения см. в Руководстве пользователя.

9. Мониторинг и реагирование на события среды выполнения

Вы можете настроить мониторинг и реагирование на события среды выполнения (runtime). Мониторинг событий среды выполнения позволяет отслеживать события на уровне отдельных подов или контейнеров в кластерах Kubernetes, включая запуск процессов, системные вызовы и обращения к конкретным функциям ядра. В ходе мониторинга и реагирования на события осуществляется их проверка через цепочку детекторов, которые обнаруживают в событии угрозы и присваивают им уровень опасности. Правила реагирования, созданные в PT CS, позволяют настроить реакции, выполняемые при обнаружении угрозы.

Детектор — это программа, написанная на тьюринг-полном языке программирования (например, Go, Python, C, C++, Rust), скомпилированная в формат WebAssembly (Wasm), которая может динамически подгружаться в систему и выгружаться без перезагрузки и позволяет реализовать безопасные и близкие по скорости к нативному коду алгоритмы проверки и анализа входящих событий.

Вы можете настроить отслеживание событий среды выполнения из конкретных источников, чтобы получать результаты проверки только требуемых событий и снизить поток событий, обрабатываемых PT CS.

Источник — это описание логики работы для eBPF-программ на специальном языке (TracingPolicy), которые загружаются и выполняются в ядре ОС на узле и позволяют отслеживать события среды выполнения с помощью механизмов `kprobe`, `uprobe`, `tracepoint` и BPF - LSM ядра Linux. Источники используют описанные выше механизмы для наблюдения за активностью процессов в контейнерах (отслеживания системных вызовов и обращений к конкретным функциям ядра) и обнаружения угроз. Всегда отслеживаются события запуска и остановки процессов в контейнерах кластера. Вы можете изменять существующие источники или добавлять новые, включив режим эксперта.

Внимание! Рекомендуется настраивать источники и фильтры событий с учетом реальной нагрузки на защищаемые кластеры, чтобы избежать чрезмерной нагрузки на PT CS. При росте нагрузки может потребоваться масштабирование системы.

В этом разделе приводится описание принципа мониторинга и реагирования на события среды выполнения в PT CS, а также инструкции по работе с такими событиями в системе.

Процесс мониторинга и реагирования на события среды выполнения состоит из следующих этапов:

1. Проверка событий среды выполнения через цепочку детекторов и установление уровня опасности для обнаруженных угроз.
2. Сохранение в БД событий в зависимости от параметров их сохранения. Сохраненные события отображаются на вкладке **События** страницы **Среда выполнения**.
3. Проверка событий по правилам реагирования, созданным в PT CS, для определения дальнейших реакций.

Если в результате проверки события сработало правило реагирования, то в системе регистрируется инцидент.

Управление мониторингом и реагированием на события среды выполнения в PT CS осуществляется в веб-интерфейсе на странице **Среда выполнения**, которая открывается при выборе в главном меню раздела **Среда выполнения**.

Среда выполнения Кластер: Центральный

Параметры События Детекторы ↶ Восстановить по умолчанию

Источники Режим эксперта + Добавить

Режим эксперта
Изменение источников может привести к неработоспособности отдельных детекторов или всей системы. Для событий от таких источников в MaxPatrol SIEM могут не регистрироваться инциденты информационной безопасности

Правила (1)
Правило для событий среды выполнения
Уведомление для событий среды ...

- ☒ Запуск и остановка процессов
- ☒ Исходящие TCP-соединения
- ☒ Обращения к важным системным файлам
- ☒ Действия со стандартными потоками ввода-вывода
- ☒ Создание и использование экземпляра интерфейса io_uring
- ☒ Загрузка и выгрузка модулей ядра
- ☒ Открытие сокета для входящих соединений
- ☒ Мониторинг устройств
- ☒ Действия с правами доступа файлов и процессов
- ☒ Использование инструментов для отладки и реверс-инжиниринга
- ☒ Мониторинг загрузки руткитов
- ☒ Использование usermode helper API

Фильтры событий + Добавить

Разрешающие

Поды runtime

Записывать события

☐ Все

☒ Только с угрозами

☐ Не записывать

Рисунок 5. Вкладка **Параметры**

Вверху страницы справа отображается раскрывающийся список **Кластер** для выбора кластера, с которым будет осуществляться работа. Если к PT CS не подключены дочерние кластеры или дочерние кластеры еще не перешли в статус **Установлен**, то для выбора доступно только значение **Центральный**.

Страница содержит следующие вкладки:

- **Параметры** — позволяет [управлять источниками событий, разрешающими и запрещающими фильтрами, а также настраивать сохранение событий \(см. раздел 9.2\)](#).
- **События** — позволяет просматривать информацию о событиях, которые были сохранены в PT CS в соответствии со значениями на вкладке **Параметры**.
- **Детекторы** — позволяет просматривать информацию о детекторах, [добавлять и удалять их \(см. раздел 9.3\)](#).

В этом разделе

[Управление источниками отслеживания событий среды выполнения \(см. раздел 9.1\)](#)

[Управление параметрами отслеживания и фильтрами событий \(см. раздел 9.2\)](#)

[Управление детекторами \(см. раздел 9.3\)](#)

9.1. Управление источниками отслеживания событий среды выполнения

В комплект поставки PT CS входит набор источников, на который распространяется лицензия PT CS. Источники из комплекта поставки и типы отслеживаемых ими событий описаны в таблице ниже.

Таблица 13. Источники и типы отслеживаемых ими событий

Название источника	Типы отслеживаемых событий
Исходящие TCP-соединения	Этот источник отслеживает функции <code>tcp_connect</code> , <code>tcp_close</code> и <code>tcp_sendmsg</code> , позволяя обнаруживать исходящие TCP-соединения (включая установку, завершение соединения и отправку TCP-пакетов). Включение этого источника может существенно увеличить поток событий и нагрузку на систему. В этом случае рекомендуется сузить поток с помощью более точных фильтров, например указать конкретные поды
Действия с правами доступа файлов и процессов	Источник отслеживает вызовы функции ядра Linux <code>commit_creds()</code> , которые могут указывать на попытки злоумышленника повысить привилегии процесса, включая получение прав суперпользователя (<code>root</code>). Кроме того, источник отслеживает вызовы LSM-функции <code>security_path_chmod()</code> с набором разрешений, включающих права на исполнение

Название источника	Типы отслеживаемых событий
Обращения к важным системным файлам	Источник отслеживает вызовы функций ядра Linux <code>security_file_permission()</code> , <code>security_mmap_file()</code> , <code>security_path_truncate()</code> , <code>security_path_link()</code> и <code>security_path_rename()</code> в адрес важных системных каталогов и файлов. К ним относятся, например, <code>/boot</code> , <code>/root/.ssh</code> , <code>/etc/shadow</code> , <code>/etc/profile</code> , <code>/etc/sudoers</code> , <code>/etc/passwd.conf</code> . Для разных файлов отслеживаются разные действия (только чтение, только запись, чтение и запись, создание жестких ссылок). Для детального изучения работы источника рекомендуется включить экспертный режим
Использование инструментов для отладки и реверс-инжиниринга (<code>ptrace</code>)	Этот источник отслеживает системный вызов <code>ptrace</code> , который может свидетельствовать об активности злоумышленника в целевой системе
Запуск и остановка процессов	Этот источник отслеживает запуск и остановку процессов в контейнерах на узлах с установленными компонентами мониторинга среды выполнения. Информация о том, в контексте какого процесса возникает то или иное событие, добавляется в события других типов (например, <code>kprobe</code>) от других источников, поэтому этот источник нельзя отключить. Его влияние на общую нагрузку системы, как правило, незначительное
Загрузка и выгрузка модулей ядра	Этот источник отслеживает вызовы <code>do_init_module</code> , <code>free_module</code> , <code>security_kernel_module_request</code> и <code>security_kernel_read_file</code> , позволяя фиксировать явную или неявную (автоматическую) загрузку и выгрузку модулей, а также попытки манипуляции модулями и иную активность злоумышленника
Мониторинг сокетов	Источник отслеживает вызовы функций ядра Linux <code>inet_csk_listen_start()</code> , <code>security_socket_create()</code> и <code>security_socket_connect()</code> , указывающих на создание сокетов для входящих соединений, сокетов специфических семейств про-

Название источника	Типы отслеживаемых событий
	токолов и попытки подключения к Unix-сокетами. Это помогает обнаруживать подозрительную сетевую активность, связанную с эксплуатацией уязвимостей ядра или использованием нестандартных протоколов, а также обращение к Unix-сокетами контейнерных сред выполнения
Монтирование устройств	Источник отслеживает вызов <code>mount</code> , позволяя обнаруживать потенциально нежелательные события, связанные с монтированием устройств
Действия со стандартными потоками ввода-вывода	Источник отслеживает вызовы функции ядра Linux <code>do_dup2()</code> , выполняющей копирование файлового дескриптора стандартного ввода (STDIN), а также создание файла именованного канала (S_IFIFO) через LSM-функцию <code>security_path_mknod()</code> . Часто такие действия свидетельствуют о том, что злоумышленник пытается запустить реверс-шелл, скрытый канал связи или другой инструмент для атаки
Мониторинг интерфейса <code>io_uring</code>	Источник отслеживает системные вызовы <code>io_uring_setup()</code> и <code>io_uring_enter()</code> , которые указывают на использование интерфейса <code>io_uring</code> в системе
Использование <code>usermode helper API</code>	Источник отслеживает вызовы функций ядра Linux <code>call_usermodehelper_setup()</code> и <code>call_usermodehelper_exec()</code> , которые указывают на настройку и запуск процессов через <code>usermode helper API</code> . Это помогает обнаружить несколько различных атак, использующих указанный интерфейс с целью запуска процессов на уровне ОС узла
Мониторинг загрузки руткитов	Источник отслеживает вызовы функции ядра Linux <code>kallsyms_lookup_name()</code> для получения адреса таблицы системных вызовов, что может свидетельствовать об активности руткита на целевой системе. Источник также отслеживает загрузку eBPF программ с помощью функции <code>bpfc_check()</code> , что может помочь в поиске руткитов использующих eBPF

Работа с источниками

Вы можете добавлять и изменять источники, включив режим эксперта. Изменения источников могут привести к неработоспособности отдельных детекторов или всей системы. При интеграции с MaxPatrol SIEM для событий от таких источников не будут регистрируются события информационной безопасности. Режим эксперта доступен только пользователям с ролью **Администратор**.

Управление источниками осуществляется в блоке **Источники** на вкладке **Параметры** страницы **Среда выполнения**. В блоке **Источники** отображаются переключатель **Режим эксперта** и список источников. Для каждого источника отображается флажок для включения или выключения отслеживания событий от этого источника. Вы можете открыть окно с информацией об источнике, нажав на строку с его названием. В окне отображаются название, описание и код источника. Для кода источника отображаются кнопки:



— для включения переноса по словам;



— для копирования кода источника;



— для сохранения кода в текстовом файле.

При включенном режиме эксперта отображается кнопка **Добавить** для добавления источника. После изменения источников или их добавления появляется кнопка **Восстановить по умолчанию**, позволяющая восстановить первоначальное состояние источников и фильтров событий. Если настроены [интеграция с Grafana и мониторинг работы компонентов](#) (см. раздел 13.5) PT CS, то после изменения или добавления источников вы можете перейти к просмотру метрик.

► Чтобы добавить источник для отслеживания событий среды выполнения:

1. В главном меню выберите **Среда выполнения**.
2. Вверху страницы справа выберите кластер, для которого вы хотите добавить источник.
3. Включите режим эксперта.
4. Нажмите **Добавить**.
5. Введите название, описание и код источника.

Примечание. Код источника будет автоматически проверен. При обнаружении ошибки будет выведено сообщение с указанием номера первой из строк, содержащих ошибки.

6. Нажмите **Добавить**.
7. Нажмите **Применить**.

Добавленный источник отобразится последним в списке. Для начала отслеживания событий от нового источника вам нужно включить его.

Примечание. Вы не можете изменить или отключить источник **Запуск и остановка процессов**. Этот источник отслеживает запуск и остановку процессов в контейнерах на узлах с установленными компонентами мониторинга среды выполнения и добавляет контекст процесса к событиям других типов (например, kprobe) от других источников.

Вы можете удалить источник, нажав кнопку , которая появляется при наведении курсора на строку с источником.

9.2. Управление параметрами отслеживания и фильтрами событий

Настройка параметров отслеживания событий

Вы можете настроить параметры отслеживания событий среды выполнения. Это позволит вам получать информацию о результатах проверки только требуемых событий и снизит нагрузку на компоненты системы.

► Чтобы настроить параметры отслеживания событий:

1. В главном меню выберите **Среда выполнения**.
2. Вверху страницы справа выберите кластер, для которого нужно настроить параметры отслеживания.

Примечание. Если при развертывании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

3. Установите флажки [для источников \(см. раздел 9.1\)](#) событий среды выполнения, которые вы хотите отслеживать.

Примечание. Вы можете посмотреть подробное описание источника, нажав на строку с его названием.

4. Добавьте разрешающие и запрещающие фильтры событий.
5. Выберите вариант записи событий:

- Если вы хотите, чтобы записывались и отображались на вкладке **События** все события, выберите **Все**.

Примечание. Запись всех событий позволяет проводить полноценные расследования, но чтобы избежать чрезмерной нагрузки на PT CS, рекомендуется настраивать источники и фильтры событий с учетом реальной нагрузки на защищаемые кластеры. При росте нагрузки может потребоваться масштабирование системы.

- Если вы хотите, чтобы записывались и отображались на вкладке **События** только те события, в которых обнаружены угрозы, выберите вариант **Только с угрозами**.

Примечание. При выборе варианта **Только с угрозами** возможности расследования инцидентов будут ограничены. Его можно выбрать, если БД не справляется с нагрузкой при параметрах, заданных в блоках **Отслеживание событий** и **Списки фильтров событий**.

- Если вы хотите, чтобы реагирование на события выполнялось, но сами события не записывались и информация о них не отображалась на вкладке **События**, выберите вариант **Не записывать**.

Примечание. При выборе варианта **Не записывать** расследование инцидентов будет невозможно.

6. Нажмите **Применить**.

Управление фильтрами событий

Вы можете добавить разрешающие и запрещающие фильтры событий, чтобы настроить поток событий, поступающих в PT CS для мониторинга и реагирования.

Добавление разрешающего фильтра означает, что события из указанных пространств имен (namespaces) и подов (pods) и с указанными метками (labels) будут обрабатываться цепочкой детекторов в PT CS. События, которые не удовлетворяют условиям фильтров, не будут обрабатываться.

Если добавлено несколько разрешающих фильтров, событие должно подпадать под условия хотя бы одного из них для обработки. При добавлении нескольких запрещающих фильтров событие исключается из проверки, если подпадает под условия хотя бы одного такого фильтра.

Если добавлены разрешающие и запрещающие фильтры, то сначала событие проверяется разрешающими фильтрами и, если оно соответствует хотя бы одному из них, то проверяется, что событие не попадает под действие запрещающих фильтров, — тогда событие обрабатывается цепочкой детекторов. Если же событие не попадает под разрешающие фильтры или попадает и под разрешающие, и под запрещающие фильтры, то событие не обрабатывается.

► Чтобы добавить фильтр событий:

1. В главном меню выберите **Среда выполнения**.
2. Вверху страницы справа выберите кластер, для которого вы хотите добавить фильтр событий.

Примечание. Если при развертывании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

3. Нажмите **Добавить фильтр**.
4. Выберите, какой фильтр вы хотите добавить: **Разрешающий** или **Запрещающий**.

5. Укажите пространства имен (namespaces) для фильтрации событий.
6. Укажите регулярные выражения для определения подов. Формат записи регулярных выражений вы можете найти [на сайте документации языка Go](#).

Примечание. Например, регулярное выражение `news-service` в виде простой строки без дополнительных символов и конструкций будет соответствовать названиям подов, которые содержат указанную строку в любом месте названия.

7. Введите метки подов в формате ключ=значение. Между ключом и значением разрешены три вида сравнений: `=`, `==`, `!=`. Формат записи для ключей и значений меток вы можете найти [в документации Kubernetes](#).

Примечание. Например, если задано значение `app.kubernetes.io/name=tetragon`, то будут отфильтрованы все поды с ключом метки `app.kubernetes.io/name` и значением `tetragon`.

8. Нажмите **Добавить**.
9. Нажмите **Применить**.

Вы можете изменить фильтр по кнопке  и удалить фильтр событий по кнопке . Невозможно удалить единственный фильтр событий, так как без фильтров будет отслеживаться слишком много событий. После удаления фильтра события, которые попадали под него, будут отслеживаться в PT CS без условий.

9.3. Управление детекторами

В комплект поставки PT CS входит набор детекторов, на который распространяется лицензия PT CS. Заказчики, сторонние поставщики и сообщество также могут разрабатывать детекторы и загружать их в систему.

Управление детекторами осуществляется на странице **Среда выполнения** на вкладке **Детекторы**. Информация обо всех детекторах (идентификатор, название, версия, описание и соответствующие тактики и техники MITRE ATT&CK) представлена в виде таблицы.

Среда выполнения

Кластер

Центральный

Параметры

События

Детекторы

Добавить

Идентификатор	Название	Версия	Тактики и техники MITRE ATT&CK	
ESC_RT_BASE64_DECODE_RUN	Base64: декодирование данных	1	Предотвращение обнаружения 3 Организация	
ESC_RT_BIN_PERM_RAISE	Изменение прав доступа к файлу	3	Повышение привилегий 2 Предотвращени	
ESC_RT_CORE_PATTERN_MODIFY	Изменение параметров генерации дампа-файла процесса	1	Повышение привилегий 1	
ESC_RT_CVE_2025_1974	Уязвимость Ingress Nightmare (запуск вредоносного кода)	1	Первоначальный доступ 1 Выполнение 1	
ESC_RT_FIFO_FILE_CREATE	Создание файла именованного канала	1	Выполнение 1	
ESC_RT_HARDLINK_CREATE	Создание жестких ссылок на системные файлы	1	Предотвращение обнаружения 1	
ESC_RT_IO_URING_CREATE	Создание экземпляра интерфейса io_uring	1	Выполнение 1	

Рисунок 6. Вкладка **Детекторы**

Информация о детекторе

По нажатию на детектор открывается боковая панель, в которой отображаются его идентификатор, название, версия, описание и информация о соответствующих тактиках и техниках матрицы MITRE ATT&CK.

По ссылке **К событиям с детектором** вы можете перейти к списку событий, в которых были обнаружены угрозы при проверке через этот детектор.

Добавление детектора

► Чтобы добавить детектор:

1. В главном меню выберите **Среда выполнения**.
2. Выберите вкладку **Детекторы**.
3. Если требуется, вверху вкладки справа выберите кластер, для которого вы хотите добавить детектор.

Примечание. Если при разворачивании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

4. Нажмите **Добавить**.
5. Выберите или перетащите файлы для загрузки.

Примечание. Детектор должен быть написан на тьюринг-полном языке программирования (например, Go, Python, C, C++, Rust) и скомпилирован в формат WebAssembly (WASM).

6. Нажмите **Добавить**.

Вы можете удалить детектор по кнопке  .

10. Управление реестрами образов

Как правило, при разработке ПО все артефакты хранятся в единых реестрах организации или публичных реестрах. Для регулярного анализа и выявления устаревших образов или образов с уязвимостями вы можете подключать к PT CS реестры с помощью API вендора или общего API — Docker API v2. В веб-интерфейсе PT CS вы можете управлять реестрами на странице **Образы**, которая открывается при выборе в главном меню раздела **Образы**.

В правой верхней части страницы отображается раскрывающийся список **Кластер**, позволяющий выбрать кластер для работы. Если к PT CS не подключены дочерние кластеры или дочерние кластеры еще не находятся в статусе **Установлен**, то для выбора доступно только значение **Центральный**.

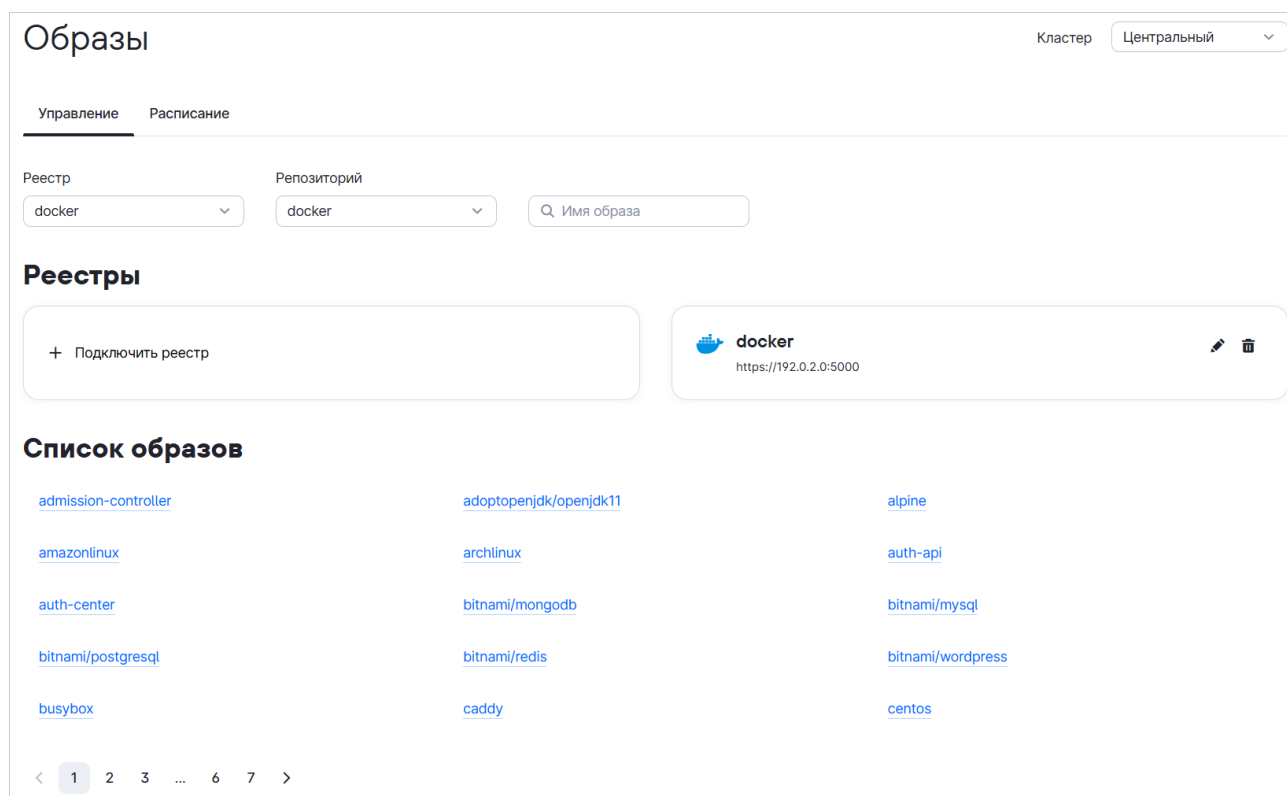


Рисунок 7. Вкладка **Управление**

На странице **Образы** доступны вкладки **Управление** и **Расписание**. На вкладке **Управление** доступно управление реестрами и работа с образами. На вкладке **Расписание** осуществляется управление расписанием периодического сканирования. Подробнее о работе с образами и расписании сканирования см. в Руководстве пользователя.

