



**Руководство  
пользователя**

# **Система управления безопасной разработкой**

• <a href="#">О платформе</a>	3
◦ <a href="#">Краткий обзор Vampy</a>	3
◦ <a href="#">С чего начать</a>	4
• <a href="#">Дашборды</a>	4
◦ <a href="#">Глобальные дашборды</a>	4
◦ <a href="#">Локальные дашборды</a>	4
◦ <a href="#">Карточки дашбордов</a>	5
■ <a href="#">Кольцевая диаграмма</a>	5
■ <a href="#">Общая информация</a>	5
■ <a href="#">Топ-5 уязвимых репозиториев</a>	5
■ <a href="#">Топ-5 разработчиков</a>	5
■ <a href="#">Фильтры по практикам</a>	5
■ <a href="#">Ретроспектива</a>	5
• <a href="#">Продукты</a>	6
◦ <a href="#">Создание продукта</a>	6
◦ <a href="#">Работа с продуктом</a>	6
◦ <a href="#">Управление доступом пользователей в продуктах</a>	7
• <a href="#">Репозитории</a>	7
◦ <a href="#">Создание репозитория</a>	7
■ <a href="#">Вручную</a>	8
■ <a href="#">Загрузка результатов</a>	8
■ <a href="#">Импорт репозитория из GitLab ★</a>	8
◦ <a href="#">Загрузка результатов сканирований</a>	8
◦ <a href="#">Удаление репозитория</a>	8
◦ <a href="#">Оценка риска</a>	9
◦ <a href="#">Управление доступом пользователей в репозиториях</a>	10
• <a href="#">Дефекты</a>	10
◦ <a href="#">Создание дефекта</a>	10
■ <a href="#">API</a>	10
■ <a href="#">Вручную</a>	10
■ <a href="#">Загрузка результатов сканирования</a>	11
◦ <a href="#">Детали дефекта</a>	11
■ <a href="#">Показать информацию об issue</a>	11
■ <a href="#">Статусы issue</a>	11
■ <a href="#">Типы статусов</a>	12
◦ <a href="#">Действия над дефектами</a>	12
■ <a href="#">Фильтры issue</a>	12
■ <a href="#">Отправка issue в Jira</a>	13
• <a href="#">Компоненты</a>	13
◦ <a href="#">Фильтры компонентов</a>	13
◦ <a href="#">Детали компонентов</a>	14
◦ <a href="#">SBOM</a>	14
• <a href="#">Активы</a>	15
◦ <a href="#">Уязвимые активы</a>	15
◦ <a href="#">Создание актива</a>	15
◦ <a href="#">Удаление актива</a>	15
◦ <a href="#">Загрузка результатов сканирований</a>	15
◦ <a href="#">Приоритет и область применения</a>	15
◦ <a href="#">Список дефектов активов</a>	15
• <a href="#">Функции</a>	15
◦ <a href="#">Список поддерживаем сканеров</a>	15
■ <a href="#">Secret Scan</a>	15
■	

	<a href="#">SAST</a>	16
	■ <a href="#">Software Composition Analysis (SCA)</a>	16
	■ <a href="#">DAST</a>	17
	■ <a href="#">IaC</a>	17
○	<a href="#">Алгоритм дедупликации</a>	17
	■ <a href="#">Примеры</a>	19
	■ <a href="#">SAST</a>	19
	■ <a href="#">SCA</a>	19
	■ <a href="#">Статусы issue</a>	19
○	<a href="#">Критерии качества (Quality gate)</a>	20
	■ <a href="#">SQG по умолчанию</a>	20
	■ <a href="#">Интерфейс</a>	20
	■ <a href="#">API</a>	21
	■ <a href="#">Примеры</a>	21
○	<a href="#">Swagger UI</a>	25
•	<a href="#">Консольная утилита vampy-cli</a>	25
○	<a href="#">Об утилите vampi-cli</a>	25
	■ <a href="#">Описание</a>	25
	■ <a href="#">Использование</a>	25
	■ <a href="#">Доступные команды</a>	25
	■ <a href="#">Глобальные параметры</a>	26
	■ <a href="#">Общие параметры</a>	26
	■ <a href="#">Параметры подключения</a>	26
	■ <a href="#">Конфигурация</a>	26
	■ <a href="#">Переменные окружения</a>	27
	■ <a href="#">Параметры командной строки</a>	27
○	<a href="#">Установка vampy-cli</a>	27
	■ <a href="#">Установка с помощью curl</a>	27
	■ <a href="#">Скачать готовые бинарные файлы</a>	27
	■ <a href="#">Использовать Docker образ</a>	27
	■ <a href="#">Конфигурация</a>	27
	■ <a href="#">Переменные окружения</a>	28
	■ <a href="#">Параметры командной строки</a>	28
○	<a href="#">Запуск сканирований</a>	28
	■ <a href="#">Запуск сканировани дефольным профилем сканирования</a>	28
	■ <a href="#">Запуск сканирования дефолтным профилем сканирования с ожиданием результатов</a>	29
	■ <a href="#">Запуск сканирования дефолтным профилем сканирования с одновременной проверкой SQG</a>	30
	■ <a href="#">Запуск выбранного сканера</a>	31
	■ <a href="#">Запуск выбранного сканера с параметрами по умолчанию</a>	31
	■ <a href="#">Запуск выбранного сканера с кастомными параметрами сканирования</a>	32
○	<a href="#">Загрузка результатов сканирований</a>	33
	■ <a href="#">Дополнительные параметры</a>	33
	■ <a href="#">Автоматическое отслеживание обработки результатов сканирования</a>	33
	■ <a href="#">Автоматическа проверка на соответствие SQG</a>	34
○	<a href="#">Работа с критериями качества (QualityGate)</a>	34
	■ <a href="#">Проверка на соответствие Quality Gate</a>	34
○	<a href="#">Получение списка репозиториев</a>	35
	■ <a href="#">Поиск репозитория по его названию</a>	36
○		

Получение списка продуктов	36
■ Поиск продукта по его названию	37
○ Получение списка фоновых задач	37
■ Проверка статусов фоновых задач	37
○ Настройка параметров вывода	37
■ Настройка параметров вывода	37
■ Стандартный	38
■ Детализированный	38
■ Расширенный	39
• Оркестрация	40
○ Предустановленные сканеры	40
■ semgrep	40
■ Trivy	40
○ Запуск сканирования	40
■ Starting a repository scan	41
■ Starting a product scan	41
○ Изменение параметров сканирований	41
■ Show scanner run options	41
■ Change scanner run options	41
■ For Docker installation	41
■ Change the list of scanners to run	41
■ Change the order and sequence of launching scanners	42
■ Change run options for semgrep scanner	42
■ Change run options for trivy scanner	42
■ Environment variables	42
■ For Kubernetes intallation	42
• Загрузка результатов сканирований	42

## О платформе

### Краткий обзор Vampry

---

Система управления безопасной разработкой (далее Vampry или Вампи).

Вампи — это самостоятельная платформа, которая помогает максимально эффективно управлять уязвимостями и может быть интегрирована в ваш жизненный цикл разработки (SDLC).