На вкладке **Управление** отображаются:

- раскрывающиеся списки **Реестр** и **Репозиторий** для фильтрации;
- кнопка **Подключить реестр** для подключения реестра;
- список подключенных реестров и кнопки для изменения параметров подключения к реестру и удаления.

В этом разделе приведены инструкции по настройке реестров разных вендоров для подключения к PT CS, а также инструкции по управлению реестрами в PT CS.

В этом разделе

[Настройка реестров для подключения к PT CS \(см. раздел 10.1\)](#)

[Работа с реестрами образов \(см. раздел 10.2\)](#)

10.1. Настройка реестров для подключения к PT CS

В PT CS вы можете сканировать образы из реестров разных вендоров. К PT CS подключается реестр, который содержит репозитории образов. У каждого образа в репозитории обязательно есть имя и один или множество тегов. Для описания действий, связанных с настройкой доступа к объектам в реестрах, используются различные термины: для реестров вендора Docker — термин «область», Nexus — «права», GitLab — «токен», JFrog — «токен», «область».

Для подключения реестра к PT CS необходимо настроить его с учетом рекомендаций ниже.

Настройка Docker Registry

Для подключения реестра Docker Registry у пользователя, от имени которого будут выполняться действия в PT CS, должны быть права на просмотр каталогов и списка тегов, на скачивание образов, которые необходимо сканировать, и доступ к конечным точкам:

- `/v2/_catalog` с областью `repository:catalog:pull,push`;
- `/v2/<Имя образа>/tag/list` с областью `repository:<Имя образа>:pull`;
- `/v2/<Имя образа>/manifests/<Имя тега>` с областью `repository:<Имя образа>:pull`;
- `/v2/<Имя образа>/blobs/{image_meta_info["config"] ["digest"]}` с областью `repository:<Имя образа>:pull`.

Настройка Nexus Repository

Для подключения реестра Nexus Repository у пользователя, от имени которого будут выполняться действия в PT CS, должны быть права администратора или права `nx-metrics-all`, `nx-repository-admin-docker-<Имя репозитория>-*` и `nx-repository-view-docker-<Имя репозитория>-*` на каждый репозиторий.

Настройка GitLab

Для подключения реестра GitLab используются токен API. Токен может быть пользовательским или групповым. Вам необходимо заменить роль **Guest**, присваиваемую по умолчанию при создании группового токена в проекте реестра, на роль **Reporter** или другую роль с более широкими полномочиями. У токена, от имени которого будут выполняться действия в PT CS, должны быть разрешения **read_api** и **read_registry**.

Настройка JFrog

Для подключения реестра JFrog у пользователя, от имени которого будут выполняться действия в PT CS, должно быть право на чтение подключаемого к PT CS реестра и доступ к конечным точкам:

- `/v2/_catalog` с областью `repository:catalog:pull,push`;
- `/v2/<Имя образа>/tag/list` с областью `repository:<Имя образа>:pull`;
- `/v2/<Имя образа>/manifests/<Имя тега>` с областью `repository:<Имя образа>:pull`;
- `/v2/<Имя образа>/blobs/{image_meta_info["config"] ["digest"]}` с областью `repository:<Имя образа>:pull`.

Настройка Yandex Cloud

Для подключения реестра вам необходимы учетная запись пользователя Yandex Cloud с правами доступа уровня роли **container-registry.viewer** или выше, идентификатор этой учетной записи, идентификатор каталога проекта, а также идентификатор авторизованного открытого ключа и закрытая часть ключа.

Вы можете узнать идентификатор учетной записи (ID) в профиле Yandex Cloud, идентификатор каталога проекта (Folder ID) — в профиле каталога внутри проекта.

Идентификатор открытого ключа и закрытую часть вы можете получить с помощью запроса на создание авторизованного ключа. Подробнее — [в документации Yandex Container Registry](#).

Настройка других реестров

Возможна интеграция с другими реестрами через стандартный интерфейс Docker API v2.

10.2. Работа с реестрами образов

Вы можете сканировать образы из удаленных реестров: Nexus, GitLab, Docker, JFrog и Yandex Cloud. Для этого вам нужно подключить реестр в PT CS.

► Чтобы подключить реестр:

1. В главном меню выберите **Образы**.
2. Вверху страницы справа выберите кластер, к которому будет подключен реестр.

Примечание. Если при развертывании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

3. Нажмите **Подключить реестр**.
4. Введите название реестра.
5. Выберите вендора реестра.
6. Выполните одно из следующих действий:
 - Если вы хотите подключить реестр без аутентификации, установите флажок **Подключить анонимно**.
 - Если вы хотите подключить реестр с аутентификацией, введите логин и пароль.
7. Введите значения других параметров для подключения реестра.

Примечание. Список параметров зависит от выбранного вендора.

8. Нажмите **Подключить**.

Примечание. Вы можете нажать **Подключить без проверки**, если хотите добавить реестр без подключения к нему. Например, если у вас пока нет к нему доступа.

Если вам больше не требуется просматривать и сканировать образы из реестра, вы можете удалить его из PT CS. При этом история сканирования образов из этого реестра сохранится и будет доступна на странице **Сканирования**.

11. Управление сервисами и шаблонами уведомлений

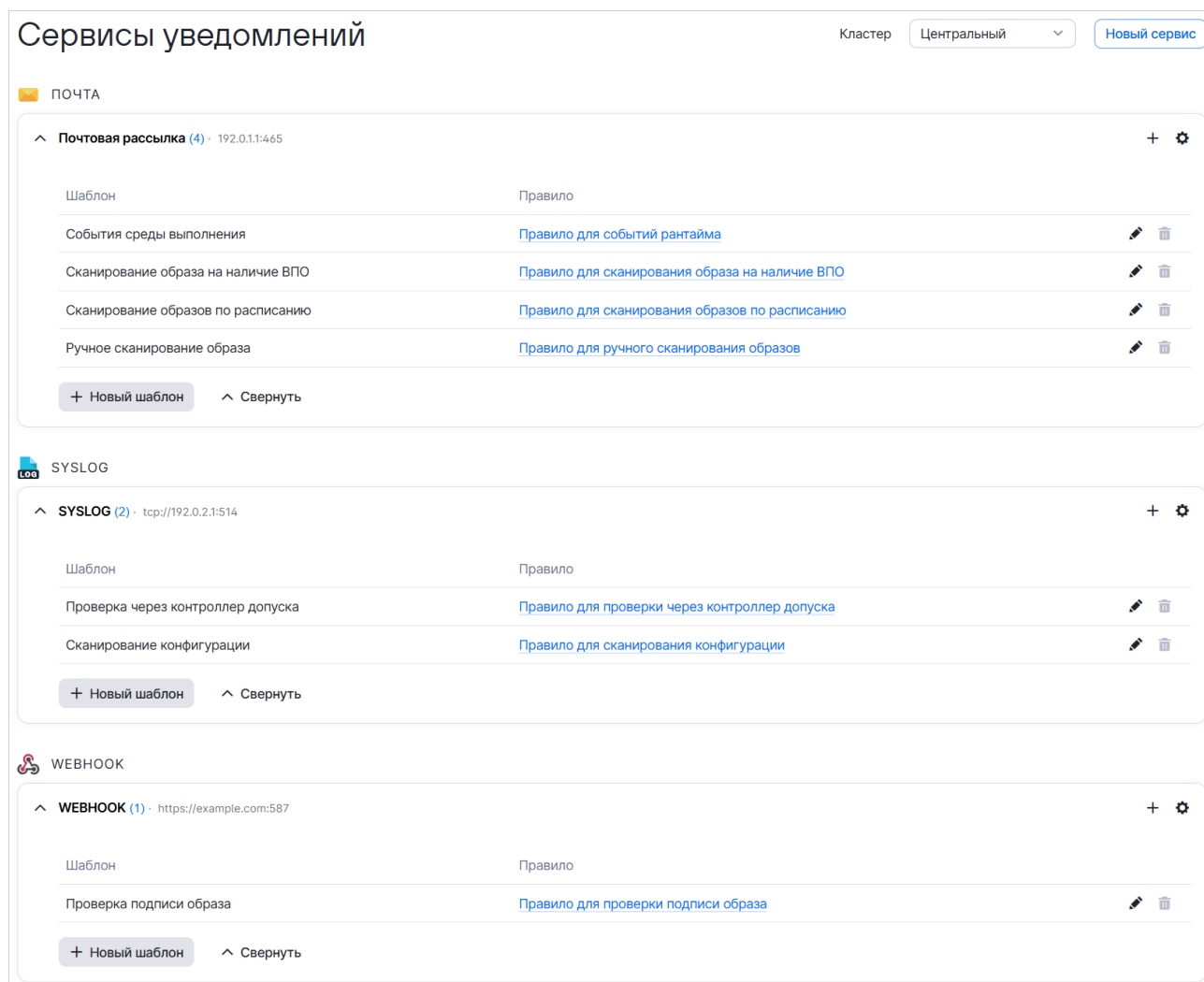
Вы можете интегрировать PT CS в существующие инфраструктуру и процессы обеспечения информационной безопасности контейнеров с помощью уведомлений по электронной почте, с помощью механизма webhook или по протоколу syslog.

Отправка электронных писем предназначена в основном для оповещения узкого круга заинтересованных лиц о событиях информационной безопасности, например сотрудников одного из отделов.

Уведомления с помощью механизма webhook позволяют реализовать интеграцию с такими сервисами, как Slack, Mattermost, Telegram или более сложными системами, например системами оркестрации (security orchestration, automation and response, SOAR). Если PT CS установлен в изолированном от интернета сегменте сети, вы можете реализовать отправку уведомлений с помощью механизма webhook через прокси-сервер, указав [его адрес \(см. раздел 5.5\)](#) при установке PT CS.

Вы можете интегрировать PT CS в процессы мониторинга информационной безопасности, настроив отправку уведомлений по протоколу syslog в системы SIEM или ILMS.

Управление сервисами и шаблонами уведомлений в PT CS осуществляется в веб-интерфейсе на странице **Сервисы уведомлений**, которая открывается при выборе в главном меню раздела **Сервисы уведомлений**.

Рисунок 8. Страница **Сервисы уведомлений**

На странице отображаются раскрывающийся список **Кластер** для выбора кластера и кнопка **Новый сервис** для подключения сервиса уведомлений. Если к PT CS не подключены дочерние кластеры или дочерние кластеры еще не перешли в статус **Установлен**, то для выбора доступно только значение **Центральный**.

На странице **Сервисы уведомлений** отображаются все подключенные сервисы, с помощью которых отправляются уведомления по шаблонам, а также шаблоны уведомлений. Отправка уведомлений настраивается в правиле реагирования. Подробную информацию о правилах реагирования см. в Руководстве пользователя.

Подключенные сервисы уведомлений сгруппированы по типам (например, почта, webhook или syslog). Рядом с названием каждого сервиса отображается количество шаблонов уведомлений, для которых он используется. В блоке с каждым сервисом отображаются кнопка **+** для добавления шаблона уведомления и кнопка **⚙** для изменения или удаления сервиса.

Вы можете удалить сервис уведомлений, если он больше не используется. Вместе с сервисом будут удалены все шаблоны уведомлений, которые были созданы для него.

Внимание! Сервис уведомлений нельзя удалить, если хотя бы один из шаблонов уведомлений в этом сервисе используется в существующем правиле реагирования. Подробную информацию о правилах реагирования см. Руководстве пользователя.

Кроме того, на странице отображаются элементы управления шаблонами, используемыми в сервисах уведомлений: кнопки **Новый шаблон** и **+** для добавления шаблона в сервис уведомлений и кнопки для изменения и удаления шаблона уведомлений.

Если шаблон уведомлений был создан по ошибке или его участникам больше не требуется отправлять уведомления, вы можете его удалить. Шаблон уведомлений можно удалить, если он не указан ни в одном правиле реагирования. Подробнее о правилах реагирования см. в Руководстве пользователя.

В этом разделе приведены инструкции по управлению сервисами и шаблонами уведомлений и описание параметров шаблонов уведомлений.

В этом разделе

[Управление сервисами и шаблонами уведомлений по электронной почте \(см. раздел 11.1\)](#)

[Управление сервисами и шаблонами уведомлений с помощью механизма webhook \(см. раздел 11.2\)](#)

[Управление сервисами и шаблонами уведомлений по протоколу syslog \(см. раздел 11.3\)](#)

[Параметры шаблонов уведомлений \(см. раздел 11.4\)](#)

11.1. Управление сервисами и шаблонами уведомлений по электронной почте

Вы можете настроить отправку уведомлений о результатах сканирования объектов и проверки действий по электронной почте. Например, уведомления о результатах сканирования образов будут получать сотрудники одного отдела, а уведомления о результатах сканирования конфигураций — сотрудники другого отдела. Для этого вам необходимо развернуть SMTP-сервер. В PT CS необходимо настроить подключение к нему, добавив сервис уведомлений, а также добавить один или несколько шаблонов уведомлений для сервиса. Вы можете использовать несколько SMTP-серверов для рассылки, добавив для каждого свой сервис уведомлений и набор шаблонов.

Добавление сервиса уведомлений по электронной почте

► Чтобы добавить сервис уведомлений:

1. В главном меню выберите **Сервисы уведомлений**.
2. Вверху страницы справа выберите кластер, для которого нужен добавить сервис уведомлений.

Примечание. Если при развертывании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

3. Нажмите **Новый сервис**.
4. Выберите тип **Почта**.
5. Введите название сервиса уведомлений.
6. Выберите тип аутентификации на SMTP-сервере.
7. Введите адрес и порт SMTP-сервера в формате <IP-адрес или доменное имя> : <Порт>.
8. Введите логин и пароль учетной записи для аутентификации на сервере.
9. Если требуется, укажите корневой сертификат удостоверяющего центра.
10. Введите адрес электронной почты отправителя, от имени которого будут приходить уведомления.
11. Если SMTP-сервер поддерживает безопасные подключения по протоколу TLS, установите флажок **Использовать TLS**.
12. Если ваш SMTP-сервер поддерживает безопасные подключения с использованием расширения STARTTLS, установите флажок **Использовать STARTTLS**.
13. Если требуется проверять безопасность соединения с сервером при использовании самоподписанного сертификата, установите флажок **Проверять безопасность соединения**.
14. Нажмите **Подключить**.

Примечание. Вы можете нажать **Подключить без проверки**, если хотите добавить сервис без подключения к SMTP-серверу. Например, если у вас пока нет доступа к нему.

После добавления сервиса уведомлений по электронной почте вам нужно указать адреса, на которые будут отправляться уведомления с помощью этого сервиса. Для этого необходимо создать один или несколько шаблонов уведомлений. В один шаблон могут быть объединены, например, специалисты из одного подразделения вашей организации.

Добавление шаблона уведомлений по электронной почте

► Чтобы добавить шаблон уведомлений:

1. В главном меню выберите **Сервисы уведомлений**.
2. Вверху страницы справа выберите кластер, для сервиса уведомлений которого нужно добавить шаблон уведомлений.

Примечание. Если при развертывании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

3. Нажмите **+** или **Новый шаблон** в блоке с сервисом уведомлений, для которого нужно добавить шаблон.
4. Введите название шаблона.
5. Введите один или несколько адресов получателей.

Примечание. В поле **Кластер** отображается кластер, для которого будет создан шаблон уведомления. Значение соответствует выбранному в раскрывающемся списке **Кластер** вверху страницы **Сервисы уведомлений** справа.

6. Если требуется, измените адрес центрального кластера.

Примечание. Адрес центрального кластера используется для формирования ссылки на событие в уведомлении. Поле **Адрес центрального кластера** заполняется автоматически. Если доступ к центральному кластеру организован нестандартным способом, например, кластер установлен в закрытом контуре, вы можете указать необходимый адрес для доступа.

7. Выберите тип события, о котором будут отправляться уведомления с помощью этого шаблона.
8. Введите тему письма, которая будет отображаться для получателей.
9. Если требуется, снимите флажок **Отправлять стандартное уведомление**.

Отобразится поле **Тело уведомления**, в котором вы можете добавить шаблон уведомления, используя [синтаксис шаблонизатора Go](#). Для каждого типа событий в PT CS определен [свой набор переменных \(см. раздел 11.4\)](#).

10. Нажмите **Добавить**.

Добавление или исключение адресата из рассылки

- Чтобы добавить или исключить адресата из рассылки:

1. В главном меню выберите **Сервисы уведомлений**.
2. Вверху страницы справа выберите кластер, для которого нужно добавить или исключить адресата из рассылки.

Примечание. Если при развертывании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

3. Нажмите  в строке с шаблоном, в котором нужно добавить или удалить получателя.
4. Добавьте или удалите один или несколько адресов электронной почты.
5. Нажмите **Сохранить**.

11.2. Управление сервисами и шаблонами уведомлений с помощью механизма webhook

Вы можете настроить отправку уведомлений о результатах сканирования объектов и проверки действий с помощью механизма webhook.

Webhook — это способ интеграции, при котором инициирующая система отправляет получателю уведомление в формате HTTP-запроса, в теле которого содержится вся необходимая информация.

Для настройки такого способа уведомлений в инфраструктуре необходим сервис, который поддерживает получение сообщений с помощью механизма webhook. Например, CI/CD-системы GitLab или Jenkins или системы оркестрации (security orchestration, automation and response, SOAR). Вы также можете развернуть webhook-сервер общего назначения, который будет выполнять необходимые вам операции.

В PT CS вам необходимо настроить подключение к webhook-серверу, добавив сервис уведомлений, а также добавить один или несколько шаблонов уведомлений для сервиса. Вы можете реализовать интеграцию с несколькими webhook-серверами, добавив для каждого свой сервис уведомлений и набор шаблонов.

Добавление сервиса уведомлений по webhook

► Чтобы добавить сервис уведомлений:

1. В главном меню выберите **Сервисы уведомлений**.
2. Вверху страницы справа выберите кластер, для которого нужен добавить сервис уведомлений.

Примечание. Если при разворачивании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

3. Нажмите **Новый сервис**.
4. Выберите тип **Webhook**.
5. Введите название сервиса уведомлений.
6. Введите URL сервера, который будет использоваться для рассылки уведомлений, а также порт (при наличии) в формате `https://<IP-адрес или доменное имя>:<Порт>`.

7. Если на сервере webhook используется базовая аутентификация (Basic auth), введите логин и пароль учетной записи для аутентификации на сервере.
8. Если требуется, укажите корневой сертификат удостоверяющего центра.
9. Если требуется проверять безопасность соединения с сервером при использовании самоподписанного сертификата, установите флажок **Проверять безопасность соединения**.
10. Нажмите **Подключить**.

Примечание. Вы можете нажать **Подключить без проверки**, если хотите добавить сервис без подключения к серверу webhook. Например, если у вас пока нет доступа к нему.

После добавления сервиса уведомлений webhook вам нужно указать пути (endpoint), на которые будут отправляться уведомления. Для этого необходимо создать один или несколько шаблонов уведомлений. Например, в одном шаблоне вы можете указать путь для отправки уведомлений о сканировании образов в одну группу Mattermost, а в другом шаблоне — путь для отправки уведомлений о сканировании конфигурации в другую группу.

Добавление шаблона уведомлений по webhook

► Чтобы добавить шаблон уведомлений:

1. В главном меню выберите **Сервисы уведомлений**.
2. Вверху страницы справа выберите кластер, для сервиса уведомлений которого нужно добавить шаблон уведомлений.

Примечание. Если при развертывании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

3. Нажмите **+** или **Новый шаблон** в блоке с сервисом уведомлений, для которого нужно добавить шаблон.
4. Введите название шаблона.

Примечание. В поле **Кластер** отображается кластер, для которого будет создан шаблон уведомления. Значение соответствует выбранному в раскрывающемся списке **Кластер** вверху страницы **Сервисы уведомлений** справа.

5. Если требуется, измените адрес центрального кластера.

Примечание. Адрес центрального кластера используется для формирования ссылки на событие в уведомлении. Поле **Адрес центрального кластера** заполняется автоматически. Если доступ к центральному кластеру организован нестандартным способом, например, кластер установлен в закрытом контуре, вы можете указать необходимый адрес для доступа.

6. Выберите тип события, о котором будут отправляться уведомления с помощью этого шаблона.
7. Введите путь (endpoint), на который будет отправляться webhook-запрос.
8. Если требуется, снимите флажок **Отправлять стандартное уведомление**.

Отобразится поле **Тело уведомления**, в котором вы можете добавить шаблон уведомления, используя [синтаксис шаблонизатора Go](#). Для каждого типа событий в PT CS определен [свой набор переменных](#) (см. раздел 11.4).

9. Укажите ключи и значения для HTTP-заголовков. Вы можете добавить неограниченное количество заголовков.
10. Нажмите **Добавить**.

11.3. Управление сервисами и шаблонами уведомлений по протоколу syslog

Вы можете настроить отправку уведомлений о результатах сканирования объектов и проверки действий по протоколу syslog для интеграции со сторонними системами, обеспечивающими централизованное хранение и обработку событий безопасности.

Полученные уведомления вы можете использовать, например, для нормализации событий информационной безопасности.

В PT CS вы можете настроить подключение к системе, поддерживающей работу по протоколу syslog, или к syslog-серверу, добавив сервис уведомлений, а также добавить один или несколько шаблонов уведомлений для сервиса. Вы можете реализовать интеграцию с несколькими системами, поддерживающими работу по протоколу syslog, добавив для каждой свой сервис уведомлений и набор шаблонов.

Добавление сервиса уведомлений через syslog

► Чтобы добавить сервис уведомлений:

1. В главном меню выберите **Сервисы уведомлений**.
2. Вверху страницы справа выберите кластер, для которого нужен добавить сервис уведомлений.

Примечание. Если при развертывании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные](#) (см. раздел 18), а затем — повторить попытку выбора дочернего кластера.

3. Нажмите **Новый сервис**.
4. Выберите тип **Syslog**.

5. Введите название сервиса уведомлений.
6. Введите адрес и порт сервера syslog в формате <Протокол>://<IP-адрес или доменное имя>:<Порт>. В качестве протокола вы можете указать значение tcp или udp.
7. Нажмите **Подключить**.

Примечание. Вы можете нажать **Подключить без проверки**, если хотите добавить сервис без подключения к серверу syslog. Например, если у вас пока нет доступа к нему.

После добавления сервиса уведомлений по протоколу syslog вам нужно создать один или несколько шаблонов уведомлений, которые будут использованы для формирования содержимого уведомлений. Формат тела уведомления может зависеть от системы, в которую оно будет отправлено. В стандартной конфигурации формат соответствует стандарту RFC 3164.

Добавление шаблона уведомлений через syslog

► Чтобы добавить шаблон уведомлений:

1. В главном меню выберите **Сервисы уведомлений**.
2. Вверху страницы справа выберите кластер, для сервиса уведомлений которого нужно добавить шаблон уведомлений.

Примечание. Если при разворачивании PT CS использовался самоподписанный сертификат, для доступа к дочернему кластеру может потребоваться перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, или [добавить сертификат в доверенные \(см. раздел 18\)](#), а затем — повторить попытку выбора дочернего кластера.

3. Нажмите **+** или **Новый шаблон** в блоке с сервисом уведомлений, для которого нужно добавить шаблон.
4. Введите название шаблона.

Примечание. В поле **Кластер** отображается кластер, для которого будет создан шаблон уведомления. Значение соответствует выбранному в раскрывающемся списке **Кластер** вверху страницы **Сервисы уведомлений** справа.

5. Если требуется, измените адрес центрального кластера.

Примечание. Адрес центрального кластера используется для формирования ссылки на событие в уведомлении. Поле **Адрес центрального кластера** заполняется автоматически. Если доступ к центральному кластеру организован нестандартным способом, например, кластер установлен в закрытом контуре, вы можете указать необходимый адрес для доступа.

6. Выберите тип события, о котором будут отправляться уведомления с помощью этого шаблона.

7. Если требуется, снимите флажок **Отправлять стандартное уведомление**.

Отобразится поле **Тело уведомления**, в котором вы можете добавить шаблон уведомления, используя [синтаксис шаблонизатора Go](#). Для каждого типа событий в PT CS определен [свой набор переменных](#) (см. раздел 11.4).

8. Нажмите **Добавить**.

11.4. Параметры шаблонов уведомлений

Вы можете создать шаблон для каждого уведомления. Для этого в PT CS поддерживаются шаблоны Go, позволяющие использовать переменные.

Вы можете создавать шаблоны уведомлений в различных форматах. Например, в формате JSON — для сервиса уведомлений по webhook и syslog, в формате HTML — для сервиса уведомлений по почте или другом удобном вам формате. Ниже в таблицах перечислены переменные для событий «Ручное сканирование образа», «Сканирование образа на наличие ВПО», «Сканирование конфигурации», «Проверка через контроллер допуска», «Проверка подписи образа» и «Среда выполнения».

Таблица 14. Переменные для события «Ручное сканирование образа»

Имя переменной	Тип данных	Описание
.notificationName	String	Название шаблона уведомлений, для которого создано уведомление
.centralCSURL	String	Адрес центрального кластера
.csClusterID	String	Идентификатор кластера, в котором произошло событие
.csClusterName	String	Имя кластера, в котором произошло событие
.ownCSURL	String	Адрес кластера, в котором произошло событие. Для центрального кластера значение будет совпадать со значением .centralCSURL
.event.GetImageName	String	Имя отсканированного образа
.event.GetBlock	Boolean	Был ли заблокирован процесс CI/CD
.event.GetSeverity	String	Уровень опасности образа
.event.GetRuleName	String	Название правила, при срабатывании которого отправляется уведомление
.event.GetRegisteredAt	Google protobuf timestamp	Время события

Имя переменной	Тип данных	Описание
<code>.event.GetComponents</code>	Array	Список объектов типа <code>Component</code> со следующей структурой: — <code>GetName</code> — имя компонента; — <code>GetVersion</code> — версия компонента; — <code>GetVulnerabilities</code> — список объектов типа <code>Vulnerability</code>
<code>.event.GetComponents → GetVulnerabilities</code>	String	Список объектов типа <code>Vulnerability</code> со следующей структурой: — <code>GetCVE</code> — идентификатор уязвимости; — <code>GetSeverity</code> — уровень опасности уязвимости

Пример уведомления о событии «Ручное сканирование образа»:

```

Просканирован образ {{.event.GetImageName}}.
Установленный уровень опасности: {{.event.GetSeverity}}.
Образ {{if not .event.GetBlock}}не{{end}} заблокирован.
Время сканирования: {{.event.GetRegisteredAt.AsTime.Format "2006-01-02 15:04:05"}}.
{{if gt (len .event.GetComponents) 0}}Список компонентов образа:{{end}}
{{range $component := .event.GetComponents}}
    Компонент {{$component.GetName}}:{{$component.GetVersion}}.
    {{range $vuln := $component.GetVulnerabilities}}
        Обнаружена уязвимость {{$vuln.GetCve}}. Уровень опасности:
        {{$vuln.GetSeverity}}.
    {{end}}
{{end}}
Перейти к событию: {{.centralCSURL}}/scanner/{{.event.GetEventId}}?
clusterUrl={{.ownCSURL}}
Уведомление было отправлено в результате срабатывания правила {{.event.GetRuleName}}.
Вы получили его, потому что включены в рассылку {{.notificationName}}.

```

Таблица 15. Переменные для события «Сканирование образа на наличие ВПО»

Имя переменной	Тип данных	Описание
<code>.notificationName</code>	String	Название шаблона уведомлений, для которого создано уведомление
<code>.centralCSURL</code>	String	Адрес центрального кластера
<code>.csClusterID</code>	String	Идентификатор кластера, в котором произошло событие

Имя переменной	Тип данных	Описание
<code>.csClusterName</code>	String	Имя кластера, в котором произошло событие
<code>.ownCSURL</code>	String	Адрес кластера, в котором произошло событие. Для центрального кластера значение будет совпадать со значением <code>.centralCSURL</code>
<code>.event.GetImageName</code>	String	Имя отсканированного образа
<code>.event.GetBlock</code>	Boolean	Был ли заблокирован процесс CI/CD
<code>.event.GetVerdict</code>	String	Уровень опасности образа
<code>.event.GetRuleName</code>	String	Название сработавшего правила
<code>.event.GetRegisteredAt</code>	Google protobuf timestamp	Время события
<code>.event.GetEventId</code>	String	UUID события
<code>.event.GetFiles</code>	Array of JSON objects	Список объектов типа <code>file</code> со следующей структурой: — <code>GetFilePath</code> — путь к файлу внутри контейнера (например, <code>/usr/bin/evil</code>); — <code>GetFileSha256</code> — хеш SHA-256 содержимого файла; — <code>GetMalwares</code> — список обнаруженного в файле ВПО
<code>.event.GetFiles</code> → <code>file.GetMalwares</code>	Array of JSON objects	Список объектов типа <code>malware</code> со следующей структурой: — <code>GetVerdict</code> — уровень опасности обнаруженного ВПО; — <code>GetMalwareType</code> — тип вредоносного кода (например, <code>trojan</code>, <code>worm</code>); — <code>GetMalwareName</code> — название ВПО; — <code>GetEngineName</code> — наименование движка детектора (например, <code>ClamAV</code>);

Имя переменной	Тип данных	Описание
		- GetEngineVersion — версия движка детектора, который нашел ВПО; - GetDatabaseVersion — версия базы сигнатур, используемой движком
.event_url	String	Полная ссылка на событие в PT CS

Пример уведомления о событии «Сканирование образа на наличие ВПО»:

```

Просканирован образ {{.event.GetImageName}}.
Установленный уровень опасности: {{.event.GetVerdict}}.
Образ {{if not .event.GetBlock}}не{{end}}заблокирован.
Сработало правило: {{.event.GetRuleName}}.
Время сканирования: {{.event.GetRegisteredAt.AsTime.Format "2006-01-02 15:04:05"}}.

```

```

{{if gt (len .event.GetFiles) 0}}
Список файлов образа:
{{range $i, $file := .event.GetFiles}}
Файл: {{$file.GetFilePath}}
SHA 256: {{$file.GetFileSha256}} {{if gt (len $file.GetMalwares) 0}}
Обнаружено ВПО:
{{range $j, $mw := $file.GetMalwares}} — verdict: {{$mw.GetVerdict}}, тип:
{{$mw.GetMalwareType}}, название: {{$mw.GetMalwareName}}, движок:
{{$mw.GetEngineName}} (версия {{$mw.GetEngineVersion}}), база данных:
{{$mw.GetDatabaseVersion}} {{if not (eq $j (sub (len $file.GetMalwares) 1))}}
{{end}}
{{end}}{{end}}
{{end}}
{{else}}
ВПО в образе не обнаружено.
{{end}}

```

```

{{if .centralCSURL}}
Ссылка на центральный PTCS: {{.centralCSURL}}
{{end}}
{{if .csClusterID}}
ID кластера PTCS: {{.csClusterID}}
{{end}}
{{if .csClusterName}}
Имя кластера PTCS: {{.csClusterName}}
{{end}}
{{if .ownCSURL}}
Ссылка на ваш PTCS кластер: {{.ownCSURL}}
{{end}}

```

```
{{if and .centralCSURL .event.GetEventId}}
```

```
Перейти к событию: {{.centralCSURL}}/scanner/{{.event.GetEventId}}{{if  
ne .ownCSURL .centralCSURL}}?clusterUrl={{.ownCSURL}}{{end}}  
{{end}}
```

```
Идентификатор события: {{.event.GetEventId}}
```

```
Уведомление было отправлено в результате срабатывания правила {{.event.GetRuleName}}.  
Вы получили его, потому что подписаны на рассылку {{.notificationName}}.
```

Таблица 16. Переменные для события «Сканирование конфигурации»

Имя переменной	Тип данных	Описание
.notificationName	String	Название шаблона уведомлений, для которого создано уведомление
.centralCSURL	String	Адрес центрального кластера
.csClusterID	String	Идентификатор кластера, в котором произошло событие
.csClusterName	String	Имя кластера, в котором произошло событие
.ownCSURL	String	Адрес кластера, в котором произошло событие. Для центрального кластера значение будет совпадать со значением .centralCSURL
.event.GetEventId	String	UUID события сканирования
.event.GetIacPath	String	Путь к отсканированному файлу с конфигурацией
.event.GetResourceType.String	String	Тип отсканированной конфигурации (например, Dockerfile, Kubernetes, Helm)
.event.GetDigest	String	Контрольная сумма (SHA) отсканированной конфигурации
.event.GetBlock	Boolean	Был ли заблокирован процесс CI/CD
.event.GetSeverity	String	Уровень опасности конфигурации
.event.GetRuleName	String	Название правила, при срабатывании которого отправляется уведомление
.event.GetRegisteredAt	Google protobuf timestamp	Время события

Имя переменной	Тип данных	Описание
<code>.event.GetComponents</code>	JSON object	Список объектов типа <code>Component</code> со следующей структурой: — <code>GetName</code> — имя компонента; — <code>GetVersion</code> — версия компонента; — <code>GetVulnerabilities</code> — список объектов типа <code>Vulnerability</code>
<code>.event.GetComponents → GetVulnerabilities</code>	Array of JSON objects	Список объектов типа <code>Vulnerability</code> со следующей структурой: — <code>GetCheckId</code> — идентификатор проверки, которую не прошла конфигурация; — <code>GetSeverity</code> — уровень опасности уязвимости; — <code>GetCategory</code> — категория проверки (например, сеть, приватность, безопасность); — <code>GetDescription</code> — краткое описание уязвимости

Пример уведомления о событии «Сканирование конфигурации»:

```
<!doctype html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
</head>
<body>
```

```

<h4>Сработало правило: <b>{{.event.GetRuleName}}.</b></h4>
<p>Конфигурации <code>{{.event.GetIacPath}}</code> присвоен {{.event.GetSeverity |
translate}} уровень опасности.</p>
{{if .event.GetBlock}}
  <b>Конфигурация заблокирована.</b>
{{end}}
<p>Время: <code>{{.event.GetRegisteredAt.AsTime.Format "02 Jan 06 15:04 MST"}}</
code></p>
{{if .event.GetComponents}}
  <h4>Компоненты конфигурации:</h4>
  <ul>
    {{range $comp := .event.GetComponents}}
      <li><b>{{$comp.GetName}}</b>{{if $comp.GetVulnerabilities}} содержит
уязвимости:<{{end}}</li>
      <ul>
        {{range $vuln := $comp.GetVulnerabilities}}
          <li><b>{{$vuln.GetCheckId}}:</b> {{ $vuln.GetDescription}}</li>
        {{end}}
      </ul>
    {{end}}
  </ul>
{{end}}
</p>
<a href="{{.centralCSURL}}/events/{{.event.GetEventId}}?
clusterUrl={{.ownCSURL}}">Перейти к событию</a>
</p>
</body>
</html>

```

Таблица 17. Переменные для события «Проверка через контроллер допуска»

Имя переменной	Тип данных	Описание
.notificationName	String	Название шаблона уведомлений, для которого создано уведомление
.centralCSURL	String	Адрес центрального кластера
.csClusterID	String	Идентификатор кластера, в котором произошло событие
.csClusterName	String	Имя кластера, в котором произошло событие
.ownCSURL	String	Адрес кластера, в котором произошло событие. Для центрального кластера значение будет совпадать со значением .centralCSURL
.event.GetEventId	String	UUID события проверки

Имя переменной	Тип данных	Описание
<code>.event.GetRuleName</code>	String	Название правила, при срабатывании которого отправляется уведомление
<code>.event.GetBlock</code>	Boolean	Был ли заблокирован запрос к API-серверу Kubernetes. Примечание. Запрос блокируется, если в нем были обнаружены угрозы или уязвимости
<code>.event.GetSeverity</code>	String	Уровень опасности действия
<code>.event.GetRegisteredAt</code>	Google protobuf timestamp	Время события
<code>.event.GetComponents</code>	Array of JSON objects	Список объектов типа <code>Component</code> со следующей структурой: — <code>GetName</code> — имя компонента; — <code>GetVersion</code> — версия компонента; — <code>GetVulnerabilities</code> — список объектов типа <code>Vulnerability</code>. Содержит один или несколько элементов, если в рамках проверки (AdmissionReview) была просканирована конфигурация
<code>.event.GetComponents → GetVulnerabilities</code>	Array of JSON objects	Список объектов типа <code>Vulnerability</code> со следующей структурой: — <code>GetCheckId</code> — идентификатор проверки, которую не прошла конфигурация; — <code>GetSeverity</code> — уровень опасности уязвимости; — <code>GetCategory</code> — категория проверки (например, сеть, приватность, безопасность); — <code>GetDescription</code> — краткое описание уязвимости

Имя переменной	Тип данных	Описание
<code>.event.GetActions</code>	Array of JSON objects	Список объектов типа <code>Action</code> (в общем случае состоит из одного элемента) со следующей структурой: — <code>GetNamespace</code> — пространство имен, в котором было совершено действие; — <code>GetRequestUid</code> — идентификатор запроса от API-сервера к admission controller. Подробную информацию о запросе AdmissionReview вы можете прочесть в документации Kubernetes; — <code>GetOperation</code> — тип действия. Возможные значения: CREATE, UPDATE, CONNECT; — <code>GetResource.GetGroup</code> — API-группа, к которой принадлежит ресурс, с которым совершается действие. Подробную информацию об API-группах вы можете прочесть в документации Kubernetes; — <code>GetResource.GetVersion</code> — версия API, описывающая ресурс, над которым совершается действие. Подробную информацию о версиях API вы можете прочесть в документации Kubernetes; — <code>GetResource.GetResource</code> — название ресурса, над которым совершается действие; — <code>GetRequestResource.GetGroup</code> — API-группа, указанная в оригинальном запросе к API-серверу. В общем случае будет содержать то же значение, что и <code>GetResource.GetGroup</code>;

Имя переменной	Тип данных	Описание
		Примечание. В отдельных случаях значения могут отличаться в зависимости от настройки <code>ValidatingWebhookConfiguration</code>. Подробную информацию вы можете прочесть в документации Kubernetes. — <code>GetRequestResource.GetVersion</code> — версия API, указанная в оригинальном запросе к API-серверу. В общем случае будет содержать то же значение, что и <code>GetResource.GetVersion</code>; Примечание. В отдельных случаях значения могут отличаться в зависимости от настройки <code>ValidatingWebhookConfiguration</code>. Подробную информацию вы можете прочесть в документации Kubernetes. — <code>GetRequestResource.GetResource</code> — название ресурса, указанное в оригинальном запросе к API-серверу. В общем случае будет содержать то же значение, что и <code>GetResource.GetResource</code>; Примечание. В отдельных случаях значения могут отличаться в зависимости от настройки <code>ValidatingWebhookConfiguration</code>. Подробную информацию вы можете прочесть в документации Kubernetes. — <code>GetSubresource</code> — субресурс, более конкретно описывающий совершаемое действие. Может быть пустым. Каждый ресурс имеет собственный набор субресурсов. Подробную

Имя переменной	Тип данных	Описание
		информацию о субресурсах вы можете прочесть в документации Kubernetes; — <code>GetRequestSubresource</code> — субресурс, указанный в оригинальном запросе к API-серверу. В общем случае будет содержать то же значение, что и <code>GetSubresource</code>; Примечание. В отдельных случаях значения могут отличаться в зависимости от настройки <code>ValidatingWebhookConfiguration</code>. Подробную информацию вы можете прочесть в документации Kubernetes. — <code>GetUserInfo.Username</code> — имя пользователя кластера Kubernetes, который совершает действие; — <code>GetUserInfo.Uid</code> — идентификатор пользователя кластера Kubernetes, который совершает действие; — <code>GetThreats</code> — список угроз, обнаруженных в процессе проверки (<code>AdmissionReview</code>)
<code>.event.GetActions → GetThreats</code>	Array of JSON objects	Список объектов типа <code>Threat</code> со следующей структурой: — <code>GetDetectorId</code> — идентификатор детектора, обнаружившего угрозу; — <code>GetDetectorName</code> — название детектора, обнаружившего угрозу;

Имя переменной	Тип данных	Описание
		— <code>GetDetectorVersion</code> — версия детектора, обнаружившего угрозу; — <code>GetSeverity</code> — уровень опасности угрозы

Пример уведомления о событии «Проверка через контроллер допуска»:

```
<!doctype html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
</head>
<body>
```

```

<h3>Сработало правило: <b>{{.event.GetRuleName}}.</b></h4>
<p>Время: <code>{{.event.GetRegisteredAt.AsTime.Format "02 Jan 06 15:04 MST"}}</code></p>
{{ $action := index .event.GetActions 0 }}
<p>Идентификатор AdmissionReview: <code>{{ $action.GetRequestUid }}</code>.</p>
{{ if .event.GetBlock }}
    <p>Действие заблокировано.</p>
{{ end }}
{{ $resource := $action.GetRequestResource }}
<p>
    Совершено действие <code>{{ $action.GetOperation }}</code> пользователем
    <code>{{ $action.GetUserInfo.Username }}</code>
    в отношении ресурса <code>{{ $resource.GetGroup }}/{{ $resource.GetVersion }}/
    {{ $resource.GetResource }}{{ if $action.GetResourceName }}/{{ $action.GetResourceName }}
    {{ end }}</code>
    {{ if $action.GetNamespace }} в пространстве имен <code>{{ $action.GetNamespace }}</code>
    <code>{{ end }}.
</p>
{{ if $action.GetThreats }}
    <h4>Обнаружены угрозы:</h4>
    <ul>
        {{ range $t := $action.GetThreats }}
            <li><b>{{ $t.GetDetectorName }} ({{ $t.GetDetectorId }}):</b>
            <b>{{ $t.GetDetectorDescription }}</b></li>
        {{ end }}
    </ul>
{{ end }}
{{ if .event.GetComponents }}
    <h4>Компоненты конфигурации:</h4>
    <ul>
        {{ range $comp := .event.GetComponents }}
            <li><b>{{ $comp.GetName }}</b>{{ if $comp.GetVulnerabilities }} содержит
            уязвимости:<code>{{ end }}</li>
            <ul>
                {{ range $vuln := $comp.GetVulnerabilities }}
                    <li><b>{{ $vuln.GetCheckId }}</b> {{ $vuln.GetDescription }}</li>
                {{ end }}
            </ul>
        {{ end }}
    </ul>
{{ end }}
<p>
    <a href="{{ .centralCSURL }}/events/{{ .event.GetEventId }}?
    clusterUrl={{ .ownCSURL }}">Перейти к событию</a>
</p>
</body>

```

Таблица 18. Переменные для события «Среда выполнения»

Имя переменной	Тип данных	Описание
<code>.notificationName</code>	String	Название шаблона уведомлений, для которого создано уведомление
<code>.centralCSURL</code>	String	Адрес центрального кластера
<code>.csClusterID</code>	String	Идентификатор кластера, в котором произошло событие
<code>.csClusterName</code>	String	Имя кластера, в котором произошло событие
<code>.ownCSURL</code>	String	Адрес кластера, в котором произошло событие. Для центрального кластера значение будет совпадать со значением <code>.centralCSURL</code>
<code>.event.GetSeverity</code>	String	Уровень опасности действия
<code>.event.GetRegisteredAt</code>	Google protobuf timestamp	Время события
<code>.event.GetBlock</code>	Boolean	Было ли заблокировано событие
<code>.event.GetRuleName</code>	String	Название правила, при срабатывании которого отправляется уведомление
<code>.event.GetEventId</code>	String	UUID события
<code>.event.GetEvent</code>	String	Объект типа <code>Event</code>, включающий параметры типов <code>String</code>, <code>*UInt32</code>, <code>[]String</code>. Параметры типа <code>String</code>: — <code>GetEventType</code> — тип события. Возможные значения: <code>PROCESS_EXEC</code>, <code>PROCESS_EXIT</code>, <code>PROCESS_KPROBE</code>, <code>PROCESS_TRACEPOINT</code>, <code>PROCESS_LOADER</code>, <code>PROCESS_UPROBE</code>; — <code>GetPodNamespace</code> — пространство имен, в котором было зарегистрировано событие; — <code>GetPodName</code> — название пода, в котором было зарегистрировано событие;