Это универсальное решение агрегирует данные о безопасности из разных источников для работы с уязвимостями и ускоряет процесс тестирования безопасности, оптимизируя рабочий процесс SDLC.

Синяя команда (Blue Team) может легко работать с выявленными проблемами, создавать задачи в Jira и отслеживать прогресс в реальном времени, а разработчики могут исправлять ошибки и обеспечивать безопасность продуктов.

## С чего начать

---

Добро пожаловать в Вампи! Ознакомьтесь со следующими ссылками, чтобы начать свою работу:

1. [Установка](#).
2. [Создание репозитория](#).
3. [Загрузка результатов сканирований](#).
4. [Интеграция с Jira](#).

Если вы не нашли ответы на свои вопросы, можете задать их сюда: [support@hexway.com](mailto:support@hexway.com).

## Дашборды

### Глобальные дашборды

---

Доступ к глобальному дашборду находится на левой панели сверху.

Он показывает несколько типов данных:

- карточки с наиболее необходимыми данными, такими как количество открытых задач;
- количество всех проектов, репозиториев и компонентов, по нажатию на которые вы будете перенаправлены в соответствующие разделы;
- топ-5 репозиториев, которые, возможно, требуют вашего внимания (выбраны по [Risk Score](#)). Нажатие на каждый элемент перенаправит вас в выбранный репозиторий;
- фильтры по практикам.

Примечание: возможность добавления собственных карточек будет реализована в будущих версиях.

### Локальные дашборды

---

Локальный дашборд имеет некоторые отличия от глобального.

1. Вы можете найти локальные дашборды на первой вкладке продукта или репозитория;
2. Главное отличие — это карточка ретроспективы. Она позволяет вам увидеть общую картину для [Risk Score](#) за последний 1 год.

Остальные [карточки](#) показывают:

- Сколько открытых задач в проекте/репозиториях;
- Сколько из них просрочено;
- Какая практика безопасности применяется.

## **Карточки дашбордов**

---

Существует несколько типов карточек на дашбордах.

Большинство элементов являются кликабельными, попробуйте!

### **Кольцевая диаграмма**

---

1. Вы можете нажать на иконку скрытия, чтобы исключить ненужные данные из колеса;
2. Наведите курсор на строку, чтобы увидеть ее подсветку на колесе;
3. Нажатие на каждый элемент перенаправит вас на связанную страницу с установленными фильтрами.

### **Общая информация**

---

На глобальном дашборде это поле показывает, сколько продуктов, репозиториев и компонентов у вас в системе. Они являются ссылками на соответствующие разделы. На локальной панели управления ссылки на продукт/репозиторий изменены на страницу задач.

### **Топ-5 уязвимых репозиториев**

---

Топ-5 репозиториев, которые, возможно, требуют вашего внимания (выбраны по [Risk Score](#)). Нажатие на каждый элемент перенаправит вас в выбранный репозиторий;

### **Топ-5 разработчиков**

---

Эта карточка считает авторов уязвимостей и отображает 5 пользователей с максимальным количеством уязвимостей.

### **Фильтры по практикам**

---

Эти фильтры можно использовать, если вы хотите увидеть задачи от определенного типа сканеров.

### **Ретроспектива**

---

Эта карточка может быть полезна для долгосрочного анализа, предоставляя информацию о том, как [Risk Score](#) продукта или репозитория изменялся с течением времени. Максимальный период, который вы можете проверить, составляет 1 год, каждый месяц считается как 30 дней. Эта информация представлена только для локальных сущностей, таких как Продукт или Репозиторий.

# Продукты

Продукты используются для группировки различных репозиторийев, которые относятся к одной теме.

## Создание продукта

---

Продукты можно найти в верхнем левом углу интерфейса. Нажмите **Create product**, чтобы добавить новый продукт.

Добавьте название продукта, установите бизнес-приоритет и добавьте описание, если необходимо.

Существует несколько способов добавить репозиторий; вы можете выбрать тот, который вам больше подходит.

Например, если вы хотите добавить несколько существующих репозиторийев, вы можете выбрать несколько файлов из списка. Также доступен поиск.

Нажмите **Add to product**, чтобы начать работать с вашим продуктом.

Вот и всё, вы создали свой первый продукт. Ура!

## Работа с продуктом

---

Посмотреть информацию о репозитории можно, нажав на кнопку со стрелочкой справа:

Здесь находится блок с общей информацией о продукте. Например, можно посмотреть полный список включенных репозиторийев, отсортировать их по критичности или добавить новые репозитории в продукт.

При нажатии на один из репозиторийев вы увидите такой же блок с общей информацией, но уже о репозитории.

Вы также можете удалить продукт, нажав на красную кнопку корзины внизу боковой панели. При удалении продукта репозитории в нём не удаляются, а просто перестают принадлежать одной группе.

Нажмите на область с названием продукта, чтобы попасть в основное окно работы с продуктом.

Главная страница продукта - это список репозиторийев. Как работать с репозиториями, можно ознакомиться [здесь](#).

## Управление доступом пользователей в продуктах

---

Основное, о чем следует знать - что права наследуются.

1. Рассмотрим самую простую ситуацию, когда у нас есть **Репозиторий А** и **Репозиторий В**, сгруппированные в **Продукт 1**. У нас также есть 2 пользователя: Алиса и Боб.

Мы вручную даем Алисе роль Editor для **Продукта 1**, а Боб получает роль Read-only. Они наследуют те же роли для **Репозитория А** и **Репозитория В** - получают кросс-роли.

Кросс-роли означают, что если мы заберем права у пользователя в **Продукте 1** или удалим **Продукт 1** вообще, то пользователь также лишится прав доступа в репозитории **А** и **В**.

2. Вторая ситуация, с которой мы можем столкнуться - это когда у нас есть **Продукт 1** и **Продукт 2**, и они оба включают в себя **Репозиторий А** и **Репозиторий В**. Пример:

Сначала мы вручную даем Бобу роль Editor для **Продукта 1** и Read-only для **Продукта 2**. Как вычислить, какие права доступа будут у Боба на репозитории? Легко, всего два правила:

1. применяется роль с большими правами;
2. ручное назначение роли переписывает кросс-роль.

Наследование также работает от репозиториев к продуктам. Например, если мы даем Алисе права Editor на **Репозиторий А**, она автоматически получит кросс-роль на всех продуктах, в которые входит **Репозиторий А**. Однако Алиса может получить только права Read-only в качестве кросс-роли для продукта, потому что доступ к продукту ей нужен лишь затем, чтобы видеть репозиторий в нём. Впрочем, как и в предыдущей ситуации, мы можем это изменить, выдав права вручную.

## Репозитории

Репозитории Vamru являются контейнерами для issues, созданных в результате процесса сканирования, завершеного в рамках пайплайна для указанного Git репозитория.

### Создание репозитория

---

Когда вы попадете на свою домашнюю страницу, вы увидите пустую страницу для репозиториев. Давайте добавим новый репозиторий.

Вы можете сделать это тремя способами:

1. [вручную](#);
2. [с помощью загрузки результатов API сканирования](#);
3. [с помощью интеграции с GitLab ★](#) (Enterprise edition).