Имя переменной	Тип данных	Описание
		— <code>GetContainerName</code> — название контейнера, в котором было зарегистрировано событие; — <code>GetContainerId</code> — идентификатор контейнера, в котором было зарегистрировано событие; — <code>GetContainerImage</code> — образ, из которого был запущен контейнер; — <code>GetFunctionName</code> — название функции, перехваченной с помощью механизма kprobes. Будет заполнено, если тип события — <code>PROCESS_KPROBE</code>; — <code>GetProcessBinary</code> — название исполняемого файла; — <code>GetProcessArguments</code> — аргументы, с которыми был запущен исполняемый файл; — <code>GetNodeName</code> — название узла кластера, в котором было зарегистрировано событие; — <code>GetParentBinary</code> — исполняемый файл родительского процесса; — <code>GetParentArguments</code> — аргументы родительского процесса; — <code>GetThreats</code> — список угроз, обнаруженных детекторами
	<code>*UInt32</code>	Параметры типа <code>*UInt32</code> : — <code>ProcessPid</code> — PID процесса; — <code>ProcessUid</code> — UID процесса; — <code>ParentPid</code> — PID родительского процесса;

Имя переменной	Тип данных	Описание
		— <code>ParentUid</code> — UID родительского процесса; — <code>ProcessSetuid</code> — эффективный UID процесса. Принимает значение, отличное от <code>NULL</code>, если у исполняемого файла установлен флаг доступа <code>setuid</code>; — <code>ProcessSetgid</code> — эффективный GID процесса. Принимает значение, отличное от <code>NULL</code>, если у исполняемого файла установлен флаг доступа <code>setgid</code>
	<code>[]String</code>	Параметры типа <code>[]String</code> : — <code>GetProcessCapEffective</code> — список эффективных привилегий (<code>effective capabilities</code>) процесса; — <code>GetProcessCapPermitted</code> — список разрешенных привилегий (<code>permitted capabilities</code>) процесса
<code>.event.GetEvent → GetThreats</code>	<code>String</code>	Список объектов типа <code>Threat</code> со следующей структурой: — <code>GetDetectorId</code> — идентификатор детектора, обнаружившего угрозу; — <code>GetDetectorName</code> — имя детектора, обнаружившего угрозу; — <code>GetDetectorVersion</code> — версия детектора, обнаружившего угрозу; — <code>GetDetectorDescription</code> — описание детектора, обнаружившего угрозу; — <code>GetSeverity</code> — уровень опасности угрозы

Пример уведомления о событии «Среда выполнения»:

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="UTF-8" />
  </head>
  <body>
    <h4>Сработало правило: <b>{{.event.GetRuleName}}.</b></h4>
    <p>
      Время:
      <code>{{.event.GetRegisteredAt.AsTime.Format "02 Jan 06 15:04 MST"}}</code>
    </p>
    {{ $event := .event.GetEvent }}
    <p>
      Выявлена подозрительная активность при запуске процесса
      <code>{{ $event.GetEvent.GetProcessBinary }}
      {{ $event.GetEvent.GetProcessArguments }}</code> в поде
      <code>{{ $event.GetPodNamespace }}.{{ $event.GetPodName }}</code> на узле
      <code>{{ $event.GetNodeName }}</code>.
    </p>
    <h4>Выявлены угрозы:</h4>
    <ul>
      {{ range $t := $event.GetThreats }}
        <li>
          <b>{{ $t.GetDetectorName }} ({{ $t.GetDetectorId }})</b>:
          {{ $t.GetDetectorDescription }}
        </li>
      {{ end }}
    </ul>
    {{ if .event.GetBlock }}
      <p>Событие заблокировано.</p>
    {{ end }}
    <p>
      Для проведения расследования перейдите в:
      <a href="https://ptcs.example.com/runtime/events">Консоль</a>
    </p>
    <p>
      <a href="{{.centralCSURL}}/events/{{.event.GetEventId}}?clusterUrl={{.ownCSURL}}">Перейти к событию</a>
    </p>
  </body>
</html>
```

Внимание! При использовании

параметров `.event.GetEvent.ProcessPid`, `.event.GetEvent.ProcessUid`, `.event.GetEvent.ParentPid`, `.event.GetEvent.ParentUid`, `.event.GetEvent.ProcessSetuid`, `.event.G`

`etEvent.ProcessSetgid` в шаблоне уведомления для `syslog` и `webhook` вам нужно реализовать проверку наличия значения для них. Иначе уведомления по такому шаблону формироваться не будут.

Пример проверки наличия значения:

```
{{if $args := $event.GetFunctionArgs}}
  "function_args": {{$args}},
{{end}}
```

Таблица 19. Переменные для события «Проверка подписи образов»

Имя переменной	Тип данных	Описание
<code>.notificationName</code>	String	Название шаблона уведомлений, для которого создано уведомление
<code>.centralCSURL</code>	String	Адрес центрального кластера
<code>.csClusterID</code>	String	Идентификатор кластера, в котором произошло событие
<code>.csClusterName</code>	String	Имя кластера, в котором произошло событие
<code>.ownCSURL</code>	String	Адрес кластера, в котором произошло событие. Для центрального кластера значение будет совпадать со значением <code>.centralCSURL</code>
<code>.event.GetBlock</code>	String	Был ли заблокирован процесс CI/CD
<code>.event.GetRuleName</code>	String	Название правила, при срабатывании которого отправляется уведомление
<code>.event.GetRegisteredAt</code>	Google protobuf timestamp	Время события
<code>.event.GetAction</code>	JSON object	Список объектов типа <code>Action</code> со следующей структурой: — <code>GetImages</code> — список объектов типа <code>Image</code>; — <code>GetNamespace</code> — пространство имен, в котором было совершено действие;

Имя переменной	Тип данных	Описание
		— <code>GetOperation</code> — тип действия. Возможные значения: <code>CREATE</code>, <code>UPDATE</code>, <code>CONNECT</code>; — <code>GetResourceName</code> — название ресурса, над которым совершается действие
<code>.event.GetAction</code> → <code>GetImages</code>	Array of JSON objects	Список объектов <code>Image</code> со следующей структурой: — <code>GetRegistry</code> — реестр, в котором хранится проверенный образ; — <code>GetImageName</code> — имя проверенного образа

Пример уведомления о событии «Проверка подписи образа»:

```
<!doctype html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>PT CS: инцидент при проверке подписи образа</title>
</head>
<body>
```

```

<h3>Инцидент</h3>
<h5>при проверке подписи образа</h5>
<h5>{{.event.GetRegisteredAt.AsTime.Format "02.01.2006 15:04:05"}}</h5>
<table>
  <tr>
    <td>
      <table>
        <tr>
          <td>
            <span>Нет или недействительна</span>
          </td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td>
      <table>
        <tr>
          <td><span>Операция: {{.event.GetAction.GetOperation}}</span></td>
        </tr>
        <tr>
          <td><span>Пространство имен: {{.event.GetAction.GetNamespace}}</span></td>
        </tr>
        <tr>
          <td><span>Ресурс: {{.event.GetAction.GetResourceName}}</span></td>
        </tr>
        {{with .event.GetAction.GetImages}}
          <tr>
            <td><span>Образы:</span></td>
          </tr>
          {{range $img := .}}
            <tr>
              <td>
                <span>{{if $img.GetRegistry}}{{$img.GetRegistry}}</span>
                <span>{{if $img.GetImageName}}</span>
              </td>
            </tr>
          {{end}}
        {{end}}
      </table>
    </td>
  </tr>
  <tr>
    <td>
      <table>

```

```

        <tr>
            <td>
                <span>Отправлено уведомление</span>
            </td>
        </tr>
        {{if .event.GetBlock}}
        <tr>
            <td>
                <span>Действие заблокировано</span>
            </td>
        </tr>
        {{end}}
    </table>
</td>
</tr>
</table>
{{if and .centralCSURL .event.GetEventId}}
    <div>
        <a href="{{.centralCSURL}}/events/{{.event.GetEventId}}{{if
ne .ownCSURL .centralCSURL}}?clusterUrl={{.ownCSURL}}{{end}}">
            Перейти к событию
        </a>
    </div>
{{end}}
</body>
</html>

```

12. Интеграция с продуктами Positive Technologies

Раздел содержит информацию об интеграции PT CS с другими системами компании Positive Technologies, а также инструкции по настройке работы PT CS с этими системами и сценарии работы при интеграции.

В этом разделе

[Интеграция PT CS с PT Sandbox и PT MultiScanner \(см. раздел 12.1\)](#)

[Интеграция PT CS с MaxPatrol SIEM \(см. раздел 12.2\)](#)

12.1. Интеграция PT CS с PT Sandbox и PT MultiScanner

В этом разделе указана информация об интеграции PT CS с PT Sandbox и PT MultiScanner. Такая интеграция позволяет выполнять в PT CS сканирование образов на наличие ВПО.

Сценарий работы при интеграции PT CS с PT Sandbox и PT MultiScanner

Сканирование образа на наличие ВПО выполняется только с использованием утилиты командной строки `ptcs-cli`. Сканирование осуществляется по следующему алгоритму:

1. Утилита `ptcs-cli` отправляет запрос на сканирование сервису `Image scanner`.
2. Сервис `Image scanner`, получив запрос, загружает из реестра образ и передает его на сканирование в PT Sandbox или PT MultiScanner.
3. PT Sandbox или PT MultiScanner возвращает сервису `Image scanner` идентификатор (ID) задания на сканирование, который `Image scanner` передает утилите `ptcs-cli`.
4. Утилита `ptcs-cli` запрашивает результат сканирования у сервиса `Image scanner` до тех пор, пока не получит его или не выйдет время, установленное в значении параметра `scan-timeout` (по умолчанию 15 минут). Запрос отправляется с периодичностью, определенной значением параметра `attempt-interval` (по умолчанию 10 секунд).
5. Сервис `Image scanner` запрашивает результат сканирования у PT Sandbox или PT MultiScanner. Как только PT Sandbox или PT MultiScanner возвращает результат сканирования, он передается сервису `history API`, и PT CS отправляет уведомление с результатами сканирования.

Настройка PT Sandbox или PT MultiScanner для интеграции с PT CS

Для интеграции PT CS с PT Sandbox версий ниже 5.35 или PT MultiScanner версий ниже 5.35 необходимо:

1. В PT Sandbox или PT MultiScanner выпустить токен доступа к публичному API с разрешенным действием **Проверка с параметрами источника**.
2. Добавить в PT Sandbox или PT MultiScanner источник с типом **API с выбранными параметрами проверки**.

Для интеграции PT CS с PT Sandbox версии 5.35 и выше или PT MultiScanner версии 5.35 и выше необходимо:

1. В PT Sandbox или PT MultiScanner выпустить токен доступа к публичному API с разрешенным действием **Проверка с параметрами источника**.
2. Добавить в PT Sandbox или PT MultiScanner источник с типом **PT CS**.

После добавления источника вам нужно настроить параметры проверки объектов от источника.

► Чтобы настроить параметры проверки объектов от источника:

1. В главном меню выберите **Источники**.
2. В панели с информацией о добавленном источнике перейдите по ссылке **Параметры проверки**.
3. В поле **Максимальное время ожидания результатов проверки в синхронном режиме** введите значение **1200**.
4. В поле **Максимальное время выполнения задания** введите значение **1600**.
5. Отключите использование тестовых YARA-правил.
6. Выберите глубину распаковки архивов.

Примечание. Глубина распаковки архивов определяет максимальное количество вложенных друг в друга архивов, которые будут распаковываться при проверке. Если распаковывать архивы не требуется, вы можете выбрать **0**, тогда архивы будут проверяться как обычные файлы. Увеличение глубины распаковки архивов снижает скорость проверки.

7. Если архивы, для которых не удалось подобрать пароль, требуется определять как потенциально опасные, в блоке **Определять как потенциально опасные** установите флажок **Зашифрованные и нераспакованные архивы**.
8. Если архивы, вложенность в которых превышает значение параметра **Глубина распаковки архивов**, требуется определять как потенциально опасные, установите флажок **Архивы с превышенной глубиной распаковки**.
9. Отключите извлечение ссылок из объектов.

10. Отключите сохранение файлов без угроз.
11. Отключите поведенческий анализ.
12. Нажмите **Сохранить**.

Подробную инструкцию по настройке PT Sandbox см. в Руководстве администратора PT Sandbox. Подробную инструкцию по настройке PT MultiScanner см. в Руководстве администратора PT MultiScanner.

Для сканирования образов на наличие ВПО после настройки интеграции нужно:

1. [Создать шаблоны уведомлений \(см. раздел 11\)](#) с типом **Сканирование образа на наличие ВПО**.
2. Создать правила реагирования с типом **Сканирование образа на наличие ВПО**. Подробнее о правилах реагирования см. в Руководстве пользователя.
3. Запустить сканирование с помощью команды `scan malware image`. Подробнее о сканировании с помощью утилиты командной строки `ptcs-cli` см. в Руководстве пользователя.

Результаты сканирования вы можете просмотреть в разделе **Сканирования**.

12.2. Интеграция PT CS с MaxPatrol SIEM

В этом разделе указана информация об интеграции PT CS с системой MaxPatrol SIEM.

Вы можете настроить отправку событий среды выполнения от кластеров, защищаемых PT CS, в MaxPatrol SIEM с помощью syslog или через API.

Внимание! События, полученные от добавленных или измененных в [режиме эксперта \(см. раздел 9.1\)](#) источников, в MaxPatrol SIEM будут отнесены к ненормализованным. Такие события не анализируются по правилам корреляции, для них не будут выявлены и зарегистрированы события информационной безопасности.

Настройка PT CS для интеграции с MaxPatrol SIEM

Для настройки отправки событий с помощью syslog нужно в PT CS создать сервис и шаблон уведомлений, а также правило реагирования, при срабатывании которого будет отправляться уведомление.

Для отправки событий по API нужно в PT CS [настроить параметры отслеживания и фильтры для событий среды \(см. раздел 9.2\)](#) выполнения и создать токен доступа к API с привилегией **Просмотр событий среды выполнения**.

Подробную информацию о настройке вы можете найти в Справочном руководстве по настройке источников MaxPatrol SIEM в разделе «Positive Technologies Container Security (PT CS): настройка источника».

Настройка MaxPatrol SIEM для интеграции с PT CS

Для отправки событий стандарта syslog в MaxPatrol SIEM нужно создать и запустить задачу на сбор данных с профилем SysLog.

Для отправки событий PT CS по API необходимо:

1. Добавить учетную запись для доступа к источнику.
2. Создать справочник для сбора событий.
3. Создать профиль для сбора событий.
4. Создать и запустить задачу на сбор событий.

Подробную информацию о настройке вы можете найти в Справочном руководстве по настройке источников MaxPatrol SIEM в разделе «Positive Technologies Container Security (PT CS): настройка MaxPatrol SIEM».

13. Администрирование PT CS

В этом разделе содержатся инструкции по администрированию PT CS.

В этом разделе

[Состав компонентов ПО и его среды функционирования \(см. раздел 13.1\)](#)

[О ролях пользователей \(см. раздел 13.2\)](#)

[Управление дочерними кластерами \(см. раздел 13.3\)](#)

[Управление учетными записями пользователей \(см. раздел 13.4\)](#)

[Мониторинг состояния PT CS \(см. раздел 13.5\)](#)

13.1. Состав компонентов ПО и его среды функционирования

В состав ПО входят следующие компоненты (все компоненты со свободной лицензией):

- реляционная система управления базами данных PostgreSQL для хранения пользовательских данных и результатов сканирования;
- интерпретатор языка программирования Python 3.10 для выполнения кода компонентов приложения;
- консольная утилита syft 1.16.0 для генерации отчетов о компонентах контейнеров;
- расширяемый кросс-платформенный веб-сервер Caddy 2.8.4 с открытым исходным кодом;
- реляционная система управления базами данных ClickHouse для хранения данных, связанных с мониторингом событий среды выполнения и реагированием на них;
- система управления базами данных Redis для хранения очереди задач;
- брокер сообщений RabbitMQ.

Среда функционирования ПО включает:

- контейнеры на базе Debian версии 11;
- инструмент для запуска мультиконтейнерных приложений Kubernetes версии 1.20 и выше.

В таблице ниже приведена информация о лицензиях компонентов ПО.

Таблица 20. Лицензии компонентов ПО

Название компонента	Лицензия	Правообладатель	Ссылка на текст лицензии
PostgreSQL 16.0.0	PostgreSQL	1996–2022, PostgreSQL Global Development Group. 1994, The Regents of the University of California	postgresql.org
Python 3.10	Python 2.0	Python Software Foundation	docs.python.org/3.10/
Syft 1.16.0	Apache License 2.0	Anchore	github.com/anchore/
Caddy 2.8.4	Apache License 2.0	Stack Holdings	github.com/caddyserver/
ClickHouse 23.8.2	Apache License 2.0	ClickHouse, Inc	github.com/ClickHouse/
Redis 7.0.11	The 3-Clause BSD License	2006–2020, Salvatore Sanfilippo	github.com/redis/redis/
RabbitMQ 3.12.6	MPL-2.0	VMware, Inc. or its affiliates	rabbitmq.com/
Checkov 2.3.220	Apache License 2.0	Palo Alto Networks	github.com/bridgecrewio/checkov/

13.2. О ролях пользователей

В PT CS используется ролевая модель управления доступом. Пользователь в PT CS может иметь одну из следующих ролей: **Администратор, Специалист по безопасности, Разработчик, Аудитор**. Кроме того, в системе есть роль **CI/CD**, которая предназначена для учетной записи, используемой при встраивании PT CS в процесс непрерывной интеграции и непрерывного развертывания и последующего сканирования объектов. Подробнее о встраивании в процесс непрерывной интеграции и непрерывного развертывания см. в Руководстве пользователя.

Роли пользователей и их привилегии описаны в таблице ниже.

Таблица 21. Роли пользователей и их привилегии

Объекты	Привилегии			
	Администратор	Специалист по безопасности	Разработчик	Аудитор
Список событий, сводка и результаты сканирования объектов на странице Сводка	Просмотр			
Виджеты и результаты проверки на странице Соответствие стандартам	Просмотр			
Проверка кластера на странице Соответствие стандартам	Запуск			—
Карта кластера	Просмотр			
Кластеры	Просмотр, создание, изменение, удаление	Просмотр		
Реестры образов	Добавление, удаление	—		
Список образов и информация о них	Просмотр			
Образы	Сканирование			—
Виджеты, таблица событий и результаты сканирования объектов на странице Сканирования образов	Просмотр			
Приоритет задачи на сканирование	Изменение			Просмотр
Параметры очереди сканирования	Изменение			Просмотр
Таблица событий и результаты проверки объектов на странице Проверка конфигураций	Просмотр			

Объекты	Привилегии			
	Администратор	Специалист по безопасности	Разработчик	Аудитор
Проверки подписи образов	Просмотр, создание, изменение, удаление		Просмотр	
Список событий на вкладке События страницы Контроллер допуска	Просмотр			
Режим эксперта	Включение, отключение	—		
Источники событий среды выполнения (при включенном режиме эксперта)	Создание, изменение, удаление	—		
Источники событий среды выполнения	Просмотр			
Детекторы, отслеживающие события	Просмотр, создание, изменение, удаление		Просмотр	
Списки фильтров событий на странице Среда выполнения	Просмотр, создание, изменение, удаление		Просмотр	
Параметры отслеживания и записи событий среды выполнения	Просмотр, изменение		Просмотр	
События среды выполнения	Просмотр			
Правила реагирования	Просмотр, создание, изменение, удаление		Просмотр	
Учетные записи пользователей	Просмотр, создание, изменение, удаление	Просмотр, удаление учетных записей пользователей с любой ролью; создание, изменение учетных записей пользователей с ролью Специалист по безопасности	—	

Объекты	Привилегии			
	Администратор	Специалист по безопасности	Разработчик	Аудитор
Сервисы уведомлений	Просмотр, со- здание, измене- ние, удаление	Просмотр		
Шаблоны для сервисов уведомлений	Просмотр, создание, изменение, удаление		Просмотр	
Параметры системы	Просмотр, со- здание, измене- ние, удаление	Просмотр		
Токены доступа	Просмотр, со- здание, отзыв, отзыв всех токе- нов доступа	Просмотр, создание, отзыв		

Сканирование с использованием консольной утилиты `ptcs-cli` доступно пользователям с ролями **Администратор**, **Специалист по безопасности**, **Разработчик** и **CI/CD**.

Примечание. Пользователю с ролью **CI/CD** доступно только сканирование с использованием консольной утилиты `ptcs-cli`.

13.3. Управление дочерними кластерами

Вы можете использовать PT CS для защиты нескольких кластеров. Первоначально вам нужно установить PT CS в центральном кластере, а затем — подключить к центральному кластеру PT CS дочерние. Это позволяет:

- упростить управление инцидентами в сложной инфраструктуре благодаря единой точке управления;
- обеспечить централизованный мониторинг и реагирование на события ИБ;
- обеспечить максимальное покрытие инфраструктуры.

Центральный кластер — это кластер, к которому подключаются дочерние кластеры и через запросы к которому осуществляется управление ими.

Дочерний кластер — это кластер, в котором развернут экземпляр PT CS, подключенный к центральному кластеру. Количество дочерних кластеров, которые вы можете подключить, ограничено приобретенной [лицензией](#) (см. [раздел 4](#)).

В дочерних кластерах не устанавливаются модули `auth API` и `cluster manager`. Такое решение обеспечивает высокую производительность и снижает сетевую нагрузку. Модуль `auth API` отвечает за управление учетными записями, которые используются для работы со всеми подключенными кластерами. Модуль `cluster-manager` обеспечивает управление подключением и удалением дочерних кластеров.

Управление экземплярами PT CS, установленными в дочерних кластерах, осуществляется в веб-интерфейсе PT CS, установленного в центральном кластере.

Примечание. Вам необходимо обеспечить [сетевое соединение между центральным и дочерними кластерами \(см. раздел 5.1\)](#) для загрузки информации о лицензии и БДУ и проксирования запросов пользователей от центрального кластера к дочерним. При недоступности дочернего кластера центральный не сможет получить данные дочернего и в веб-интерфейсе будет отображена ошибка 404.

Вы можете увидеть все подключенные дочерние кластеры, подключить новые или удалить их на странице **Кластеры**, которая открывается при выборе в главном меню раздела **Кластеры**.

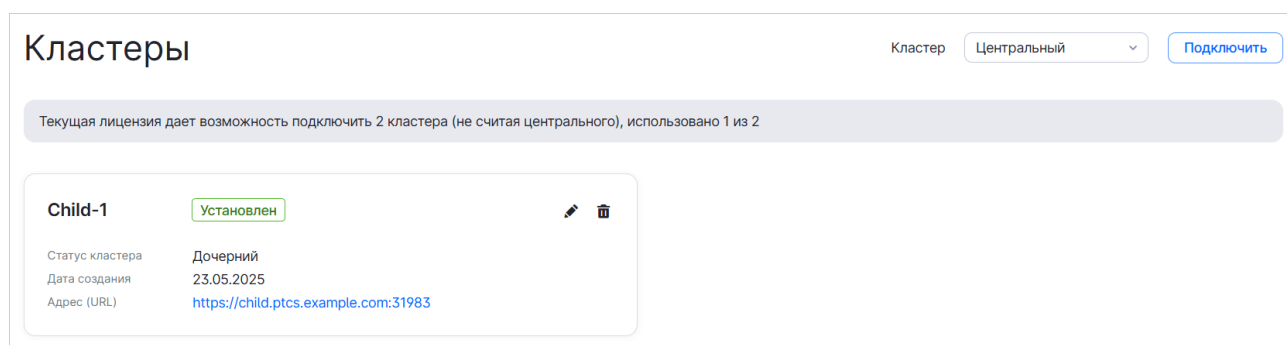


Рисунок 9. Страница **Кластеры**

На странице отображаются:

- раскрывающийся список **Кластер** для выбора кластера, с которым будет осуществляться работа. Выбранное значение будет установлено и на других страницах веб-интерфейса;

Примечание. Вы также можете выбрать кластер в правом верхнем углу всех страниц интерфейса, кроме страницы **Пользователи**.

- кнопка **Подключить** для подключения нового дочернего кластера и получения команды для его установки;
- информация о количестве кластеров, которые можно подключить в соответствии с лицензией, и количестве подключенных кластеров;
- блоки с информацией о подключенных кластерах.

В каждом блоке с информацией о подключенном кластере отображаются:

- имя подключенного кластера;
- статус установки кластера. Кластер может находиться в одном из следующих статусов установки: **Не установлен** и **Установлен**;
- дата создания кластера в веб-интерфейсе;
- статус кластера;
- кнопки для изменения имени кластера и удаления кластера. Если кластер выбран для работы в веб-интерфейсе PT CS, то кнопка удаления для него неактивна.

В этом разделе

[Подключение дочернего кластера к PT CS \(см. раздел 13.3.1\)](#)

[Удаление дочернего кластера \(см. раздел 13.3.2\)](#)

13.3.1. Подключение дочернего кластера к PT CS

Для подключения дочернего кластера вам сначала нужно добавить дочерний кластер в веб-интерфейсе PT CS и получить команду установки. Полученную команду установки нужно выполнить в дочернем кластере, после чего он будет инициализирован в PT CS. Инициализация выполняется автоматически без участия пользователя.

Примечание. В шагах ниже в скобках указаны [параметры скрипта генерации конфигурационного файла Helm-чарта \(см. раздел 5.5\)](#), соответствующие параметрам, задаваемым при подключении дочернего кластера.

Шаг 1. Настройка подключения к реестру с образами PT CS или серверу обновлений update.ptsecurity.com

Вы можете установить PT CS в дочерних кластерах с использованием реестра с образами PT CS или с использованием Helm-чарта, размещенного на сервере обновлений update.ptsecurity.com. В первом случае вам необходимо настроить доступ к реестру с образами PT CS, во втором — настроить доступ к серверу обновлений update.ptsecurity.com. Если вы планируете использовать сервер обновлений update.ptsecurity.com, то вам необходимо заранее [получить логин и пароль из файла лицензии \(см. раздел 5.4\)](#).

- ▶ Чтобы настроить подключение к реестру с образами PT CS или серверу обновлений update.ptsecurity.com:
 1. В главном меню выберите **Кластеры**.
 2. Нажмите **Подключить**.
 3. Выполните одно из следующих действий:
 - Если вы хотите установить PT CS с использованием образов, введите адрес реестра с образами PT CS (`--registry`).
 - Если вы хотите установить PT CS с использованием Helm-чарта, размещенного на сервере обновлений, введите адрес сервера обновлений update.ptsecurity.com.
 4. Введите имя пользователя и пароль для авторизации на сервере с реестром образов PT CS или на сервере обновлений update.ptsecurity.com.
 5. Если образы PT CS хранятся в реестре GitLab, включите использование сокращенных имен образов (`--short-image-names`).

Примечание. При установке с сервера обновлений update.ptsecurity.com использование сокращенных имен образов включать не нужно.

6. Нажмите **Следующий шаг**.

Шаг 2. Настройка подключения к базам данных и сервису RabbitMQ

В зависимости от прогнозируемой нагрузки вы можете развернуть БД ClickHouse, PostgreSQL, Redis и сервис RabbitMQ из дистрибутива PT CS одновременно с установкой самого PT CS или заранее из отдельного дистрибутива. Используемые версии СУБД должны удовлетворять [программным требованиям \(см. раздел 5.1\)](#). Для подключения к СУБД вам могут потребоваться сертификаты. Вы можете использовать уже существующие сертификаты или сгенерировать новые.

- Чтобы настроить подключение к базам данных и сервису RabbitMQ:

1. В блоке параметров **ClickHouse** выполните одно из следующих действий:
 - Если вы хотите развернуть СУБД ClickHouse из установочного дистрибутива PT CS, выберите **Установка в кластере**.
 - Если вы хотите использовать СУБД ClickHouse, развернутую из отдельного дистрибутива на другом сервере, выберите **Внешнее подключение**.
2. Введите имя пользователя и пароль.
3. Если СУБД ClickHouse поддерживает безопасные подключения по протоколу TLS, включите использование TLS.
4. Если требуется проверять безопасность соединения с СУБД ClickHouse с помощью сертификата, включите проверку безопасности соединения.
5. Если вы выбрали внешнее подключение и включили проверку безопасности соединения, введите текст сертификата.

Примечание. Если выбрана установка в кластере и включена проверка безопасности соединения, то для проверки безопасности соединения будет использован уже имеющийся сертификат PT CS.

6. Если вы выбрали установку в кластере и хотите сохранять данные даже после перезагрузки сервера, включите использование persistent storage.

Примечание. Вы можете ввести название существующего класса хранилища, который будет использоваться для хранения данных СУБД ClickHouse (`--clickhouse-storage-class`). Если он не указан, будет использован класс хранилища по умолчанию.

7. Если вы выбрали внешнее подключение, введите адрес сервера для подключения к СУБД ClickHouse (`--clickhouse`).
8. Если вы выбрали внешнее подключение, введите имя БД ClickHouse, которая будет использоваться для хранения данных кластера (`--clickhouse-name`).

Примечание. Если для дочерних кластеров будут использоваться те же экземпляры СУБД ClickHouse и PostgreSQL, что и для центрального, вам нужно создать отдельные базы данных для каждого дочернего кластера. Если для каждого кластера планируется использовать отдельные экземпляры СУБД ClickHouse и PostgreSQL, вы можете задать одинаковые имена БД для разных кластеров.

9. В блоке параметров **PostgreSQL** настройте подключение к СУБД PostgreSQL по аналогии с настройкой подключения к СУБД ClickHouse.
10. В блоке параметров **Redis** настройте подключение к Redis по аналогии с настройкой подключения к СУБД ClickHouse.

Примечание. Для каждого кластера вам нужно использовать отдельный экземпляр сервиса Redis.

11. В блоке параметров **RabbitMQ** выполните одно из следующих действий:
 - Если вы хотите развернуть RabbitMQ из установочного дистрибутива PT CS, выберите **Установка в кластере**.
 - Если вы хотите использовать RabbitMQ, развернутый из отдельного дистрибутива на другом сервере, выберите **Внешнее подключение**.
12. Введите имя пользователя (`--rabbitmq-user`) и пароль.
13. Если вы выбрали установку в кластере и хотите сохранять данные даже после перезагрузки сервера, включите использование persistent storage.

Примечание. Вы можете ввести название существующего класса хранилища, который будет использоваться для хранения данных RabbitMQ (`--rabbitmq-storage-class`). Если он не указан, будет использован класс хранилища по умолчанию.

14. Если вы выбрали внешнее подключение, введите адрес RabbitMQ (`--rabbitmq`).

Примечание. Для каждого кластера вам нужно использовать отдельный экземпляр брокера сообщений RabbitMQ.

15. Нажмите **Следующий шаг**.

Шаг 3. Настройка доступа к Grafana и Prometheus для мониторинга

Если вы хотите подключить дочерний кластер без настройки мониторинга, вы можете пропустить этот шаг с помощью кнопки **Следующий шаг**.

► Чтобы настроить мониторинг в дочернем кластере:

1. Подключите мониторинг.
2. Выполните одно из следующих действий:
 - Если вы хотите развернуть Grafana и Prometheus из установочного дистрибутива PT CS, выберите **Установка в кластере**.

- Если вы хотите использовать Grafana и Prometheus, развернутые из отдельных дистрибутивов на других серверах, выберите **Внешнее подключение**.
3. Выполните одно из следующих действий:
 - Если вы выбрали установку в кластере, в блоке параметров **Grafana** введите имя пользователя и пароль.
 - Если вы выбрали внешнее подключение, введите адрес сервера для подключения к Grafana (`--grafana`).
 4. Если вы выбрали установку в кластере и хотите сохранять данные даже после перезагрузки сервера, в блоке параметров **Prometheus** включите использование persistent storage.

Примечание. Вы можете ввести название существующего класса хранилища, который будет использоваться для хранения данных Prometheus (`--prometheus-storage-class`). Если он не указан, будет использован класс хранилища по умолчанию.

5. Нажмите **Следующий шаг**.

Шаг 4. Настройка подключения к веб-интерфейсу

На этом шаге вы можете настроить подключение к веб-интерфейсу через Ingress-контроллер, с помощью сервиса NodePort или используя оба способа. Если вы не настроите доступ к веб-интерфейсу через Ingress-контроллер или с помощью сервиса NodePort, вам нужно обеспечить сетевой доступ к модулю reverse proxy по адресу дочернего кластера. Кроме того, пользователям PT CS необходим сетевой доступ к кластерам, так как запросы из веб-интерфейса отправляются напрямую в дочерние кластеры.

► Чтобы настроить подключение к веб-интерфейсу:

1. Если вы хотите настроить доступ через Ingress-контроллер, в блоке параметров **Ingress** включите доступ к PT CS через Ingress (`--deploy-ingress`).
 2. Введите имя класса Ingress-контроллера (`--ingress-class`).
 3. Введите имя хоста, на котором установлен Ingress-контроллер (`--ingress-hostname`).
 4. Введите SSL- или TLS-сертификат для подключения к хосту с Ingress-контроллером в формате PEM.
- Примечание.** Вы можете использовать уже имеющийся в инфраструктуре сертификат или сгенерировать новый.
5. Введите закрытый ключ сертификата для подключения к хосту с Ingress-контроллером в формате PEM.
 6. Если вы хотите настроить доступ через сервис NodePort, в блоке параметров **NodePort** включите доступ к PT CS через NodePort (`--deploy-nodeport`).

7. Введите порт для подключения к PT CS по протоколу HTTP или gRPC в диапазоне от 30000 до 32767 (--nodeport).
8. Нажмите **Следующий шаг**.

Шаг 5. Настройка общих параметров

► Чтобы настроить общие параметры:

1. Если вы хотите, чтобы блокировались запросы к kube-apiserver при ошибках в работе PT CS, в блоке **Контроллер допуска** включите блокировку действий при ошибках (--admission-webhook-block-on-failure).
2. Если вы хотите, чтобы действия, выполняемые в отдельных пространствах имен, не проверялись через контроллер допуска, введите игнорируемые namespaces (--admission-ignored-namespaces).

Вы можете ввести несколько пространств имен.

3. В блоке **Общие параметры** введите длительность хранения результатов сканирования в часах (--history-retention-interval).
4. Если PT CS используется в закрытом контуре, введите адрес прокси-сервера в формате <Протокол>://<Адрес>:<Порт> (--proxy-url).

В качестве протокола вы можете указать значение http или https.

5. Введите адрес дочернего кластера в формате <Протокол>://<Адрес> (--own-ptcs-url).

В качестве протокола вы можете указать значение http или https.

6. Введите адрес центрального кластера в формате <Протокол>://<Адрес> (--central-ptcs-url).

В качестве протокола вы можете указать значение http или https.

7. Введите пространство имен (namespace), в котором будет выполнена установка PT CS.
8. Введите имя кластера для отображения в веб-интерфейсе PT CS.
9. Нажмите **Получить команду установки**.

Шаг 6. Получение команды установки

После того как все параметры заполнены и нажата кнопка **Получить команду установки**, будет открыта страница с командами установки и кнопкой для скачивания файла `values.yaml` с конфигурационными параметрами Helm-чарта.

Для установки PT CS в дочернем кластере вы можете выполнить команду установки или команду установки YAML. Если вы хотите установить PT CS с помощью команды установки YAML, то вам необходимо сохранить файл `values.yaml` с параметрами Helm-чарта в том же репозитории, в котором размещены образы PT CS.

► Чтобы получить команду установки:

1. Выполните одно из следующих действий:
 - Скопируйте команду установки.
 - Скопируйте команду установки YAML.
2. Если вы скопировали команду установки YAML, то нажмите **Скачать values.yaml**.
3. Нажмите **Завершить**.

Шаг 7. Установка в дочернем кластере

Вы можете установить PT CS с помощью команды установки.

► Чтобы установить в дочернем кластере с помощью команды установки:

1. Подключитесь к кластеру, в котором нужно установить PT CS.
2. Выполните команду установки.

Примечание. Если образы PT CS и Helm-чарт опубликованы в разных реестрах или используется локальный архив Helm-чарта, то вам необходимо изменить путь к Helm-чарту в команде установки YAML вручную. Образы PT CS и Helm-чарт [могут быть опубликованы в реестре \(см. раздел 5.6\)](#) при установке в центральном кластере.

После установки PT CS дочерний кластер должен пройти инициализацию. Инициализация включает получение лицензии и БДУ из центрального кластера и регистрацию в нем. Эти действия выполняются без участия пользователя. После успешной инициализации статус установки на странице **Кластеры** в веб-интерфейсе изменится на **Подключен**. Если статус не изменяется дольше 30 минут, [проанализируйте журналы событий компонента PTCS manager \(см. раздел 18\)](#).

Вы также можете установить PT CS с помощью команды установки YAML.

► Чтобы установить PT CS в дочернем кластере с помощью команды установки YAML:

1. Подключитесь к дочернему кластеру, в котором нужно установить PT CS.
2. Перейдите в каталог, в котором сохранен файл `values.yaml`.
3. Выполните команду установки YAML.

Примечание. Если образы PT CS и Helm-чарт опубликованы в разных реестрах или используется локальный архив Helm-чарта, то вам необходимо изменить путь к Helm-чарту в команде установки YAML вручную. Образы PT CS и Helm-чарт [могут быть опубликованы в реестре \(см. раздел 5.6\)](#) при установке в центральном кластере.

После установки PT CS дочерний кластер должен пройти инициализацию. Инициализация включает получение лицензии и БДУ из центрального кластера и регистрацию в нем. Эти действия выполняются без участия пользователя. После

успешной инициализации статус установки на странице **Кластеры** в веб-интерфейсе изменится на **Подключен**. Если статус не изменяется дольше 30 минут, [проанализируйте журналы событий компонента PTCS manager \(см. раздел 18\)](#).

Шаг 8. Добавление самоподписанных сертификатов в доверенные

Работа с дочерними кластерами осуществляется в веб-интерфейсе центрального кластера. Если при установке PT CS в центральном кластере были сгенерированы самоподписанные сертификаты, то получить доступ к данным дочерних кластеров не удастся из-за невозможности подтвердить безопасность соединения. Поэтому перед выбором дочернего кластера пользователям нужно [добавить корневой сертификат в доверенные или перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении \(см. раздел 18\)](#).

См. также

[Установка из онлайн-реестра \(см. раздел 5.4\)](#)

13.3.2. Удаление дочернего кластера

Если PT CS в дочернем кластере еще не установлен и дочерний кластер не зарегистрирован в PT CS, то вы можете удалить дочерний кластер из БД и веб-интерфейса.

Если PT CS уже установлен в дочернем кластере и дочерний кластер зарегистрирован в PT CS, то удаление осуществляется в несколько этапов:

1. Получение команды удаления в веб-интерфейсе PT CS.
2. Запуск команды удаления в дочернем кластере.
3. Дерегистрация дочернего кластера в PT CS. Происходит автоматически без участия пользователя.

Внимание! После удаления зарегистрированного кластера просмотр событий сканирования образов и конфигураций, проверки через контроллера допуска и мониторинг событий среды выполнения для этого кластера будет недоступен.

Удаление дочернего кластера, незарегистрированного в PT CS

► Чтобы удалить дочерний кластер, незарегистрированный в PT CS:

1. В главном меню выберите **Кластеры**.
2. Нажмите  в блоке с кластером, который необходимо удалить.

Информация о кластере будет удалена из БД и веб-интерфейса PT CS.

Удаление дочернего кластера, зарегистрированного в PT CS

► Чтобы удалить дочерний кластер, зарегистрированный в PT CS:

1. В главном меню выберите **Кластеры**.
2. Нажмите  в блоке с кластером, который необходимо удалить.

Примечание. Кнопка  неактивна, если кластер выбран в раскрывающемся списке **Кластер** вверху страницы справа.

3. Скопируйте полученную команду удаления.
4. Подключитесь к дочернему кластеру с помощью утилиты `kubectl` или других утилит для управления кластером.
5. Выполните скопированную команду удаления.

Дерегистрация дочернего кластера в PT CS и его удаление из веб-интерфейса будут выполнены автоматически через некоторое время после выполнения команды удаления.

13.4. Управление учетными записями пользователей

Вы можете использовать для доступа в PT CS локальные и корпоративные учетные записи. Для использования корпоративных учетных записей вам нужно настроить подключение к LDAP-серверу. Вы можете управлять учетными записями пользователей в веб-интерфейсе PT CS на странице **Пользователи**, которая открывается при выборе в главном меню раздела **Пользователи**.

Примечание. Управление учетными записями пользователей дочерних кластеров осуществляется в веб-интерфейсе центрального кластера. Пользователи могут использовать свою учетную запись для работы со всеми подключенными кластерами.

Пользователи				
Список пользователей		Синхронизация с LDAP	Правила маппинга ролей	Добавить пользователя
Логин	Электронная почта	Роль	Учетная запись	
admin@ptcs.example.com	—	Администратор	Системная	 
developer@ptcs.example.com	—	Разработчик	Системная	 
s_operator@ptcs.example.com	—	Специалист по безопасности	LDAP	 
auditor@ptcs.example.com	—	Аудитор	Системная	 

Рисунок 10. Вкладка **Список пользователей**

На странице отображаются вкладка **Список пользователей** для управления учетными записями, созданными в PT CS, вкладки **Синхронизация с LDAP** и **Правила маппинга ролей** для управления корпоративными учетными записями.

На вкладке **Список пользователей** отображаются:

- кнопка **Добавить пользователя** для добавления локальной учетной записи;
- таблица с информацией о локальных и корпоративных учетных записях;
- кнопки для изменения и удаления учетных записей.

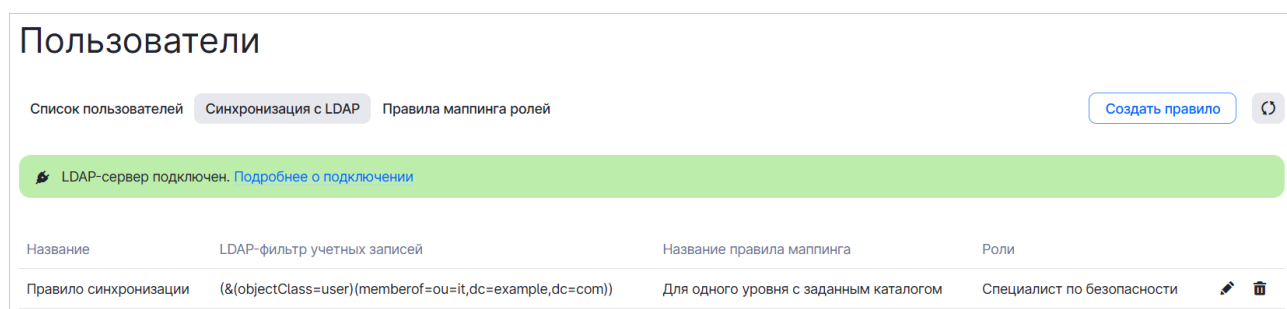


Рисунок 11. Вкладка **Синхронизация с LDAP**

На вкладке **Синхронизация с LDAP** отображаются:

- Кнопка **Подключить** для настройки подключения к LDAP-серверу. Отображается, только если не настроено подключение к LDAP-серверу.
- Ссылка **Подробнее о подключении** для просмотра параметров подключения к LDAP-серверу, их изменения или отключения LDAP-сервера. Отображается, только если настроено подключение к LDAP-серверу.
- Кнопка **Создать правило** для создания правила синхронизации с LDAP-сервером.
- Таблица с информацией о правилах синхронизации с LDAP-сервером.
- Кнопки для изменения и удаления для каждого правила.
- Кнопка  для синхронизации с LDAP-сервером.

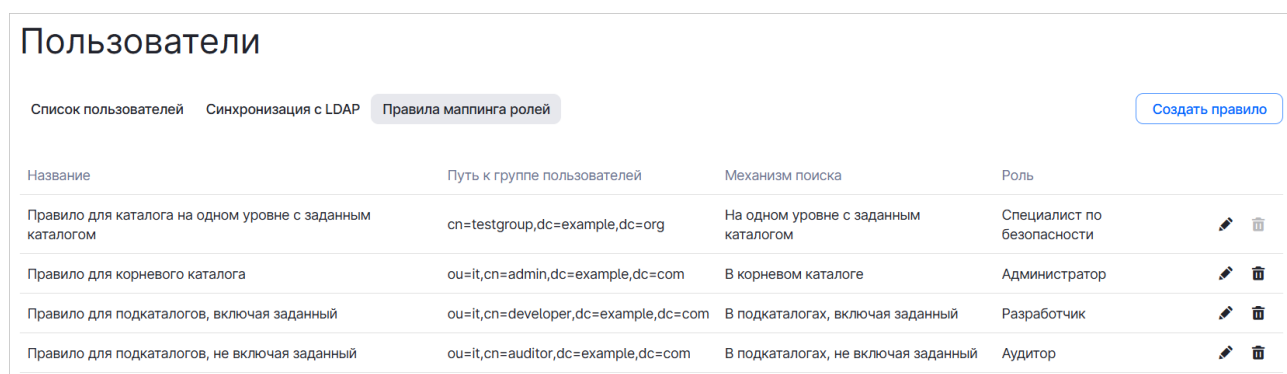


Рисунок 12. Вкладка **Правила маппинга ролей**

На вкладке **Правила маппинга ролей** отображаются:

- кнопка **Создать правило** для создания правила маппинга ролей;
- таблица с информацией о правилах маппинга ролей;
- кнопки для изменения и удаления для каждого правила.

В этом разделе

[Управление локальными учетными записями \(см. раздел 13.4.1\)](#)

[Управление корпоративными учетными записями \(см. раздел 13.4.2\)](#)

13.4.1. Управление локальными учетными записями

Локальные учетные записи создаются и настраиваются в PT CS и могут быть использованы для работы только с этим продуктом. В этом разделе содержатся инструкции по управлению локальными учетными записями.

► Чтобы создать учетную запись пользователя:

1. В главном меню выберите **Пользователи**.
2. Нажмите **Добавить пользователя**.
3. Введите логин и пароль.
4. Введите адрес электронной почты пользователя.
5. Выберите [роль \(см. раздел 13.2\)](#), которая будет присвоена учетной записи пользователя.
6. Нажмите **Добавить**.

Вы можете изменить параметры учетной записи, в том числе пароль, с помощью  или удалить ее с помощью .

Примечание. Новый пароль не должен быть пустым или слишком простым, не должен совпадать с предыдущим и должен содержать от 8 до 16 символов, заглавные и строчные буквы латинского алфавита и цифры.

Примечание. Если в момент удаления учетной записи пользователь работает в веб-интерфейсе PT CS, его сессия будет завершена.

Примечание. Удаление учетной записи недоступно, если это единственная учетная запись с ролью **Администратор**.

13.4.2. Управление корпоративными учетными записями

Если в вашей организации есть сервер, на котором реализован LDAP-протокол, то вы можете настроить подключение к нему, чтобы пользователи могли входить в PT CS с помощью корпоративных учетных записей. Можно подключить только один LDAP-сервер.

Для синхронизации с LDAP-сервером и использования корпоративных учетных записей вам нужно:

- Подключить LDAP-сервер.
- Создать правила маппинга ролей. Правила маппинга соотносят роли пользователей в PT CS с ролями или группами, настроенными на LDAP-сервере. Например, если создать правило маппинга ролей между группой `ou=it,cn=soperator,dc=example,dc=com` и ролью **Специалист по безопасности**, то после выполнения синхронизации в PT CS будут перенесены или обновлены учетные записи пользователей LDAP и им будет присвоена роль **Специалист по безопасности**.
- Создать правила синхронизации с LDAP-сервером. Правила синхронизации позволяют задать LDAP-фильтры для правил маппинга.

Примечание. Вы также можете сначала создать правила синхронизации с LDAP-сервером, а затем — правила маппинга и выбрать их в уже существующих правилах синхронизации.

- Выполнить синхронизацию с LDAP-сервером. После этого в PT CS будут добавлены корпоративные учетные записи.

При настройке подключения к LDAP-серверу вам нужно указать каталог, в котором будет осуществляться поиск учетных записей. Например, если задано значение `ou=it,dc=example,dc=com`, то поиск будет выполняться в каталоге `it`, в котором хранятся учетные записи пользователей ИТ-подразделения.

Вы можете сузить область поиска с помощью LDAP-фильтра в правиле синхронизации. Например, если вы укажете в LDAP-фильтре `(&(objectClass=user)(memberof=ou=it,dc=example,dc=com))`, то из каталога `ou=it,dc=example,dc=com` будут отобраны записи с классом объекта `user`, то есть учетные записи пользователей. Если LDAP-фильтры не заданы в правилах синхронизации, то используется фильтр, заданный в параметрах подключения к LDAP-серверу.

В LDAP-фильтре вы можете задать несколько условий, используя следующие операторы:

- `&` — для одновременного выполнения всех условий;
- `!` — для указания условия, которое не должно быть выполнено;
- `|` — для обозначения условий, для которых достаточно выполнения хотя бы одного из них.

В правилах маппинга вы можете настроить сопоставление учетных записей, отобранных из каталогов LDAP-сервера, с ролями учетных записей в PT CS. Например, указать путь к группе `ou=it,cn=developer,dc=example,dc=com` для поиска учетных записей из группы `developer` ИТ-подразделения и присвоения им роли **Разработчик**.

Если настроено подключение к LDAP-серверу и созданы правила синхронизации и маппинга, то пользователи смогут авторизоваться с помощью корпоративных учетных записей даже при отсутствии соединения с LDAP-сервером. В таком случае будут использоваться данные, полученные при последней синхронизации. Например, если учетная запись была добавлена после нее, то пользователь не сможет авторизоваться. Если после синхронизации пароль был изменен, то пользователь не сможет авторизоваться с новым паролем, но сможет — со старым.

Подключение к LDAP-серверу

Для подключения к LDAP-серверу вам нужно использовать учетную запись с правом на чтение каталога пользователей. Кроме того, у такой учетной записи должно быть право на чтение атрибута `userPassword`, если необходимо обеспечить возможность пользователям, учетные записи которых были добавлены при синхронизации с LDAP-сервером, авторизоваться при отсутствии соединения с LDAP-сервером. Это право позволит захешировать данные учетных записей в PT CS.

► Чтобы настроить подключение к LDAP-серверу:

1. В главном меню выберите **Пользователи**.
2. Выберите вкладку **Синхронизация с LDAP**.
3. Нажмите **Подключить**.
4. Введите адрес LDAP-сервера в формате <IP-адрес или доменное имя>:<Порт>.

Примечание. Если требуется подключиться к серверу анонимно, вы можете включить анонимный режим. В этом случае поля **Имя учетной записи** и **Пароль** скрываются.

5. Введите имя учетной записи пользователя и пароль.

Примечание. В качестве идентификатора учетной записи пользователя может быть использован один из ее атрибутов, например UID или логин. В поле **Имя учетной записи** вам нужно указать значение того атрибута, который будет задан в поле **Атрибут пользователей**. Например, если в поле **Атрибут пользователей** вы укажете UID, то в поле **Имя учетной записи** вам нужно указать значение UID учетной записи пользователя.

6. Если требуется шифровать соединение по протоколу TLS, то установите флажок **Использовать TLS**.
7. Введите базовую часть уникального имени LDAP-пользователя.
8. Введите уникальный идентификатор пользователей.
9. Введите LDAP-фильтр для поиска учетных записей пользователей. В качестве фильтра вы можете указать класс объекта, который однозначно указывает, что этот объект в каталогах LDAP-сервера является учетной записью пользователя.
10. Нажмите **Подключить**.

Примечание. Вы можете нажать **Подключить без проверки**, если хотите добавить LDAP-сервер без подключения к нему.

Создание правила маппинга ролей

► Чтобы создать правило маппинга ролей:

1. В главном меню выберите **Пользователи**.
2. Выберите вкладку **Правила маппинга ролей**.
3. Нажмите **Создать правило**.
4. Введите название правила.
5. Введите путь к каталогу LDAP-сервера, в котором будет осуществляться поиск учетных записей пользователей.
6. Выберите механизм поиска учетных записей пользователей в каталогах LDAP-сервера.
7. Выберите **роль** (см. [раздел 13.2](#)), которая будет присвоена учетной записи пользователя.
Выбранная роль будет присвоена учетным записям пользователей, которые будут добавлены при синхронизации с LDAP-сервером.
8. Нажмите **Создать**.

Создание правила синхронизации с LDAP-сервером

► Чтобы создать правило синхронизации с LDAP-сервером:

1. В главном меню выберите **Пользователи**.
2. Выберите вкладку **Синхронизация с LDAP**.
3. Нажмите **Создать правило**.
4. Введите название правила.
5. Введите LDAP-фильтр для поиска учетных записей пользователей на LDAP-сервере.

Примечание. Если в правиле синхронизации не будет задан фильтр учетных записей, то будет использован фильтр из параметров подключения к LDAP-серверу.

6. Выберите правило маппинга ролей.

После создания правил вам необходимо выполнить синхронизацию с LDAP-сервером, нажав кнопку  на вкладке **Синхронизация с LDAP**.

Вы не можете изменить параметры корпоративной учетной записи, в том числе пароль, в веб-интерфейсе PT CS. Вы можете изменить пароль, используя средства для работы с учетными записями LDAP-сервера.

Вы можете отключить соединение с LDAP-сервером, нажав сначала ссылку **Подробнее о подключении**, а затем — кнопку **Отключить**. В PT CS останутся только [локальные учетные записи](#) (см. [раздел 13.4.1](#)).

13.5. Мониторинг состояния PT CS

В PT CS реализован мониторинг параметров работоспособности и целостности системы и ее сервисов. Источником для мониторинга служат статистические данные за определенные промежутки времени. Для визуализации, мониторинга и анализа системных событий и метрик PT CS может быть интегрирован с Grafana. Для хранения и просмотра системных журналов PT CS поддерживает возможность интеграции с Grafana Loki.

В этом разделе

[Работа с дашбордами \(см. раздел 13.5.1\)](#)

[Просмотр записей в системном журнале \(см. раздел 13.5.2\)](#)

[Формат сообщений в системном журнале \(см. раздел 13.5.3\)](#)

13.5.1. Работа с дашбордами

[Grafana](#) — система для визуализации данных. Grafana отображает информацию в удобном для анализа виде и может использоваться для мониторинга работы PT CS.

После установки веб-интерфейс Grafana будет доступен по адресу <Адрес веб-интерфейса PT CS в центральном кластере>/grafana. Для входа необходимо ввести логин и пароль, указанные при установке Grafana.

Дашборды в Grafana

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

Grafana предоставляет набор дашбордов для мониторинга различных аспектов работы PT CS. Каждый дашборд отображает специфическую информацию о состоянии системных компонентов и сервисов. На каждом дашборде можно выбрать кластер, для которого нужно отображать информацию.

В комплект поставки входят следующие дашборды:

- Admission — предназначен для мониторинга работы детекторов для проверки через контроллер допуска. Отображает количество ошибок при их работе и продолжительность работы цепочки детекторов.
- Cluster status — предоставляет информацию о работоспособности внутренних сервисов ClickHouse, RabbitMQ, PostgreSQL, текущем состоянии кластера (центральный, дочерний и не зарегистрированный, дочерний и зарегистрированный), статусе обновления лицензий PT CS и баз данных уязвимостей. Для получения информации о работоспособности внешних сервисов вам нужно дополнительно настроить внутреннюю систему Prometheus или установить внешнюю систему Prometheus.

- Go app — предоставляет основные метрики для сервисов, разработанных на языке программирования Go. Отображает ключевые показатели производительности и стабильности этих компонентов.
- gRPC — предоставляет сводную информацию о взаимодействии между сервисами PT CS (типах запросов, их статусов, длительности) с использованием протокола gRPC. Поскольку большинство внутренних взаимодействий между сервисами осуществляется через gRPC, все связанные метрики собраны в этом дашборде для удобного анализа.
- Public API — предназначен для мониторинга работы публичного API. Отображает состояние и производительность внешнего интерфейса, обеспечивая контроль доступности и качества предоставляемых услуг.
- Reverse проху — предоставляет информацию о доступности и работе компонента Reverse проху. Все внутренние и внешние запросы, поступающие в систему PT CS, направляются через компонент Reverse проху. Дашборд позволяет выявлять аномалии в работе сервиса и оперативно реагировать на возможные проблемы.
- Runtime — создан для работы с метриками сервисов, предназначенных для внутреннего развертывания в кластере. В случае использования внешних компонентов, таких как RabbitMQ или ClickHouse, для корректной фильтрации по параметру **Кластер** и при наличии конфигурации Grafana для нескольких кластеров необходимо добавить метку cluster к метрикам этих компонентов (аналогично другим метрикам системы). Также отображается информация о подключении компонента Runtime monitor к RabbitMQ.
- Container registry — предоставляет информацию о времени обработки задания на сканирование по расписанию (без учета времени сканирования), количестве запросов от компонента Container registry integrator к компоненту Policy enforcer для проверки существования правил реагирования для конкретного образа, количестве успешных и не успешных запросов к реестрам с образами и количестве запросов к сканеру для проверки образа на уязвимости в рамках обработки заданий на сканирование по расписанию. Эта информация позволяет осуществлять мониторинг нагрузки, связанной с обработкой заданий на сканирование по расписанию, своевременно выявлять аномалии в работе модуля и реагировать на возможные проблемы.
- Tetragon — предоставляет информацию об использовании центрального процессора и ОЗУ инструментом Tetragon, а также — информацию об активности событий и политик, времени обработки событий, потерях событий из-за переполнения буферов и состоянии загрузки политик на узлах для инструмента Tetragon. График со временем обработки событий отображается, если включен параметр `kernel.bpf_stats_enabled`. Подробнее о параметре вы можете прочесть на сайте tetragon.io.

Каждый дашборд содержит аннотации и уведомления, которые помогают оперативно реагировать на изменения состояния компонентов.

Если используется Grafana из дистрибутива PT CS, то дашборды будут доступны в веб-интерфейсе Grafana. Если используется внешняя Grafana, то вам нужно импортировать предварительно настроенные дашборды для мониторинга работы PT CS. Файлы с дашбордами доступны в каталоге `dashboards` архива с Helm-чартом установочного дистрибутива.

► Чтобы импортировать дашборд PT CS в Grafana:

1. Войдите в веб-интерфейс Grafana.
2. В главном меню по кнопке  выберите раздел **Dashboards**.
3. Нажмите **New** → **Import**.
4. Нажмите **Upload dashboard JSON file**.
5. Выберите файл дашборда PT CS.
6. Нажмите **Load**.

13.5.2. Просмотр записей в системном журнале

Записи о событиях PT CS сохраняются в системном журнале, если при развертывании был указан [параметр \(см. раздел 5.5\)](#) --enable-logger. После установки веб-интерфейс Grafana будет доступен по адресу <Адрес веб-интерфейса PT CS в центральном кластере>/grafana. Для входа необходимо ввести логин и пароль, указанные при установке Grafana.

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 **Explore** .
3. В раскрывающемся списке сверху выберите источник данных **Loki**.
4. Выберите режим редактирования запроса **Code**.
5. Нажмите **Log browser**.
6. В блоке параметров **Label filters** выберите метки и их значения, по которым нужно искать записи в журнале.
7. В блоке параметров **Find values for the selected labels** укажите значения выбранных меток.

Примечание. При формировании запросов вам нужно учитывать [форматы сообщений в системном журнале \(см. раздел 13.5.3\)](#). Например, для поиска событий создания шаблона уведомлений необходимо указать метку object со значением notification и метку operation со значением create.

8. Нажмите **Show logs**.
9. Нажмите **Run query**.

13.5.3. Формат сообщений в системном журнале

В PT CS регистрируются следующие события:

- вход в систему, выход из системы;
- создание учетной записи пользователя, изменение учетной записи (в том числе, изменение полномочий пользователя), удаление учетной записи;
- создание LDAP-подключения, изменение LDAP-подключения, синхронизация с LDAP, удаление LDAP-подключения;
- создание правила реагирования, изменение правила реагирования, удаление правила реагирования;
- создание сервиса уведомлений, изменение сервиса уведомлений, удаление сервиса уведомлений;
- создание шаблона уведомлений, изменение шаблона уведомлений, удаление шаблона уведомлений;
- создание токена, удаление токена;
- создание подключения к реестру образов, удаление подключения к реестру образов;
- запуск сканирования конфигурации;
- запуск сканирования образа;
- изменение конфигурации детекторов;
- изменение параметров отслеживания и фильтров событий среды выполнения;
- активация лицензии;
- обновление базы уязвимостей;
- подключение кластера и генерация команды установки.

Сообщения отправляются при выполнении соответствующих действий в PT CS. Формат сообщений одинаковый.

Тело сообщения:

```
{  
  "authorized": "<Bool>",  
  "user_id": "<UUID>",  
  "object": "<String>",  
  "operation": "<String>",  
  "args": <JSON-object>,  
  "result": <JSON-object>  
}
```

Таблица 22. Поля сообщения

Поле (объект)	Тип данных	Описание
authorized	Boolean	Прошел ли запрос авторизацию
user_id	String (UUID)	Идентификатор пользователя, инициировавшего запрос
object	String	Тип объекта, над которым выполнено действие: — <code>user</code> — вход в систему, выход из системы, создание учетной записи пользователя, изменение учетной записи, удаление учетной записи; — <code>ldap</code> — создание LDAP-подключения, изменение LDAP-подключения, удаление подключения; — <code>ldap_sync</code> — синхронизация с LDAP; — <code>rule</code> — создание правила реагирования, изменение правила реагирования, удаление правила реагирования; — <code>integration</code> — создание сервиса уведомлений, изменение сервиса уведомлений, удаление сервиса уведомлений; — <code>notification</code> — создание шаблона уведомлений, изменение шаблона уведомлений, удаление шаблона уведомлений; — <code>repository</code> — создание подключения к реестру образов, удаление подключения к реестру образов; — <code>token</code> — создание токена, удаление токена; — <code>event_processor_config</code> — изменение конфигурации детекторов; — <code>runtime_monitor_config</code> — изменение параметров отслеживания и фильтров событий среды выполнения —; — <code>license</code> — активация лицензии; — <code>cluster</code> — обновление базы уязвимостей — <code>vuIndb</code>; — <code>cluster</code> — подключение кластера и генерация команды установки
operation	String	Название выполненного действия: — <code>create</code> — создание учетной записи пользователя, создание LDAP-подключения, синхронизация с LDAP, создание правила реагирования, создание

Поле (объект)	Тип данных	Описание
		сервиса уведомлений, создание шаблона уведомлений, создание подключения к реестру образов, создание токена, подключение кластера и генерация команды установки; — <code>update</code> — изменение учетной записи, изменение LDAP-подключения, изменение правила, изменение сервиса уведомлений, изменение шаблона уведомлений; — <code>delete</code> — удаление правила, удаление сервиса уведомлений, удаление шаблона уведомлений, удаление подключения к реестру образов, удаление токена; — <code>execute</code> — запуск сканирования конфигурации, запуск сканирования образа; — <code>update</code> — удаление учетной записи, удаление LDAP-подключения, изменение конфигурации детекторов, изменение параметров отслеживания и фильтров событий среды выполнения, обновление базы уязвимостей; — <code>activate</code> — активация лицензии; — <code>create</code> — подключение кластера и генерация команды установки
<code>args</code>	JSON object	Параметры запроса
<code>result</code>	JSON object	Параметры ответа на запрос

14. Настройка учетной записи пользователя

При работе с интерфейсом PT CS каждый пользователь использует собственную учетную запись. Вы можете менять пароль своей учетной записи, а также создавать и отзываться токены доступа к публичному API PT CS.

Личный кабинет пользователя доступен в главном меню по кнопке .

В этом разделе

[Смена своего пароля \(см. раздел 14.1\)](#)

[Управление токенами доступа \(см. раздел 14.2\)](#)

14.1. Смена своего пароля

Вы можете сменить свой пароль для входа в PT CS.

► Чтобы сменить свой пароль:

1. В главном меню выберите  → **Смена пароля**.
2. Введите новый пароль.

Примечание. Новый пароль не должен быть пустым или слишком простым, не должен совпадать с предыдущим или должен содержать от 8 до 16 символов, заглавные и строчные буквы латинского алфавита и цифры.

3. Нажмите **Сохранить**.

14.2. Управление токенами доступа

Токен доступа используется для авторизации запросов к REST API. Подробнее о REST API см. в Справочном руководстве по REST API.

Управление токенами доступа в веб-интерфейсе осуществляется на странице **Токены доступа**, которая открывается при нажатии в главном меню  → **Токены доступа**.

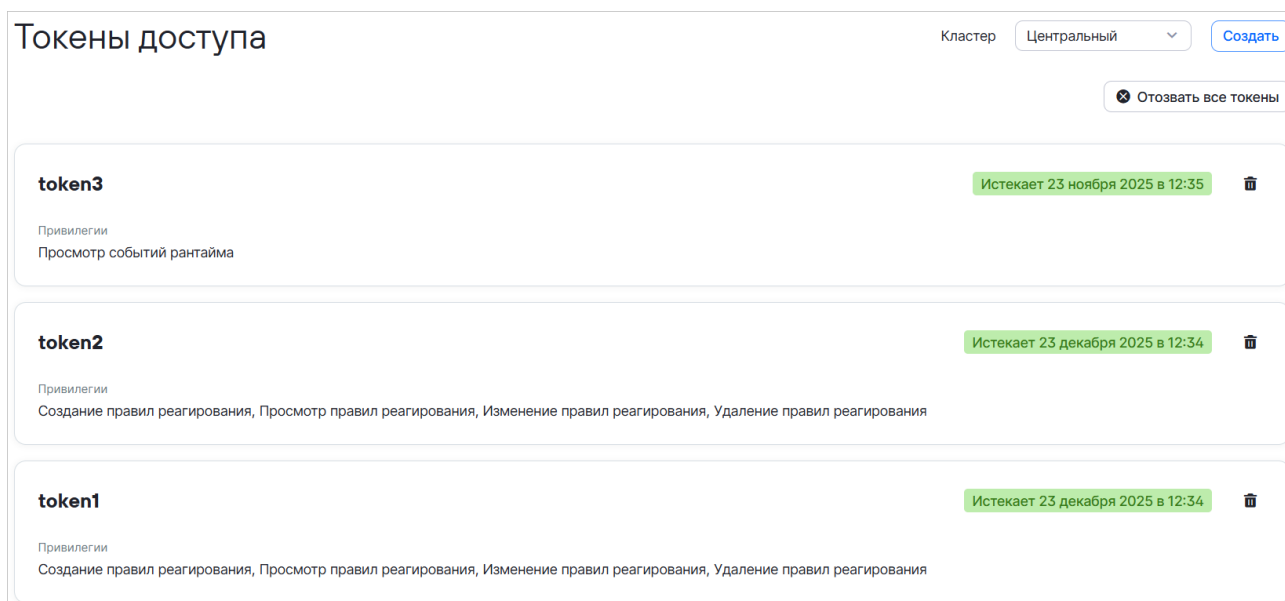


Рисунок 13. Страница **Токены доступа**

В правой верхней части страницы отображается раскрывающийся список **Кластер**, позволяющий выбрать кластер для работы. Если к PT CS не подключены дочерние кластеры или дочерние кластеры еще не находятся в статусе **Установлен**, то для выбора доступно только значение **Центральный**.

На странице отображаются:

- кнопка **Создать** для создания токена;
- кнопка **Отозвать все токены**;
- блоки с параметрами токенов доступа.

В каждом блоке отображаются:

- название токена доступа;
- срок действия;
- список доступных привилегий;
- кнопка  для отзыва токена доступа.

Вы можете создать один или несколько токенов доступа для своей учетной записи.

► Чтобы создать токен доступа:

1. В главном меню нажмите  → **Токены доступа**.
2. Вверху страницы справа выберите кластер, для запросов к которому нужно создать токен.
3. Нажмите **Создать**.

4. Введите название токена доступа.
5. Выберите срок действия токена.
6. Включите необходимые для токена привилегии.

Примечание. Вы можете включить только те привилегии, которые доступны [роли вашей учетной записи \(см. раздел 13.2\)](#).

7. Нажмите **Создать**.

Внимание! Скопируйте и сохраните полученный токен доступа. После обновления страницы его просмотр и копирование будут недоступны.

Вы можете отозвать токен доступа, нажав  в блоке с ним, или отозвать все токены доступа, нажав **Отозвать все токены**. После отзыва токена авторизация запросов к REST API с его использованием будет невозможна.

15. Обновление PT CS

Внимание! При обновлении PT CS в центральном кластере до версии 1.2 необходимо обновить PT CS в дочерних кластерах. После обновления результаты сканирований, произведенных в предыдущих версиях, станут недоступны. Это обусловлено тем, теперь за сканирование отвечает новый микросервис и модель данных событий сканирования значительно расширена. Кроме того, новый микросервис позволяет управлять задачами сканирования образов и изменять их приоритеты. Из-за этих архитектурных изменений старые результаты сканирований невозможно автоматически перенести в новую версию, так как они не содержат необходимых данных и не поддерживают новую логику работы с задачами. Если вам нужны отчеты прошлых сканирований, обязательно скачайте их до начала обновления. После обновления результаты новых сканирований будут доступны для на странице **Сканирования образов**.

► Чтобы обновить PT CS с версии 1.1 до версии 1.2:

1. Получите данные для соединения с PostgreSQL, выполнив команду:

```
kubect1 get secret -n <Пространство имен> postgresql -o json | jq '.data | map_values(@base64d)'
```

Пример вывода команды:

```
{
  "postgres-addr": "postgresql",
  "postgres-db": "ptcs",
  "postgres-password": "admin",
  "postgres-ssl-check-cert": "true",
  "postgres-ssl-mode": "true",
  "postgres-user": "admin"
}
```

2. Подключитесь к базе данных PostgreSQL, используя полученные учетные данные.
3. Сохраните резервную копию таблицы `detectors`.
4. Удалите содержимое таблицы `detectors`, выполнив команду:


```
truncate table detectors
```
5. Если в предыдущих версиях были загружены детекторы, не входящие в поставку перекомпилируйте их с использованием `tinycgo` версии 0.41.1.
6. Скопируйте и сохраните параметры `encryption`, `publicAccessTokenSalt` и `token` из конфигурационного файла Helm-чарта (`values.yaml`) от предыдущей инсталляции.

Например:

```
keys:
  encryption: 6234d...a0bbd
  publicAccessTokenSalt: 4572d...856ff
  token: c1f3a...c2b1d
```

7. Удалите сертификаты для безопасного взаимодействия между компонентами PT CS.

Примечание. По умолчанию для хранения сертификатов используется конфигурационный файл Helm-чарта `values.yaml`. При развертывании PT CS вы могли сохранить сертификаты в другом файле. В таком случае вам необходимо удалить сертификаты из этого файла.

8. Выполните одно из следующих действий:

- Сгенерируйте и добавьте в конфигурационный файл Helm-чарта сертификаты для безопасного взаимодействия между модулями PT CS.
- Запустите скрипт `generate_values.sh` с теми же параметрами, что и при установке предыдущей версии, для автоматического добавления сертификатов в конфигурационный файл Helm-чарта.

9. Добавьте в конфигурационный файл Helm-чарта параметры `encryption`, `publicAccessTokenSalt` и `token` от предыдущей инсталляции и сохраните изменения.

10. Замените секцию `ptcs-manager` на `cs-manager` в конфигурационном файле Helm-чарта:

```
cs-manager:
  flusUrl: "https://update.ptsecurity.com"
```

Примечание. Остальные параметры совместимы со старым форматом, поэтому обновление пройдет успешно и без их переименования, но рекомендуется переименовать их сразу. Подробная информация о переименовании параметров и секций приведена в таблице после инструкции.

11. Удалите объекты `StatefulSet` и `Deployment`, выполнив команды:

```
kubectl delete statefulset redis-master -n $<Пространство имен>
kubectl delete statefulset --all -n $<Пространство имен> --cascade=orphan
kubectl delete deploy prometheus-server grafana -n $<Пространство имен>
```

Внимание! Удаление объектов приведет к временной остановке сервисов мониторинга (Prometheus, Grafana) и отключению автоматического восстановления подов до завершения обновления. Данные сохраняются: удаление объектов не затрагивает тома (PVC), и новые поды подключатся к ним после обновления. Указание параметра `--cascade=orphan` предотвращает удаление подов вместе со `StatefulSet`, поэтому они продолжат работу до создания нового контроллера. Для выполнения команд `kubectl` в целевом пространстве имен требуются права администратора.

12. Выполните одно из следующих действий:

- Если вы заполнили конфигурационный файл Helm-чарта вручную, выполните команду для установки нового пакета PT CS:

```
helm upgrade --install <Имя релиза> <Путь к архиву с Helm-чартом> --version <Номер версии>
--namespace <Пространство имен> --create-namespace --values <Имя конфигурационного файла
Helm-чарта>.yaml
```
- Если вы заполнили конфигурационный файл Helm-чарта с помощью скрипта `generate_values.sh`, скопируйте и выполните полученную в результате работы скрипта команду установки.

Таблица 23. Переименование параметров и секций в конфигурационном файле Helm-чарта

Название в версии 1.1	Название в версии 1.2	Поддерживается совместимость
global.ownPtcsUrl	global.ownCsUrl	Да
global.centralPtcsUrl	global.centralCsUrl	Да
postgresql.primary.*	postgresql.*	Да
redis.master.*	redis.*	Да
prometheus.server.*	prometheus.*	Да
auth-api.administrator.username/.password	administrator.username/.password	Да
ptcs-manager.*	cs-manager.*	Нет

Если вы используете PT CS для защиты нескольких кластеров, то вам необходимо:

1. Обновить PT CS в центральном кластере по инструкции выше.
2. [Подключить дочерние кластеры \(см. раздел 13.3.1\)](#) в веб-интерфейсе PT CS в центральном кластере.
3. Обновить PT CS в дочерних кластерах с помощью команд, полученных при их подключении.

Примечание. После обновления перекомпилированные детекторы нужно [заново загрузить \(см. раздел 9.3\)](#) в веб-интерфейсе.

16. Настройка локального сервера обновлений

PT CS может обновляться в изолированном от интернета сегменте сети с использованием локального сервера обновлений. Возможны две схемы развертывания локального сервера обновлений.

Если PT CS установлен в изолированном от интернета сегменте сети, вы можете использовать схему обновления с двумя локальными серверами обновлений: один в изолированном сегменте сети, где установлен PT CS, другой — в демилитаризованной зоне. Локальный сервер в демилитаризованной зоне будет загружать обновления с глобального сервера обновлений Positive Technologies. Для передачи обновлений в закрытый сегмент сети вы можете либо [вручную копировать их при помощи внешнего носителя \(см. раздел 16.5\)](#), либо [настроить автоматическую передачу обновлений \(см. раздел 16.6\)](#) — если между локальными серверами обновлений есть сетевое взаимодействие.

Если из изолированного сегмента сети организации есть доступ в интернет (напрямую или через прокси-сервер), вы можете реализовать схему обновления с одним локальным сервером обновлений, расположенным в демилитаризованной зоне. Он будет загружать обновления с глобального сервера обновлений Positive Technologies и передавать их в изолированный сегмент сети.

В этом разделе

[Аппаратные и программные требования к локальному серверу обновлений \(см. раздел 16.1\)](#)

[Установка локального сервера обновлений \(см. раздел 16.2\)](#)

[Активация лицензии локального сервера обновлений \(см. раздел 16.3\)](#)

[Настройка локального сервера обновлений для работы через прокси-сервер \(см. раздел 16.4\)](#)

[Ручной перенос обновлений PT CS в закрытый сегмент сети \(см. раздел 16.5\)](#)

[Настройка автоматического переноса обновлений PT CS в закрытый сегмент сети \(см. раздел 16.6\)](#)

[Конфигурационный файл локального сервера обновлений \(см. раздел 16.7\)](#)

[Проверка и изменение параметров локального сервера обновлений \(см. раздел 16.8\)](#)

16.1. Аппаратные и программные требования к локальному серверу обновлений

Локальный сервер обновлений может быть установлен как на физическом сервере, так и в виртуальной среде.

Аппаратные требования

Для работы сервера требуются следующие минимальные аппаратные ресурсы:

- 1 ядро процессора;
- 2 ГБ оперативной памяти;
- 100 ГБ свободного места на диске.

Программные требования

Локальный сервер обновлений поддерживает установку на операционные системы семейства Linux: Astra Linux Special Edition 1.7, Debian 10, Debian 11, Ubuntu 18.04 или Ubuntu 22.04.

16.2. Установка локального сервера обновлений

► Чтобы установить локальный сервер обновлений:

1. Скопируйте архив с дистрибутивом PT CS в любой каталог.
2. Перейдите в каталог с загруженным архивом:
`cd <Имя каталога>`
3. Распакуйте установочный комплект в один из каталогов, например в `ptcs-installer`.
`tar xvf ptcs-<Версия продукта>.tar -C ptcs-installer/`
4. Перейдите в каталог с установочным комплектом:
`cd ptcs-installer/`
5. Запустите установку локального сервера обновлений:
`sudo dpkg -i pt-update-mirror-<Версия продукта>.deb`

Например:

```
sudo dpkg -i pt-update-mirror-2.2.1639.deb
```

Вы можете проверять состояние сервера с помощью команды `systemctl status pt-update-mirror.service` и просматривать его журналы с помощью команды `journalctl -u pt-update-mirror.service`.

После установки локального сервера в демилитаризованной зоне вам нужно [активировать на нем лицензию](#) (см. [раздел 16.3](#)). Локальный сервер обновлений в закрытом сегменте сети не требует активации, поскольку не подключается к публичному серверу обновлений Positive Technologies.

16.3. Активация лицензии локального сервера обновлений

После установки локального сервера обновлений нужно активировать его лицензию. Это необходимо для аутентификации локального сервера на глобальном сервере обновлений Positive Technologies.

Примечание. Если перенос обновлений на локальный сервер будет осуществляться вручную, активировать его лицензию не нужно.

Вы можете активировать лицензию с помощью токена лицензии, который хранится в файле `license-access-token.key`, полученном на физическом носителе или в электронном письме на адрес, указанный при покупке.

Если локальный сервер обновлений должен подключаться к интернету через прокси-сервер, перед активацией лицензии нужно [настроить подключение к этому прокси-серверу \(см. раздел 16.4\)](#).

Активация лицензии с помощью серийного номера

- ▶ Чтобы активировать лицензию с помощью серийного номера,

выполните команду:

```
sudo /opt/pt/pt-update-mirror/bin/pt-update-mirror license activate --serial-number
<Серийный номер лицензии>
```

Активация лицензии с помощью токена лицензии

- ▶ Чтобы активировать лицензию с помощью токена лицензии,

выполните команду:

```
sudo /opt/pt/pt-update-mirror/bin/pt-update-mirror license activate --license-token <Путь
к файлу license-access-token.key>
```

16.4. Настройка локального сервера обновлений для работы через прокси-сервер

Для подключения локального сервера обновлений к интернету через прокси-сервер необходимо указать параметры этого подключения в конфигурационном файле локального сервера.

Локальный сервер обновлений поддерживает подключение к прокси-серверу по протоколам HTTP и HTTPS.

Подключение к прокси-серверу по протоколу HTTP

- ▶ Чтобы настроить подключение локального сервера обновлений к прокси-серверу:

1. На локальном сервере обновлений откройте конфигурационный файл `config.json`:

```
sudo nano /etc/pt-update-mirror/config.json
```
2. В качестве значения параметра `proxy` укажите IP-адрес (и при необходимости порт) используемого прокси-сервера:

```
"proxy": "http://<IP-адрес прокси-сервера>:<Порт>"
```

3. Если прокси-сервер требует аутентификации, укажите логин и пароль для подключения:

```
"proxy-user": "<Логин>",
"proxy-password": "<Пароль>"
```

4. Выполните одно из следующих действий:

- Если вы хотите, чтобы некоторые маршруты не проксировались при включенном режиме проксирования, то задайте список таких маршрутов с помощью параметра `proxy-mode-blacklist`:

```
"proxy-mode-blacklist": ["/telemetry/v1/push/"]
```

Примечание. Режим проксирования запросов включен по умолчанию.

- Если вы хотите, чтобы запросы не проксировались, отключите режим проксирования запросов, задав параметру `proxy-mode` значение `false`:

```
"proxy-mode": false,
```

5. Сохраните изменения в файле `config.json`.
6. Перезапустите локальный сервер обновлений:

```
sudo systemctl restart pt-update-mirror.service
```

Подключение к прокси-серверу по протоколу HTTPS

Для проверки подлинности при передаче данных по протоколу HTTPS необходимо выпустить сертификат SSL, который должен:

- соответствовать формату PEM;
- использовать подпись с применением алгоритма шифрования SHA-256;
- использовать алгоритм шифрования ключей RSA;
- содержать закрытый незашифрованный ключ в формате PEM длиной не менее 2048 бит;
- включать в себя область применения сертификата Digital Signature или Key Encipherment;
- содержать в расширении Subject Alternative Name (SAN) запись о доменном имени или IP-адресе сервера с установленным веб-интерфейсом продукта;
- если в цепочке между корневым сертификатом и сертификатами компонентов и систем есть промежуточные сертификаты — включать в себя всю цепочку сертификатов.

Для корректной работы необходимо добавить выпущенный сертификат в список доверенных.

- Чтобы добавить сертификат в список доверенных:

1. Скопируйте файлы сертификата в каталог `/usr/local/share/ca-certificates/` на узле, на который установлен локальный сервер обновлений.
2. Обновите список доверенных сертификатов:

```
sudo update-ca-certificates
```

► Чтобы настроить подключение локального сервера обновлений к прокси-серверу:

1. На локальном сервере обновлений откройте конфигурационный файл `config.json`:

```
sudo nano /etc/pt-update-mirror/config.json
```

2. В качестве значения параметра `proxy` укажите IP-адрес (и при необходимости порт) используемого прокси-сервера:

```
"proxy": "https://<IP-адрес прокси-сервера>:<Порт>"
```

3. Если прокси-сервер требует аутентификации, укажите логин и пароль для подключения:

```
"proxy-user": "<Логин>",
```

```
"proxy-password": "<Пароль>"
```

4. Выполните одно из следующих действий:

- Если вы хотите, чтобы некоторые маршруты не проксировались при включенном режиме проксирования, то задайте список таких маршрутов с помощью параметра `proxy-mode-blacklist`:

```
"proxy-mode-blacklist": ["/telemetry/v1/push/"]
```

Примечание. Режим проксирования запросов включен по умолчанию.

- Если вы хотите, чтобы запросы не проксировались, отключите режим проксирования запросов, задав параметру `proxy-mode` значение `false`:

```
"proxy-mode": false,
```

5. Сохраните изменения в файле `config.json`.

6. Перезапустите локальный сервер обновлений:

```
sudo systemctl restart pt-update-mirror.service
```

16.5. Ручной перенос обновлений PT CS в закрытый сегмент сети

Если между локальными серверами обновлений отсутствует сетевое взаимодействие, вам нужно вручную перенести обновления в закрытый сегмент сети для последующего обновления PT CS.

► Чтобы вручную перенести обновления в закрытый сегмент сети:

1. На локальном сервере обновлений в демилитаризованной зоне запустите получение обновлений с глобального сервера обновлений Positive Technologies:

```
sudo /opt/pt/pt-update-mirror/bin/pt-update-mirror repository update
```

2. Запустите экспорт репозитория с обновлениями в файл:

```
sudo /opt/pt/pt-update-mirror/bin/pt-update-mirror repository export --repo vm-expertise <Название файла>.tgz
```

Примечание. Выполнение команды экспорта без ключа `--repo <Название репозитория>` позволяет экспортировать все репозитории базы обновлений в указанный архив. Вы можете просмотреть список репозиторий в базе обновлений локального сервера с помощью команды `sudo /opt/pt/pt-update-mirror/bin/pt-update-mirror repository view`.

3. Скопируйте с помощью внешнего носителя полученный файл архива в каталог на локальном сервере обновлений в закрытом сегменте сети, принадлежащий пользователю **pt-update-mirror**.
4. На локальном сервере обновлений в закрытом сегменте сети импортируйте обновления из скопированного файла архива:

```
sudo /opt/pt/pt-update-mirror/bin/pt-update-mirror repository import <Путь к архиву>/<Название архива>.tgz
```

16.6. Настройка автоматического переноса обновлений PT CS в закрытый сегмент сети

Если между локальными серверами обновлений есть сетевое взаимодействие, вы можете настроить их подключение друг к другу. Это позволит автоматически переносить обновления с глобального сервера обновлений Positive Technologies в PT CS в закрытом сегменте сети через цепочку локальных серверов обновлений.

Вы можете настроить автоматический перенос обновлений как по протоколу HTTP, так и по протоколу HTTPS.

Настройка автоматического переноса обновлений PT CS в закрытый сегмент сети при подключении по протоколу HTTP

Для подключения по протоколу HTTP вместо порта по умолчанию (80) используется порт 8553. Если требуется, вы можете [изменить порт \(см. раздел 16.8\)](#) для подключения.

- Чтобы настроить автоматический перенос обновлений в закрытый сегмент сети при подключении по протоколу HTTP:

1. На локальном сервере обновлений в изолированном сегменте откройте конфигурационный файл `config.json`:
2. В качестве значения параметра `update-server` укажите адрес локального сервера обновлений в демилитаризованной зоне:

`"update-server": "http://<Адрес локального сервера обновлений>:<Порт>"`

Примечание. Указанный в параметре порт локального сервера обновлений должен быть открыт для входящих соединений.

3. В качестве значения параметра `verify_ssl` укажите `false`.
4. Если подключение к локальному серверу в демилитаризованной зоне выполняется через прокси-сервер, [настройте параметры подключения к прокси-серверу \(см. раздел 16.4\)](#).
5. Сохраните изменения в файле `config.json`.
6. Перезапустите сервис обновлений в закрытом сегменте сети:

```
sudo systemctl restart pt-update-mirror.service
```

Настройка автоматического переноса обновлений PT CS в закрытый сегмент сети при подключении по протоколу HTTPS

Для подключения по протоколу HTTPS вместо порта по умолчанию (443) используется порт 8743. Если требуется, вы можете [изменить порт \(см. раздел 16.8\)](#) для подключения.

Для проверки подлинности при передаче данных по протоколу HTTPS необходимо выпустить сертификат SSL, который должен:

- соответствовать формату PEM;
- использовать подпись с применением алгоритма шифрования SHA-256;
- использовать алгоритм шифрования ключей RSA;
- содержать закрытый незашифрованный ключ в формате PEM длиной не менее 2048 бит;
- включать в себя область применения сертификата Digital Signature или Key Encipherment;
- содержать в расширении Subject Alternative Name (SAN) запись о доменном имени или IP-адресе сервера с установленным веб-интерфейсом продукта;
- если в цепочке между корневым сертификатом и сертификатами компонентов и систем есть промежуточные сертификаты — включать в себя всю цепочку сертификатов.

Для корректной работы необходимо назвать файлы сертификата и его ключа `cert.crt` и `cert.key` соответственно и поместить их в каталог `/etc/pt-update-mirror/https_certs/` на локальном сервере обновлений.

- Чтобы настроить автоматический перенос обновлений в закрытый сегмент сети при подключении по протоколу HTTPS:

1. На локальном сервере обновлений в изолированном сегменте откройте конфигурационный файл `config.json`:

```
sudo nano /etc/pt-update-mirror/config.json
```

2. В качестве значения параметра `update-server` укажите адрес локального сервера обновлений в демилитаризованной зоне:

```
"update-server": "https://<Адрес локального сервера обновлений>:<Порт>"
```

Примечание. Указанный в параметре порт локального сервера обновлений должен быть открыт для входящих соединений.

3. Если подключение к локальному серверу в демилитаризованной зоне выполняется через прокси-сервер, [настройте параметры подключения к прокси-серверу \(см. раздел 16.4\)](#).

4. Сохраните изменения в файле `config.json`.

5. Перезапустите сервис обновлений в закрытом сегменте сети:

```
sudo systemctl restart pt-update-mirror.service
```

16.7. Конфигурационный файл локального сервера обновлений

Конфигурационный файл дает возможность задать тип загрузчика, количество символов (разрядов) в имени версии для определения релизов, количество хранимых обновлений по каждому релизу, порядок загрузки, учет редакции обновления, параметры подключения к сервису update.ptsecurity.com.

По умолчанию скачивается одна последняя версия каждого доступного репозитория.

В таблице ниже приведены параметры конфигурационного файла локального сервера обновлений.

Таблица 24. Параметры конфигурационного файла локального сервера обновлений

Параметр	Описание
<code>check_edition</code>	Определяет, нужно ли учитывать значение параметра <code>edition</code>
<code>count_number_on_version_parse</code>	Задаёт количество разрядов в имени версии для определения релиза. Значение по умолчанию — 1
<code>db-path</code>	Задаёт путь по умолчанию для хранения обновлений. Значение по умолчанию — <code>var/opt/pt/pt-update-mirror</code>
<code>edition</code>	Задаёт редакцию для репозитория, если для параметра <code>check_edition</code> установлено значение <code>true</code> . Значение по умолчанию — пустая строка

Параметр	Описание
http_disabled	Отключает использование HTTP в веб-сервисе. Возможные значения: <code>true</code> (отключить), <code>false</code> (включить)
http_port	Задаёт порты для доступа по протоколу HTTP для веб-сервиса
https_port	Задаёт порты для доступа по протоколу HTTPS для веб-сервиса
logrotate	Задаёт ротацию журналов. Возможные значения: <code>size</code> (при размере журнала 50 МБ), <code>daily</code> (журналы ротируются в полночь)
log-level	Задаёт уровень журналирования. Возможные значения: <code>DEBUG</code> , <code>INFO</code> , <code>WARNING</code> , <code>ERROR</code>
minimal_release	Задаёт минимальный релиз для загрузки по репозиторию
order	Задаёт порядок загрузки репозиториях внутри списка репозиториях, доступных данной лицензии. Значение по умолчанию — 1
proxy	Задаёт адрес прокси-сервера. Поддерживается HTTP-протокол. Адрес должен начинаться с <code>http://</code>
proxy-mode	Включает режим проксирования запросов. Значение по умолчанию — <code>true</code>
proxy-mode-blacklist	Задаёт список маршрутов, которые не проксируются при включённом режиме проксирования. Значение по умолчанию — <code>["/telemetry/v1/push/"]</code>
proxy-user	Задаёт логин пользователя для подключения к прокси-серверу
proxy-password	Задаёт пароль пользователя для подключения к прокси-серверу
store_release_versions	Задаёт количество версий для хранения внутри релиза. Значение по умолчанию — 1
synchronizer	Задаёт тип загрузчика. Возможные значения: <code>ComponentSynchronizer</code> , <code>DockerSynchronizer</code> , <code>PackmanSynchronizer</code> . Значение по умолчанию — <code>ComponentSynchronizer</code>
update-server	Задаёт адрес сервиса обновления с <code>https://</code>
verify_ssl	Определяет, использовать ли SSL при подключении к локальному серверу обновлений. Значение по умолчанию — <code>true</code>

Ниже приведен пример конфигурационного файла локального сервера обновлений `config.json`.