## **Вручную**

---

1. Нажмите на кнопку в правом верхнем углу и выберите **Create manually**:
2. Затем вы увидите форму для заполнения. Вы можете установить имя, URL вашего репозитория, его критичность и, конечно, можете добавить описание.
3. После завершения нажмите **Save**. Ваш первый репозиторий готов!
4. Выпадающее меню по синей стрелке позволит вам быстро установить приоритет. Вы также можете получить доступ к настройкам репозитория, нажав на правый угол поля репозитория (красная стрелка).
5. Вы можете отсортировать репозитории здесь:
  1. ...и, конечно, искать по названиям проектов:

## **Загрузка результатов**

---

Вы можете загрузить результаты [используя API](#).

## **Импорт репозитория из GitLab ★**

---

Перед импортом репозитория из GitLab вам нужно настроить интеграцию. Вы можете найти инструкцию [здесь](#).

Чтобы импортировать проекты в hexway Vamру, просто выберите нужную интеграцию из списка

затем выберите нужный репозиторий из списка:

## **Загрузка результатов сканирований**

---

TODO

## **Удаление репозитория**

---

Чтобы удалить репозиторий:

1. Нажмите на стрелку с правой стороны репозитория, чтобы открыть его описание.
2. Нажмите на значок удаления в нижней части окна:

Обратите внимание, что вы можете удалить только репозитории без issue. Если ваш репозиторий не пуст, сначала вам нужно удалить все issue из него.

Чтобы удалить ветку в репозитории:

1. Нажмите на стрелку с правой стороны репозитория, чтобы открыть его описание, и прокрутите вниз.
2. Наведите курсор на ветку, которую хотите удалить, и справа появится иконка.

После подтверждения действия ветка будет удалена.

Обратите внимание, что вы не можете удалить ветку по умолчанию и ветки, содержащие issue.

## Оценка риска

---

У каждого репозитория есть уровень риска - **Risk Score**. Он высчитывается по следующей формуле:

где:

- **Multiplier** - коэффициент, зависящий от критичности issue. Значения по умолчанию:
  - Critical = 10
  - High = 5
  - Medium = 2
  - Low = 1
- **Issues** - количество issue с этим уровнем критичности
- **Criticality** - уровень критичности репозитория выставляется вручную

Если репозиторий не был создан автоматически, вы можете добавить его вручную, нажав на **+Repository** в правом верхнем углу страницы. Вам нужно указать следующие параметры:

- **Repository name** - название вашего репозитория;
- **URL** - адрес удалённого репозитория;
- **Slug** - уникальный идентификатор репозитория, обычно в виде строки, содержащей только строчные буквы и дефисы;
- **Description** - опциональное поле для добавления описания репозитория.

После заполнения всех необходимых полей нажмите **Create**, чтобы создать репозиторий.

## Управление доступом пользователей в репозиториях

---

Чтобы перейти к меню управления, есть два способа:

1. Нажмите на аватар любого участника команды;
2. Или нажмите на стрелку справа и прокрутите вниз до раздела "Команда".

В обоих случаях вы увидите изображение, похожее на это:

В синей форме вы можете изменить роль пользователя или удалить пользователя из репозитория. Нажмите на кнопку  в красной форме, чтобы добавить нового пользователя.

## Дефекты

### Создание дефекта

---

#### API

---

Вы можете загрузить дефекты [используя API](#).

#### Вручную

---

1. Нажмите на кнопку **+Issue**.
2. Вы увидите интерфейс с некоторыми полями для заполнения.

**CVE** - [Система Общих Уязвимостей и Экспозиций](#) предоставляет ссылочный метод для публично известных уязвимостей в области информационной безопасности.

**CWE** - [Перечень Общих Уязвимостей](#) представляет собой систему категорий для аппаратных и программных уязвимостей и слабостей.

3. Как только вы дадите вашей issue название, вы сможете увидеть её превью.
4. Если вам нравится то, что вы видите, нажмите **Create**. Если нет, нажмите Назад и измените все, что хотите.

⚠ Вы не сможете изменить проблему после её создания!

Нажмите в верхнем левом углу, чтобы скрыть блок с информацией об issue:

## Загрузка результатов сканирования

1. Чтобы загрузить результат сканирования, используйте кнопку **Scan**:
2. Вам будет предложено выбрать сканер (например, Semgrep):
3. И ветку (вы можете выбрать одну из существующих или создать новую):
4. После того как вы выберете файл сканирования, нажмите **Submit**, чтобы загрузить файл в систему. Через несколько мгновений (в зависимости от размера файла) ваши проблемы будут загружены.

## Детали дефекта

Issue — это уязвимости, обнаруженные в системе в результате сканирования.

Как только вы получите первые issue, импортировав файл сканирования, вы сможете:

- просмотреть описание issue;
- применять фильтры и сортировать их;
- отправлять issue в Jira и просматривать статусы из Jira.

## Показать информацию об issue

Чтобы просмотреть информацию об issue:

1. В левом меню выберите **Issues**.
2. Один раз щелкните на issue. Описание появится в правом столбце:

На странице issue вы также можете переключаться между ними, используя стрелки **Previous** и **Next** в правом верхнем углу проблемы.

Нажмите **Close**, чтобы вернуться к списку issue.

## Статусы issue

Только администратор платформы может добавлять и редактировать статусы issue на платформе.

Статус можно управлять непосредственно из списка issue.

1. Войдите в платформу как администратор;
2. Откройте список issue;
3. Нажмите на любой **Status** issue:

Вы можете переименовать статус, нажав на значок редактирования.

Вы также можете редактировать статусы в карточках issue:

Или выберите несколько issue в списке и измените их статусы:

### Типы статусов

Существует 2 типа статусов: **Открытые** и **Закрытые**.

#### Открытые:

Статус	Описание
New	Новая issue
Recurrent	Issue, которая была найдена при повторном скане
Reopened	Issue, которая была закрыта, но найдена снова
Confirmed	Issue была подтверждена
Check required	Issue требует повторной проверки

#### Закрытые:

Статус	Описание
Risk accepted	Issue не будет исправлена, и вы принимаете риски
Fixed	Issue была исправлена и закрыта
Won't fix	Issue не будет исправлена и закрыта (возможно, ложноположительная или несущественная)

Триггеры, при которых меняется статус:

- Если та же самая issue найдена снова, **New** меняется на **Recurrent**.
- Если issue не была найдена при втором сканировании, статусы **New**, **Recurrent**, **Reopened** и **Confirmed** меняются на **Fixed**.
- Если issue, которая была **Fixed**, найдена при втором сканировании, она переходит в статус **Reopened**.

## Действия над дефектами

### Фильтры issue

Как только вы войдете в репозиторий, вы увидите список issue. Этот список можно гибко настраивать под свои задачи.

1. **Фильтры.** Оно может фильтровать по **Title**, **File path**, **CVE**.
2. **Отображаемые колонки.** Выберите с помощью флажков, какие колонки вы хотите отобразить.
3. **Порядок сортировки.** Вы можете выбрать несколько колонок для сортировки ваших issue. Число рядом с иконкой показывает приоритет сортировки; вы также можете выбрать порядок по возрастанию/убыванию, нажав на иконку.

4. **Статус issue.** Нажмите на статусы, чтобы отображать только issue с выбранными статусами.
5. **Расширенный интерфейс фильтров.** Нажмите кнопку, чтобы увидеть больше доступных фильтров.

Все выбранные вами фильтры отображаются в верхней части окна, вы можете удалить один фильтр или полностью сбросить все фильтры. В этом окне вы также можете фильтровать проблемы по **Branch**, **Tool**, который был использован для сканирования, наличию **Solution**.

## Отправка issue в Jira

---

Примечание: для использования этой опции ваш проект должен быть подключен к Jira (см. раздел [Интеграции](#)).

Чтобы отправить issue в Jira:

1. В левом меню выберите **Issues**.
2. Выберите одну или несколько issue в списке, используя флажки.
3. Нажмите **Task** и выберите имя вашего подключения к Jira:
4. Вы можете просмотреть детали, заполнить дополнительные поля, а затем нажать **Send**:

Вы можете отправить одну issue в несколько проектов Jira одновременно. Чтобы увидеть все задачи Jira, созданные из одних и тех же issue, нажмите на **Tasks** в списке issue:

Или нажмите на issue, чтобы увидеть список в детальном описании:

## Компоненты

Список компонентов (сторонних компонентов, библиотек), используемых в указанном [репозитории](#).

Список заполняется автоматически в результате процесса сканирования, завершеного в рамках пайплайна.

## Фильтры компонентов

---

Список компонентов (сторонние компоненты, библиотеки), которые используются в указанном [репозитории](#).

Список заполняется автоматически в результате процесса сканирования, завершеного в рамках пайплайна.

1. **Filter field.** Работает по имени и версии.
2. **Displayed columns.** Выберите с помощью флажков, какие столбцы вы хотите отображать.

3. **Sorting order.** Вы можете выбрать некоторые столбцы для сортировки ваших проблем. Номер рядом с иконкой показывает приоритет порядка; вы также можете выбрать порядок ASC или DESC, нажав на иконку. Этот фильтр сохраняется, когда вы переключаетесь между репозиториями.
4. **SBOM.** Вы можете экспортировать или импортировать файл в формате JSON, содержащий SBOM.

Также есть отдельная страница для компонентов - доступ к ней находится на панели слева.

Здесь вы также можете отфильтровать по **Risk score**, **Name** или по дате создания/обновления.

При нажатии на любой компонент вы перейдете в список проблем, вызванных этим компонентом.

## Детали компонентов

---

Для каждого компонента представлены следующие данные:

- **Component name** - ну, вы знаете :)
- **Severity** - общее количество проблем для каждого уровня серьезности
- **Versions** - показывает используемую версию компонента и актуальную версию
- **Risk Score** - рассчитывается по формуле:

где:

- **Multiplier** - множитель, назначенный каждому уровню критичности. Значения по умолчанию:
  - Критический = 10
  - Высокий = 5
  - Средний = 2
  - Низкий = 1
- **Issues** - количество проблем с этим уровнем риска, найденных для данного компонента

## SBOM

---

Если вам необходимо работать со SBOM, вы можете найти файл здесь:

Вы можете экспортировать JSON или загрузить собственный файл. Импорт и экспорт происходит в **CycloneDX** формате.

# АКТИВЫ

## Уязвимые активы

---

TODO

## Создание актива

---

TODO

## Удаление актива

---

TODO

## Загрузка результатов сканирований

---

TODO

## Приоритет и область применения

---

TODO

## Список дефектов активов

---

TODO

# ФУНКЦИИ

## Список поддерживаем сканеров

---

[Здесь инструкция по загрузке результатов сканирования.](#)

### Secret Scan

---

Название сканера	VV_SCAN_TYPE	Поддерживаемые языки	Ссылка
Trufflehog3	TRUFFLEHOG3	All	<a href="#">Github</a>
Gitlab Secret detection	GITLAB_SECRET_DETECT	All	<a href="#">Gitlab</a>
Trufflehog	TRUFFLEHOG	All	<a href="#">Github</a>

**SAST**

Название сканера	VV_SCAN_TYPE	Поддерживаемые языки	Ссылка
Semgrep	SEMGREP	C#, Go, Java, JavaScript, JSX, JSON, PHP, Python, Ruby, Scala, Terraform, TypeScript, TSX	<a href="#">Github</a>
ESLint	ESLINT	ECMAScript, JavaScript	<a href="#">Github</a>
Checkmarx	CHECKMARX	C#, Go, Java, JavaScript, JSX, JSON, PHP, Python, Ruby, Scala, Terraform, TypeScript, TSX	<a href="#">Checkmarx</a>
GitLab SAST	GITLAB_SAST	NET Core, .NET Framework, Go, Java, Python, React, JavaScript, TypeScript	<a href="#">Gitlab</a>
Snyk Code	SNYK_CODE	.NET, Apex, Bazel, C/C++, Elixir, Go, Java and Kotlin, JavaScript, PHP, Python, Ruby, Scala, Swift and Objective-C, TypeScript, VB.NET	<a href="#">Snyk.io</a>
SARIF	SARIF	VCE	<a href="#">SarifW</a>
Sonarqube	sonarqube	VCE	<a href="#">sonars</a>
PT AI	PT_AI	VCE	<a href="#">PT AI</a>
AppScreener	SARIF	VCE	<a href="#">rt-solar.n</a> <a href="#">solar.n</a> <a href="#">solar_a</a>

**Software Composition Analysis (SCA)**

Название сканера	VV_SCAN_TYPE	Тип сканируемых объектов
Trivy Image	TRIVY_IMAGE	OS packages, software dependencies
Trivy File system	TRIVY_FS	All
Dependency Check	DEPENDENCY_CHECK	software dependencies
Grype	GRYPE	OS packages (Alpine, Amazon Linux, CentOS, Debian, Distroless, Oracle Linux (RHEL), Ubuntu), software dependencies (Ruby, Java, JavaScript, Python, Dotnet, Golang)
Snyk SCA	SNYK_SCA	OS packages (Alpine, Amazon Linux, CentOS, Debian, Distroless, Oracle Linux (RHEL), Ubuntu), software dependencies (Ruby, Java, JavaScript, Python, Dotnet, Golang)
Gitlab Container security	GITLAB_CONTAINER_SECURITY	OS packages, software dependencies

Название сканера	VV_SCAN_TYPE	Тип сканируемых объектов
Gemnasium	GEMNASIUM	OS packages, software dependencies
PT AI	PT_AI	software dependencies
CodeScoring	SARIF	OS packages, software dependencies

## DAST

---

Название сканера	VV_SCAN_TYPE	Тип сканируемых объектов	Ссылка
Nuclei	NUCLEI	Network, Web	<a href="#">Github</a>
ZAP	ZAP	Network, Web	<a href="#">ZAP</a>
Gitlab DAST	GITLAB_DAST	Network, Web	<a href="#">Gitlab DAST</a>
Acunetix	ACUNETIX	Network, Web	<a href="#">Acunetix</a>
Burp	BURP	WEB	<a href="#">Burp</a>
PT AI	PT_AI	WEB	<a href="#">PT AI</a>

## IaC

---

Название сканера	VV_SCAN_TYPE	Тип сканируемых объектов	Link
kics	KICS	IaC	<a href="#">Github</a>

## Алгоритм дедупликации

---

**Finding** — это результат одного сканирования, который содержит информацию о уязвимости. **Finding** считается родительским, если это является первой находкой для уязвимости, она создает **issue**.