```
{
  "products":
  {
```

```

    "ptcs":
    {
        "count_number_on_version_parse": 1,
        "store_release_versions": 1,
        "order": 1,
        "check_edition": false,
        "edition": ""
    }
},
"update-server": "https://update.ptsecurity.com",
"log-level": "INFO",
"logrotate": "size",
"proxy": "...",
"proxy-user": "...",
"proxy-password": "...",
"verify_ssl": true,
"db-path": "/var/opt/pt/pt-update-mirror",
"http_disabled": false,
"http_port": 8553,
"https_port": 8743,
"proxy-mode": true,
"proxy-mode-blacklist": ["/telemetry/v1/push/"]
}

```

16.8. Проверка и изменение параметров локального сервера обновлений

Проверка состояния сервера

- Чтобы проверить состояние локального сервера обновлений,

на локальном сервере выполните команду:

```
sudo systemctl status pt-update-mirror
```

На экране отобразится информация о параметрах загрузки и состоянии локального сервера.

Проверка состояния таймера получения обновлений

- Чтобы проверить состояние таймера получения обновлений,

на локальном сервере выполните команду:

```
sudo systemctl status pt-update-mirror-update.timer
```

На экране отобразится информация о параметрах загрузки и состоянии таймера получения обновлений.

Проверка состояния сервиса получения обновлений

- ▶ Чтобы проверить состояние сервиса получения обновлений,

на локальном сервере выполните команду:

```
sudo systemctl status pt-update-mirror-update
```

На экране отобразится информация о параметрах загрузки и состоянии сервиса получения обновлений.

Просмотр журналов событий сервиса получения обновлений

- ▶ Чтобы просмотреть журнал событий сервиса получения обновлений,

на локальном сервере выполните команду:

```
journalctl -u pt-update-mirror.service
```

На экране отобразится список событий сервиса получения обновлений.

Изменение времени получения обновлений

По умолчанию обновление запускается в 13, 27, 42 и 58 минут каждого часа. Вы можете изменить эти значения в параметрах таймера получения обновлений.

- ▶ Чтобы изменить время получения обновлений:

1. Откройте файл `pt-update-mirror-update.timer`:

```
sudo nano etc/systemd/system/pt-update-mirror-update.timer
```
2. В блоке параметров `Timer` в качестве значения параметра `OnCalendar` укажите нужное время получения обновлений (в формате `systemd.timer`).
3. Сохраните изменения в файле `pt-update-mirror-update.timer`.
4. Примените изменения:

```
sudo systemctl daemon-reload
```
5. Перезапустите таймер получения обновлений:

```
sudo systemctl restart pt-update-mirror-update.timer
```

Изменение порта подключения по протоколам HTTP и HTTPS

По умолчанию локальный сервер обновлений использует для подключения по протоколам HTTP и HTTPS порты 8553 и 8743 соответственно. Вы можете изменить эти значения в параметрах сервиса получения обновлений.

► Чтобы изменить порт подключения:

1. Откройте файл `pt-update-mirror.service`:

```
sudo nano etc/systemd/system/pt-update-mirror.service
```

2. Выполните одно из следующих действий:

Если для подключения локального сервера используется протокол HTTP, в блок параметров `Service` в конец строки с параметром `ExecStart` добавьте ключ `--bind 0.0.0.0:<Порт>`.

Если для подключения локального сервера используется протокол HTTPS, в блок параметров `Service` в конец строки с параметром `ExecStart` добавьте ключ `--https-port <Порт>`.

3. Сохраните изменения в файле `pt-update-mirror.service`.

4. Примените изменения:

```
sudo systemctl daemon-reload
```

5. Перезапустите локальный сервер обновлений:

```
sudo systemctl restart pt-update-mirror
```

17. Удаление PT CS

Вы можете запустить удаление PT CS с любого сервера, который соответствует [программным требованиям](#) (см. раздел 5.1).

► Чтобы удалить PT CS:

1. Перейдите в каталог с установочным комплектом:

```
cd ptcs-installer/
```

2. Выполните команду для удаления с параметром `--namespace`:

```
helm uninstall ptcs --namespace <Пространство имен>
```

18. Решение проблем

В этом разделе описываются возможные проблемы в работе PT CS и варианты их решения.

Установка PT CS завершается с ошибкой

Во время выполнения установочного скрипта отображается ошибка `default storage class not found!` или `storage class <Класс системы хранения> not found`. Установка PT CS завершается с ошибкой.

Проблема может возникать, если [в скрипте для генерации конфигурационного файла Helm-чарта \(см. раздел 5.5\)](#) не указан параметр `--storage-class`, а класс хранилища (StorageClass), заданный по умолчанию, в кластере не найден.

Для решения проблемы вам нужно указать корректное название класса системы хранения данных в значении параметра `--storage-class` и при необходимости переопределить классы систем хранения данных для PostgreSQL, Redis, RabbitMQ и ClickHouse с помощью параметров `--database-storage-class`, `--redis-storage-class`, `--rabbitmq-storage-class`, `--clickhouse-storage-class` соответственно.

- Чтобы узнать доступные в кластере классы систем хранения,

выполните команду:

```
kubectl get storageclass
```

Если список пустой, значит, системы хранения в кластере не развернуты. В этом случае вы можете развернуть нужную вам систему хранения. Подробнее о разворачивании систем хранения вы можете прочесть [в документации Kubernetes](#).

Утеряны данные после обновления или переустановки PT CS

После переустановки PT CS или обновления до новой версии утеряны все данные в продукте.

Проблема может возникать, если при обновлении или переустановке продукта в установочном скрипте были указаны параметры подключения к БД, которые отличаются от указанных при первой установке PT CS.

Для решения проблемы переустановите PT CS с теми параметрами подключения к БД, которые были указаны при первой установке продукта. Вы можете просмотреть эти параметры в файле `values.yaml`, который использовался при установке.

- Чтобы получить список параметров, указанных при установке PT CS, если файл `values.yaml` по каким-либо причинам просмотреть невозможно,

выполните команду:

```
helm get values <Имя релиза> -n <Пространство имен, в котором установлен PT CS> | tail -n +2 > values.yaml
```

Тайм-аут при подключении PT CS к внутренним сервисам

При подключении PT CS к внутренним сервисам балансировщик нагрузки отправляет пустой ответ с кодом HTTP, равным 0.

Проблема может возникать, если для инфраструктуры, в которой развернут PT CS, параметрам `network_timeout` и `connection_timeout` заданы значения, которых не хватает для подключения к сервисам.

Для решения проблемы вам нужно указать большие значения для параметров `network_timeout` и `connection_timeout` (например, 300 мс).

Принудительно завершается сканирование образов размером более 1,5 ГБ

Если размер образа превышает 1,5 ГБ, то процесс сканирования завершается принудительно из-за недостатка объема ОЗУ.

Проблема может возникать, если для модуля `vulnerability scanner` установлен лимит объема ОЗУ 1 ГБ.

Для решения проблемы вам нужно увеличить лимит объема ОЗУ для модуля `vulnerability scanner`. Увеличить лимит возможно изменив файл `values.yaml` и обновив PT CS для того, чтобы изменения были применены, или вручную без обновления PT CS.

Внимание! Если вы меняете параметры вручную, их значения могут быть сброшены при обновлении PT CS.

► Чтобы увеличить лимит объема ОЗУ:

1. Задайте значение объема ОЗУ в корневой секции файла `values.yaml`:

```
vuln-scanner:
  resources:
    limits:
      memory: <Значение>
```

Примечание. Значение рассчитывается как размер самого большого образа во всех репозиториях, умноженный на 1,5.

2. Выполните команду для установки нового пакета системы с параметрами `release_name` (имя релиза, всегда — `ptcs`), `chart_path` (путь к локальному файлу с Helm-чартом) и `namespace` (пространство имен, в котором будет установлен PT CS):
`helm upgrade --install <Имя релиза> <Путь к архиву с Helm-чартом> --namespace <Пространство имен, в котором установлен PT CS> --create-namespace --values values.yaml`

► Чтобы увеличить лимит объема ОЗУ вручную:

1. Выполните команду:
`kubectl edit deploy vuln-scanner -n <Пространство имен, в котором установлен PT CS>`
2. Задайте необходимое значение параметра `memory` в секции `resources` → `limits` шаблона `template` контейнера `vuln-scanner`.

Примечание. Значение рассчитывается как размер самого большого образа во всех репозиториях, умноженный на 1,5.

Не запускается менеджер очередей при использовании домена кластера, отличного от `cluster.local`

При запуске PT CS не запускается менеджер очередей RabbitMQ при использовании с доменом кластера, отличным от `cluster.local`.

Проблема может возникать, если скрипт развертывания менеджера очередей не может проверить готовность подов, выполнив пробу `readiness (readinessProbe)`, так как не может найти нужный адрес сервиса.

Для решения проблемы вам нужно указать в файле `values.yaml` в значении параметра `clusterDomain` для сервиса RabbitMQ название домена вашего кластера.

Образы не загружаются в реестр

При попытке загрузить образы в реестр появляется ошибка `Requested access to resource denied` и образы не загружаются.

Проблема может возникать, если некорректно настроены права доступа к реестру или реестр не может прочитать длинные имена образов для внешних компонентов PT CS.

Для решения проблемы вам нужно убедиться, что у используемой учетной записи достаточно прав для загрузки образов, и указать [параметр \(см. раздел 5.5\)](#) `--short-image-names` для использования короткого имени в скриптах `push_images.sh` и `generate_values.sh`.

Не открывается страница **Образы** после обновления

После обновления PT CS вы не можете перейти на страницу **Образы**, отображается ошибка 404. В журналах событий компонента `container-registry-integrator` и режиме отладки в браузере вы видите ошибку `Unexpected error: 'utf-8' codec can't decode byte 0xf6 in position 2: invalid start byte`.

Проблема может возникать, если вы регенерировали файл `values.yaml`, включая сертификаты и ключи шифрования, и использовали его для обновления. Данные о подключении к реестрам зашифрованы старым ключом.

Для обновления системы вам рекомендуется использовать существующий `values.yaml` или скопировать из него секции с ключами шифрования и сертификатами в новый файл.

Для получения доступа к странице **Образы** вы можете обновить систему с использованием корректного файла `values.yaml` (рекомендуется) или в БД PostgreSQL очистить информацию в таблице `repository` и подключить реестры заново.

Дочерний кластер не переходит в статус установки Установлен

После установки PT CS в дочернем кластере в веб-интерфейсе на странице **Кластеры** дочерний кластер остается в статусе **Не установлен** и не переходит в статус **Установлен** дольше 30 минут.

Проблема может возникать, если после установки PT CS в дочернем кластере не удалось получить лицензию, БДУ и зарегистрироваться в центральном кластере.

Для решения проблемы вам нужно подключиться к дочернему кластеру, проанализировать журналы событий модуля `ptcs-manager` и устранить обнаруженные проблемы.

► Чтобы посмотреть журнал событий модуля `ptcs-manager`:

1. Подключитесь к дочернему кластеру при помощи утилиты `kubectl` или других утилит для управления кластером.
2. Откройте журнал событий модуля `ptcs-manager`, выполнив команду:

```
kubectl logs ptcs-manager-<Имя пода ptcs-manager> -n <Пространство имен, в котором установлен PT CS>
```

Например:

```
kubectl logs ptcs-manager-68845d9955-pf761 -n ptcs
```

Выполнение команды удаления дочернего кластера не завершается

Выполнение команды удаления дочернего кластера не завершается.

Проблема может возникать, если центральный кластер был удален до выполнения команды удаления дочернего кластера.

Для решения проблемы вам нужно остановить задание (job) `unregister-pre-delete-hook` во время выполнения команды удаления.

► Чтобы удалить дочерний кластер, если выполнение команды удаления дочернего кластера не завершается,

остановите задание (job) `unregister-pre-delete-hook`, выполнив команду:

```
kubectl delete jobs/unregister-pre-delete-hook
```

Вы также можете удалить дочерний кластер принудительно.

► Чтобы удалить дочерний кластер принудительно,

выполните команду удаления с параметром `--no-hooks`:

```
helm uninstall ptcs --namespace <Пространство имен, в котором установлен PT CS> --no-hooks
```

Не удалось получить данные из дочернего кластера при переключении в веб-интерфейсе

После выбора дочернего кластера в раскрывающемся списке **Кластер** не удастся получить данные, отображается ошибка «Не удалось подключиться к кластеру. Проверьте сетевой доступ или откройте URL дочернего кластера, игнорируя предупреждение о небезопасном соединении, а затем повторите попытку».

Проблема может возникать, если отсутствует сетевой доступ к дочернему кластеру или используются самоподписанные сертификаты.

Для решения проблемы вам нужно проверить сетевой доступ и добавить сертификат в доверенные или перейти по URL дочернего кластера, добавив его в исключения безопасности или игнорируя предупреждение о небезопасном соединении, а затем — повторить попытку выбора кластера.

► Чтобы добавить сертификат в доверенные:

1. Скопируйте корневой сертификат из файла `values.yaml` или отдельного файла с сертификатами.
2. Добавьте полученный сертификат в доверенные в соответствии с инструкцией к используемой ОС.

Сервис runtime monitor не отслеживает события

Сервис runtime monitor не отслеживает события, и в журнале событий сервиса отображается ошибка «Address already in use».

Сервис runtime monitor работает в режиме hostNetwork, то есть использует сетевое пространство имен хоста. Проблема может возникать, если порт на хосте уже занят.

Для решения проблемы вы можете [задать порты для работы сервиса \(см. раздел 5.8.2\)](#) в конфигурационном файле Helm-чарта.

Сервис runtime monitor не запускается

Сервис runtime monitor не запускается, и в журнале событий отображается ошибка «Address already in use».

Проблема может возникать, если порт, используемый по умолчанию контейнерами сервиса runtime monitor, уже занят.

Для решения проблемы вы можете [изменить порт для работы контейнеров сервиса runtime monitor \(см. раздел 5.8.2\)](#) в конфигурационном файле Helm-чарта.

19. О технической поддержке

Гарантийная поддержка доступна для всех владельцев действующих лицензий на PT CS 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 гарантийной 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 и сроки ее реализации остаются на усмотрение Positive Technologies. Заявки принимаются [на портале технической поддержки](#).

Портал технической поддержки

[На портале технической поддержки](#) вы можете круглосуточно создавать и обновлять заявки и читать новости продуктов.

Для получения доступа к portalу технической поддержки нужно создать учетную запись, используя адрес электронной почты в официальном домене вашей организации. Вы можете указать другие адреса электронной почты в качестве дополнительных. Добавьте в профиль название вашей организации и контактный телефон — так нам будет проще с вами связаться.

Поддержка на портале предоставляется на русском и английском языках.

Условия предоставления гарантийной поддержки

Для получения гарантийной 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 гарантийной 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 гарантийной 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 гарантийной 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 25. 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 службы 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. Список поддерживаемых баз данных угроз и уязвимостей

PT CS поддерживает следующие базы данных угроз и уязвимостей:

- [Национальная база данных уязвимостей Национального института стандартов и технологий \(NIST\);](#)
- [банк данных угроз ФСТЭК России;](#)
- [банк данных уязвимостей ФСТЭК России;](#)
- [база данных уязвимостей для пакетов Python;](#)
- [база данных уязвимостей для пакетов Go;](#)
- [база данных уязвимостей для менеджера пакетов npm;](#)
- [база данных уязвимостей для менеджера пакетов NuGet;](#)
- [руководство по управлению уязвимостями для Ubuntu;](#)
- [обновления безопасности для «РЕД ОС»;](#)
- [оперативные обновления для Astra Linux Special Edition;](#)
- [советы по безопасности для Apache HTTP Server;](#)
- [последние исправления для ALT Linux;](#)
- [рекомендации по безопасности при использовании Debian;](#)
- [обновления безопасности для Oracle;](#)
- [рекомендации по обнаружению и предотвращению угроз в облачных сервисах Oracle;](#)
- [рекомендации по безопасности для Red Hat.](#)

Приложение Б. Список поддерживаемых платформ контейнеризации

В этом разделе приведен список поддерживаемых платформ контейнеризации. Поддержка подтверждена сертификатами совместимости. Вы можете использовать PT CS для обеспечения безопасности платформ, указанных в таблице ниже.

Таблица 26. Поддерживаемые платформы контейнеризации

Название	Правообладатель	Версии
Nova Container Platform	ООО «Орион»	Все версии
Deckhouse Kubernetes Platform	АО «Флант»	Все версии
Боцман	ООО «Платформа Боцман»	Все версии

Глоссарий

детектор (среда выполнения)

Программа, написанная на тьюринг-полном языке программирования (например, Go, Python, C, C++, Rust), скомпилированная в формат WebAssembly (WASM), которая может динамически подгружаться в систему и выгружаться без перезагрузки и позволяет реализовать безопасные и близкие по скорости к нативному коду алгоритмы проверки и анализа входящих событий.

дочерний кластер

Кластер, в котором развернут экземпляр PT CS, подключенный к центральному кластеру.

инцидент

Одно или несколько нежелательных или неожиданных событий, которые могут нарушить информационную безопасность организации.

источник (среда выполнения)

Описание логики работы для eBPF-программ на специальном языке (TracingPolicy), которые загружаются и выполняются в ядре ОС на узле и позволяют отслеживать события среды выполнения с помощью механизмов kprobe, uprobe, tracepoint и BPF-LSM ядра Linux. Источники используют описанные выше механизмы для наблюдения за активностью процессов в контейнерах (отслеживания системных вызовов и обращений к конкретным функциям ядра) и обнаружения угроз. Всегда отслеживаются события запуска и остановки процессов в контейнерах кластера.

метка

Пара «ключ — значение», присваиваемая объектам кластера (например, подам) для их идентификации, группировки и упрощения управления ими.

нода

Физическая или виртуальная машина в составе кластера, на которой запускаются поды с контейнерами и которая содержит все необходимые для их работы компоненты.

образ

Неизменяемый файл, который содержит исходный код, библиотеки, зависимости, конфигурации и инструменты, необходимые для запуска приложения. Служит шаблоном для создания Docker-контейнера.

под

Базовая единица развертывания в Kubernetes, представляющая собой один или несколько контейнеров, которые совместно используют ресурсы нод (например, сеть, диск, пространство имен пользователей).

правило реагирования

Набор условий, который определяет действия при обнаружении уязвимостей, угроз, ВПО или недействительной подписи в ходе сканирования или проверки объектов.

пространство имен

Механизм, предназначенный для логического разделения и изоляции ресурсов внутри кластера.

реестр

Стороннее хранилище, которое содержит репозитории однотипных объектов (образов).

репозиторий

Логическая часть реестра, в которой хранятся образы, объединенные общим признаком, например принадлежащие одной команде разработки.

сервис уведомлений

Подключаемый внешний сервис, который позволяет настроить отправку уведомлений по шаблонам при срабатывании правил реагирования. Примером такого сервиса может быть почтовая служба, корпоративная система учета времени и задач или корпоративная система обмена сообщениями.

сканер

Компонент PT CS, отвечающий за проверку объектов сканирования (например, образов) в системе по базе уязвимостей Positive Technologies.

угроза ИБ

Возможность нарушения информационной безопасности, в результате которого может быть нанесен ущерб организации или пользователю.

уровень опасности образа

Показатель, который определяет оценку свойств информационной безопасности образа по результатам его сканирования в PT CS.

уровень опасности уязвимости

Экспертная оценка, которая определяет степень влияния обнаруженной уязвимости на безопасность компонентов образа.

уязвимость

Наличие возможности для злоумышленников получить непосредственный доступ к коду программного продукта, его настройкам, конфигурации, возникающей в результате наличия в коде ошибок, дефектов, аномалий.

центральный кластер

Кластер, к которому подключаются дочерние кластеры и через запросы к которому осуществляется управление ими.



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.