**Issue** — это сущность в Vampry, которая содержит всю информацию о уязвимости, полученную из **finding**.

**Finding** считается **duplicate**, если во время импорта и работы алгоритма дедупликации была найдена совпадающая существующая **issue**. В этом случае дубликат **finding** связывается с **issue**.

Если дубликат определен, он связывается с **issue** и отображается в разделе Duplicate. Данные в **issue** берутся из родительской **finding**, дубликаты **finding** не обновляют их. Единственное поле, которое может быть обновлено — это **solution**.

**Дедубликация** — это алгоритм, который помогает Vampry распознавать идентичные **issues** в результатах сканирования.

Результат сканирования содержит несколько **findings**, которые включают в себя несколько полей, в зависимости от сканера. Обычно все сканы содержат **repository name**, **branch name** и внутренний дубликат отпечатка сканера — **SD ID**, но некоторые поля специфичны для различных видов сканеров.

SAST	SCA	DAST
CVE	CVE	CVE
CWE	CWE	CWE
file path	file path	IP
file line number	file line number	domain name
code snippet	library name	
	library version	

Основная проблема заключается в том, что не все сканеры заполняют эти поля, поэтому иногда мы можем получить несколько **finding** от разных сканеров, которые в основном ссылаются на одну и ту же **issue**. Для каждой **finding** Vampry создает несколько хешей на основе полей, которые содержит данная **finding**. Они используются для сравнения **findings** с существующими **issues**, чтобы выяснить, получили ли мы дубликат или нам нужно создать новую **issue**.

Таким образом, мы создаем следующие хеши для каждой **finding**

Приоритет для сравнения	SAST	SCA	DAST
1	repository + branch + SD ID (required SD ID)	repository + branch + SD ID (required SD ID)	repository + branch + SD ID (required SD ID)
2	repository + branch + CVE + CWE + filepath + code snippet (необходимы поля filepath и code snippet)	repository + branch + CVE + CWE + filepath + file line number + library name + library version (необходимы поля library name и library version)	repository + branch + CVE + CWE + domain name + IP address (необходимо поле CVE)
3	repository + branch + CVE + CWE + filepath + file line number (необходимы поля filepath и file line number)	repository + branch + CVE + CWE + filepath + file line number + library name (необходимы поля library name и library version)	repository + branch + CVE + CWE + title

Приоритет для сравнения	SAST	SCA	DAST
		пустое для поиска поле library version)	
4	repository + branch + CVE + CWE + filepath + file line number + code_snippet + title	repository + branch + CVE + CWE + filepath + file line number + library name + library version + title	

Когда новые **findings** связываются с существующей **issue**, то и их хеши также связываются. В итоге мы имеем **issue**, которая имеет несколько хешей, полученных из всех **findings**, связанных с ней.

## Примеры

---

### SAST

---

Допустим, у нас есть 2 **findings**: **A** в качестве родительской и **B** как **finding**, которую нужно проверить на дубликат. У них есть все поля для SAST, **A** была найдена с помощью Semgrep, а **B** — с помощью Checkmarx. Как это выглядит в нашей системе? Красные стрелки показывают путь для наших примеров.

Здесь мы видим, что они не совпадают, поскольку у них разные отпечатки, полученные от разных сканеров. Но когда мы проверяем другие поля, мы понимаем, что это одна и та же **issue**.

### SCA

---

Мы нашли **Finding A**, с заполненными полями. Он описывает уязвимость в кусочке кода, поддерживаемом библиотекой версии 1.0. Мы обновили версию библиотеки до 1.1, но уязвимость никуда не делась, и следующий скан выявляет **Finding B**.

Схема выглядит довольно просто.

## Статусы issue

---

В основном существует 4 статуса для **issues**:

- New Issue,
- Recurrent,
- Fixed,
- Reopened.

Когда мы загружаем новый скан и алгоритм находит дубликаты, он превращает **New Issue** в **Recurrent** и **Fixed** в **Reopened**.

Когда алгоритм не находит дубликатов для существующих **issues**, он устанавливает все из них в статус **Fixed**.

## Критерии качества (Quality gate)

---

SQG (Security Quality Gate) - это контроль качества, используется для проверки безопасности продукта или репозитория.

### SQG по умолчанию

---

На текущий момент в Vampru доступны 5 видов SQG - по умолчанию, низкий, средний, высокий и критический, в соответствии с критичностью продукта/репозитория.

Severity	Practice	Порог количества issue (по умолчанию)	Порог количества issue (низкий)	Порог количества issue (средний)	Порог количества issue (высокий)	критический
Критический	SAST	0	10	5	0	
Критический	DAST	0	10	5	0	
Критический	SCA	0	20	10	0	
Высокий	SAST	0	20	10	5	
Высокий	DAST	0	20	10	5	
Высокий	SCA	10	80	40	20	
Средний	SAST	5	40	20	10	
Средний	DAST	5	40	20	10	
Средний	SCA	20	160	80	40	
Низкий	SAST	10	80	40	20	
Низкий	DAST	10	80	40	20	
Низкий	SCA	25	200	100	50	

Примечание: в будущих версиях появится возможность создавать кастомные SQG.

## Интерфейс

---

Когда вы открываете Продукты или Репозитории, вы можете увидеть под именем каждого элемента, прошел ли он контроль качества. Нажмите на стрелку справа, чтобы открыть боковую панель информации о элементе и проверить детали.

## API

---

Чтобы убедиться, что продукт/репозиторий прошёл контроль качества:

- Создайте пользователя-бота с правами Админ или Редактор и запишите сгенерированный токен (или обратитесь за этим к администратору платформы). Добавьте его в нужный репозиторий.
- Запустите следующий запрос CURL:

```
curl -G "$SEC_VV_URL/ext/v1/quality_gate/" \
-H "Authentication: $SEC_VV_KEY" \
-d "product=$VAMPY_PRODUCT_SLUG" \
-d "repository=$VAMPY_REPO_SLUG" \
```

где

- **SEC\_VV\_UR** - путь до сервера Vampy, например: `http://vmp-demo.corp.hexway.io/api/`;
- **SEC\_VV\_KEY** - API токен бот-пользователя;
- **VAMPY\_PRODUCT\_SLUG** - Слаг продукта, который проходит проверку качества;
- **VAMPY\_REPO\_SLUG** - Слаг продукта, который проходит проверку качества. Равносильно `CI_PROJECT_PATH`;

△ Используйте только один слаг: для продукта или для репозитория, иначе вы получите ошибку.

В результате вы должны получить статус для контроля качества, выбранный в зависимости от критичности продукта/репозитория.

Статус контроля качества будет одним из 4-х вариантов:

- **IN PROGRESS** - этот статус означает, что SQG рассчитывается, и вам следует проверить его позже. После обработки изменений по одному из следующих пунктов;
- **FAILED** - это означает, что некоторые условия не были выполнены. Вы можете проверить ответ для получения деталей;
- **SKIPPED** - этот статус применяется в основном для условий, которые вы отключили. Если все условия были отключены, контроль качества будет автоматически пропущен.
- **PASSED** - процесс завершен успешно, контроль качества пройден.
- **ERROR** - возникла какая-то техническая проблема.

## Примеры

---

Запрос:

```
bash curl -G "172.16.3.198/api/ext/v1/quality_gate/" \ -H
"Authentication:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImJvdF91c2VyIiwiaWF0Ij
\ -d "product=default" \
```

ОТВЕТ:

```
{
  "id": "fb716e80-dab3-48bb-885b-de0c136549ff",
  "status": "FAILED",
  "lastCalculated": "2024-10-07T03:00:59.178413Z",
  "relation": {
    "name": "groenendael",
    "slug": "product-7b66bed",
    "id": "c43beb36-2079-43bf-8996-c940fd3e05fc",
    "type": "PRODUCT"
  },
  "details": {
    "name": "Default Quality Gate for criticality \"HIGH\"",
    "slug": "default-quality-gate-high",
    "criticality": "HIGH",
    "conditions": [
      {
        "enabled": true,
        "scannerTypes": [
          "SAST"
        ],
        "severities": [
          "CRITICAL"
        ],
        "maxValue": 0,
        "currentValue": 0,
        "status": "PASSED"
      },
      {
        "enabled": true,
        "scannerTypes": [
          "DAST"
        ],
        "severities": [
          "CRITICAL"
        ],
        "maxValue": 0,
        "currentValue": 0,
        "status": "PASSED"
      },
      {
        "enabled": true,
        "scannerTypes": [
          "SCA"
        ],
        "severities": [
          "CRITICAL"
        ],
        "maxValue": 0,
        "currentValue": 0,
        "status": "PASSED"
      }
    ]
  }
}
```

```
},
{
  "enabled": true,
  "scannerTypes": [
    "SAST"
  ],
  "severities": [
    "HIGH"
  ],
  "maxValue": 5,
  "currentValue": 4,
  "status": "PASSED"
},
{
  "enabled": true,
  "scannerTypes": [
    "DAST"
  ],
  "severities": [
    "HIGH"
  ],
  "maxValue": 5,
  "currentValue": 0,
  "status": "PASSED"
},
{
  "enabled": true,
  "scannerTypes": [
    "SCA"
  ],
  "severities": [
    "HIGH"
  ],
  "maxValue": 20,
  "currentValue": 0,
  "status": "PASSED"
},
{
  "enabled": true,
  "scannerTypes": [
    "SAST"
  ],
  "severities": [
    "MEDIUM"
  ],
  "maxValue": 10,
  "currentValue": 80,
  "status": "FAILED"
},
{
  "enabled": true,
  "scannerTypes": [
```

```
    "DAST"
  ],
  "severities": [
    "MEDIUM"
  ],
  "maxValue": 10,
  "currentValue": 0,
  "status": "PASSED"
},
{
  "enabled": true,
  "scannerTypes": [
    "SCA"
  ],
  "severities": [
    "MEDIUM"
  ],
  "maxValue": 40,
  "currentValue": 0,
  "status": "PASSED"
},
{
  "enabled": true,
  "scannerTypes": [
    "SAST"
  ],
  "severities": [
    "LOW"
  ],
  "maxValue": 20,
  "currentValue": 106,
  "status": "FAILED"
},
{
  "enabled": true,
  "scannerTypes": [
    "DAST"
  ],
  "severities": [
    "LOW"
  ],
  "maxValue": 20,
  "currentValue": 0,
  "status": "PASSED"
},
{
  "enabled": true,
  "scannerTypes": [
    "SCA"
  ],
  "severities": [
    "LOW"
  ]
}
```

```
    ],  
    "maxValue": 50,  
    "currentValue": 0,  
    "status": "PASSED"  
  }  
],  
"errorMessage": ""  
}  
}
```

## Swagger UI

---

TODO

# Консольная утилита `vampry-cli`

## Об утилите `vampry-cli`

---

### Описание

---

`vampry-cli` — это утилита командной строки для взаимодействия с сервером `Vampry`. Она предоставляет возможность управления репозиториями, продуктами, процессами сканирования и другими функциями через простые команды.

`vampry-cli` легко использовать в конвейерах CI/CD для запуска сканирования и прерывания релизов, если сканирование не удалось или результаты сканирования не соответствуют указанным критериям релиза.

### Использование

---

`./vampry-cli` [глобальные параметры] команда [параметры команды]

### Доступные команды

---

- **upload**  
Загружает результаты сканирования, которые уже существуют.
- **scan**  
Запускает процесс сканирования указанным сканером для указанного репозитория.
- **quality-gate**  
Отображает результаты `QualityGate` для выбранного продукта или репозитория.
- **bg-task**  
Проверяет статус фоновой задачи и выводит подробности.

- **products**  
Получает список продуктов.
- **repositories**  
Получает список репозиториев.
- **help, h**  
Показывает список доступных команд или справку по одной из команд.

## Глобальные параметры

### Общие параметры

Параметр	Описание	Значение по умолчанию
--help, -h	Показывает справку.	
--details	Показывает детализированный вывод для запрошенного действия (например, таблицу с результатами QualityGate).	false
--verbose	Показывает дополнительный вывод (например, выполнение каждого шага действия).	false
--version, -v	Выводит только версию программы.	false

### Параметры подключения

Параметр	Описание	Значение по умолчанию
--api-token value	API-токен для подключения к серверу Vampy (или значение из переменной окружения).	значение из переменной \$VAMPY_API_TOKEN
--api-version value	Версия API Vampy.	v1
--timeout value	Таймаут в секундах.	120
--vampy-url value	URL-адрес сервера Vampy (или значение из переменной окружения).	значение из переменной \$VAMPY_URL

## Конфигурация

Для использования vampy-cli необходимо определить два **обязательных** параметра подключения:





```

├─ Scanner: TRIVY_FS
├─ DockerImage: registry.hexway.io/hexway/vampy-scan-images/trivy:lates
└─ Command: trivy fs --scanners vuln,secret,misconfig -f json --output

```

Task: <https://vampy.hexway.io/admin/bgTasks?23=3df2a2f5-2dfd-4445-89c5-632>  
 Status: PENDING

## Запуск сканирования дефолтным профилем сканирования с ожиданием результатов

Если вы просто запускаете сканирование, то необходимо периодически проверять его состояние, так как сканирование запускается не сразу, а в порядке очереди.

Для того чтобы вам этого не делать мы реализовали параметр `--check-task`

```
vampy-cli scan --repository vmp/product/vampy-engine --check-task --detail
```

```
-----
Scan process for repository "vmp/product/vampy-engine" ...
```

```
-----
  Scanning added to the queue and will be executed soon
Repository: [CRITICAL]Vampy Engine
Link: https://vampy.hexway.io/repositories/1234494-738c-404b-aa1b-14a8959d
ScanProfile "Default" | RunMode: "CONCURRENT"
Commands:
```

```
-----
├─ Semgrep
├─ Scanner: SEMGREP
├─ DockerImage: registry.hexway.io/hexway/vampy-scan-images/semgrep:lat
└─ Command: semgrep scan --json --output "${RESULTS_FILE}" --config=p/o
├─ Trivy: FileSystem
├─ Scanner: TRIVY_FS
├─ DockerImage: registry.hexway.io/hexway/vampy-scan-images/trivy:lates
└─ Command: trivy fs --scanners vuln,secret,misconfig -f json --output

```

Task: <https://vampy.hexway.io/admin/bgTasks?id=30994826-d818-482c-aea5-871>  
 Status: PENDING

```
-----
Checking Background Task 30994826-d818-482c-aea5-8715d1db3c28 for repository
```

```
-----
  Background Task: 'Scan repository 'Vampy Engine''
Link: https://vampy.hexway.io/admin/bgTasks?id=30994826-d818-482c-aea5-871
Status: SUCCESS
SubTasks:
```

NAME	STARTED	FIN
Prepare data for orchestration	2024-12-18T04:49:56.612932Z	202

Execute orchestration task	2024-12-18T04:49:57.463707Z	202
Await orchestration service callback	2024-12-18T04:50:44.098105Z	

## Запуск сканирования дефолтным профилем сканирования с одновременной проверкой SQG

Так же вы можете сразу после выполнения сканирования автоматически проверять репозиторий или продукт на соответствие выбранным критериям качества (Security Quality Gates)

Для этого запустите `vampy_cli` с параметрами `--check-qq` или `--check-full`

```
vampy-cli scan --repository vmp/product/vampy-engine --check-full --detail
```

```
-----
Scan process for repository "vmp/product/vampy-engine" ...
```

```
-----
  Scanning added to the queue and will be executed soon
```

```
Repository: [CRITICAL]Vampy Engine
```

```
Link: https://vampy.hexway.io/repositories/12344494-738c-404b-aa1b-14a8959
```

```
ScanProfile "Default" | RunMode: "CONCURRENT"
```

```
Commands:
```

```
-----
├─ Semgrep
│   ├── Scanner: SEMGREP
│   ├── DockerImage: registry.hexway.io/hexway/vampy-scan-images/semgrep:late
│   └── Command: semgrep scan --json --output "${RESULTS_FILE}" --config=p/o
├─ Trivy: FileSystem
│   ├── Scanner: TRIVY_FS
│   ├── DockerImage: registry.hexway.io/hexway/vampy-scan-images/trivy:lates
│   └── Command: trivy fs --scanners vuln,secret,misconfig -f json --output
```

```
Task: https://vampy.hexway.io/admin/bgTasks?id=ae898310-ed27-42d1-b076-498
```

```
Status: PENDING
```

```
-----
Checking Background Task ae898310-ed27-42d1-b076-498322ced2f0 for reposito
```

```
-----
  Background Task: 'Scan repository 'Vampy Engine''
```

```
Link: https://vampy.hexway.io/admin/bgTasks?id=ae898310-ed27-42d1-b076-498
```

```
Status: SUCCESS
```

```
SubTasks:
```

NAME	STARTED	FIN
Prepare data for orchestration	2024-12-18T04:56:00.537908Z	202
Execute orchestration task	2024-12-18T04:56:01.190371Z	202
Await orchestration service callback	2024-12-18T04:56:47.906176Z	

```
-----
Checking SQG for repository 'vmp/product/vampy-engine' ...
-----
```

```
QualityGate: 'Critical'
Repository: 'Vampy Engine'
Link: https://vampy.hexway.io/repositories/lefd4494-738c-404b-aalb-14a8959
Status: FAILED
Results:
```

STATUS	SCANNER TYPE	SEVERITIES	MAX VALUE	CURRENT VALUE
PASSED	SAST	CRITICAL	0	0
PASSED	DAST	CRITICAL	0	0
PASSED	SCA	CRITICAL	0	0
FAILED	SAST	HIGH	0	1
PASSED	DAST	HIGH	0	0
PASSED	SCA	HIGH	10	10
PASSED	SAST	MEDIUM	5	0
PASSED	DAST	MEDIUM	5	0
PASSED	SCA	MEDIUM	40	36
PASSED	SAST	LOW	10	0
PASSED	DAST	LOW	10	0
PASSED	SCA	LOW	100	67

## Запуск выбранного сканера

Вы можете выполнять сканирования не только группой сканеров, используя профили сканирования, но так же запускать один выбранный сканер.

## Запуск выбранного сканера с параметрами по умолчанию

Запустим semgrep с параметрами по-умолчанию (--config=p/owasp-top-ten --metrics=off), дождемся завершения сканирования и сразу проверим SQG

```
vampy-cli scan --repository vmp/product/vampy-engine --scanner SEMGREP --c
```

```
-----
Scan process for repository "vmp/product/vampy-engine" | scanner "SEMGREP"
-----
```

```
Scanning added to the queue and will be executed soon
Repository: [CRITICAL]Vampy Engine
Link: https://vampy.hexway.io/repositories/lefd1234-738c-404b-aalb-14a8959
ScanProfile "Specific Scanners" | RunMode: "CONCURRENT"
Commands:
```

```
-----
— Semgrep
```

- └─ Scanner: SEMGREP
- └─ DockerImage: registry.hexway.io/hexway/vampy-scan-images/semgrep:lat
- └─ Command: semgrep scan --json --output "\${RESULTS\_FILE}" --config=p/o

Task: <https://vampy.hexway.io/admin/bgTasks?id=123fd3b9-fb21-487b-bf61-ca1>  
 Status: PENDING

-----  
 Checking Background Task 968fd3b9-fb21-487b-bf61-ca1a62bc89e3 for repository  
 -----

Background Task: 'Scan repository 'Vampy Engine''  
 Link: <https://vampy.hexway.io/admin/bgTasks?id=123fd3b9-fb21-487b-bf61-ca1>  
 Status: SUCCESS  
 SubTasks:

NAME	STARTED	FIN
Prepare data for orchestration	2024-12-18T05:05:44.636376Z	202
Execute orchestration task	2024-12-18T05:05:45.321419Z	202
Await orchestration service callback	2024-12-18T05:06:18.484481Z	

## Запуск выбранного сканера с кастомными параметрами сканирования

Вы можете запустить выбранный вами сканер и **сами** определить параметры его запуска

Для этого выполните используйте опцию --scanner и --scanner-options

Запустим semgrep с параметрами --config=p/ai.python.detect-openai.detect-openai

```
vampy-cli scan --repository vmp/product/vampy-engine --scanner SEMGREP --s
```

```
-----  

Scan process for repository "vmp/product/vampy-engine" | scanner "SEMGREP"  

-----
```

```
Scanning added to the queue and will be executed soon  

Repository: [CRITICAL]Vampy Engine  

Link: https://vampy.hexway.io/repositories/1efd4123-738c-404b-aa1b-14a8959  

ScanProfile "Specific Scanners" | RunMode: "CONCURRENT"  

Commands:
```

- ```
-----  

└─ Semgrep
  └─ Scanner: SEMGREP
  └─ DockerImage: registry.hexway.io/hexway/vampy-scan-images/semgrep:lat
  └─ Command: semgrep scan --json --output "${RESULTS_FILE}" --config=p/a
```

Task: <https://vampy.hexway.io/admin/bgTasks?id=3123225e-a41c-4950-a474-23c>  
 Status: PENDING

-----  
 Checking Background Task 3b95225e-a41c-4950-a474-23c457638a16 for repository  
 -----

Background Task: 'Scan repository 'Vampy Engine''  
 Link: <https://vampy.hexway.io/admin/bgTasks?id=3123225e-a41c-4950-a474-23c457638a16>  
 Status: PROGRESS | unable to finish after 121 seconds  
 SubTasks:

| NAME                                 | STARTED                     | FINISHED                    |
|--------------------------------------|-----------------------------|-----------------------------|
| Prepare data for orchestration       | 2024-12-18T05:16:52.930801Z | 2024-12-18T05:16:53.727320Z |
| Execute orchestration task           | 2024-12-18T05:16:53.727320Z | 2024-12-18T05:16:53.727320Z |
| Await orchestration service callback |                             |                             |

## Загрузка результатов сканирований

Вы можете загрузить результаты сканирования любого из поддерживаемых сканеров в Vampy используя команду upload

Для этого необходимо указать:

- репозиторий или продукт (--repository/--product)
- файл для загрузки (--file)
- имя сканера, результаты которого загружаются (--scanner)

Пример загрузки результатов сканирования сканера TRIVY репозитория vmp/product/vampy-engine

```
vampy-cli upload --repository vmp/product/vampy-engine --file ~/scans_trivy
ScanUpload: 0101e9e8-a49c-49d2-b7fb-a129bfcc9d3e
```

В результате вы получите id фоновой задачи, статус которой необходимо отслеживать для получения информации об успешности/не успешности результатов загрузки сканирования

## Дополнительные параметры

### Автоматическое отслеживание обработки результатов сканирования

Используя опцию --check-full вы можете автоматически отслеживать обработку результатов сканирования

```
vampy-cli upload --repository vmp/product/vampy-engine --file ~/scans_trivy
ScanUpload: e75c4c39-8c15-4166-8711-f5fefc398816
QG: FAILED
```

## Автоматическая проверка на соответствие SQG

Используя опцию `--check-qq` вы автоматически получите результаты прохождения Security Quality Gate после загрузки результатов сканирования

```
vampy-cli upload --repository vmp/product/vampy-engine --file ~/scans_trivy
```

```
-----
Run scan result uploading for repository vmp/product/vampy-engine | scanned
-----
```

```
  Scan successful uploaded
Parser: TRIVY_IMAGE
Repository: [CRITICAL]Vampy Engine
Link: https://vampy.hexway.io/repositories/11234494-738c-404b-aa1b-14a8959
```

```
-----
Checking SQG for repository 'vmp/product/vampy-engine' ...
-----
```

```
  QualityGate: 'Critical'
Repository: 'Vampy Engine'
Link: https://vampy.hexway.io/repositories/11234494-738c-404b-aa1b-14a8959
Status: FAILED
Results:
```

| STATUS | SCANNER TYPE | SEVERITIES | MAX VALUE | CURRENT VALUE |
|--------|--------------|------------|-----------|---------------|
| PASSED | SAST         | CRITICAL   | 0         | 0             |
| PASSED | DAST         | CRITICAL   | 0         | 0             |
| FAILED | SCA          | CRITICAL   | 0         | 1             |
| PASSED | SAST         | HIGH       | 0         | 0             |
| PASSED | DAST         | HIGH       | 0         | 0             |
| FAILED | SCA          | HIGH       | 10        | 20            |
| PASSED | SAST         | MEDIUM     | 5         | 0             |
| PASSED | DAST         | MEDIUM     | 5         | 0             |
| FAILED | SCA          | MEDIUM     | 40        | 62            |
| PASSED | SAST         | LOW        | 10        | 0             |
| PASSED | DAST         | LOW        | 10        | 0             |
| FAILED | SCA          | LOW        | 100       | 137           |

## Работа с критериями качества (QualityGate)

Для работы с SQG используйте параметр `quality-gate`

### Проверка на соответствие Quality Gate

Вы можете проверять соответствие продукта/репозитория (`--product/--repository`) заданным параметрам QG

В результате выполнения команды вы не только получите информацию о неуспешности/успешности прохождения, но так же и статус код, который может быть использован для принятия решения о необходимости останавливать пайплайн

```
vampy-cli quality-gate --repository vmp/product/vampy-engine --details
```

```
-----
Checking SQG for repository 'vmp/product/vampy-engine' ...
-----
```

```
QualityGate: 'Critical'
Repository: 'Vampy Engine'
Link: https://vmp-prod.corp.hexway.io/repositories/1efd4494-738c-404b-aalb
Status: FAILED
Results:
```

| STATUS | SCANNER TYPE | SEVERITIES | MAX VALUE | CURRENT VALUE |
|--------|--------------|------------|-----------|---------------|
| PASSED | SAST         | CRITICAL   | 0         | 0             |
| PASSED | DAST         | CRITICAL   | 0         | 0             |
| FAILED | SCA          | CRITICAL   | 0         | 1             |
| PASSED | SAST         | HIGH       | 0         | 0             |
| PASSED | DAST         | HIGH       | 0         | 0             |
| FAILED | SCA          | HIGH       | 10        | 20            |
| PASSED | SAST         | MEDIUM     | 5         | 0             |
| PASSED | DAST         | MEDIUM     | 5         | 0             |
| FAILED | SCA          | MEDIUM     | 40        | 62            |
| PASSED | SAST         | LOW        | 10        | 0             |
| PASSED | DAST         | LOW        | 10        | 0             |
| FAILED | SCA          | LOW        | 100       | 137           |

## Получение списка репозиториев

Для того чтобы получить список доступных репозиториев, выполните:

```
vampy-cli repositories
```

```
-----
Getting list or repositoryes...
-----
```

```
Repositories (20 of 1000):
```

| NAME                        | SLUG                                 |
|-----------------------------|--------------------------------------|
| Repository Tools            | commons/tools/repo-tools             |
| wh-5                        | vmp/integration-test-repos/wh-5      |
| wh-test-4                   | vmp/integration-test-repos/wh-test-4 |
| wh-test-3                   | vmp/integration-test-repos/wh-test-3 |
| wh-test-2                   | vmp/integration-test-repos/wh-test-2 |
| pts/product/depotship-proxy | pts/product/depotship-proxy          |
| Development                 | commons/docs/development             |

|                   |                                      |
|-------------------|--------------------------------------|
| TestNotifications | test-notifications                   |
| Base Engine 3.13  | commons/base-images/base-engine-3-13 |
| Base Engine 3.12  | commons/base-images/base-engine-3-12 |
| Base Engine 3.11  | commons/base-images/base-engine-3-11 |
| Vampy CLI         | vmp/product/vampy-cli                |
| artifacts-manip   | commons/tools/artifacts-manip        |
| vm45              | vmp/integration-test-repos/vm45      |
| vm4               | vmp/integration-test-repos/vm4       |

## Поиск репозитория по его названию

```
vampy-cli repositories --search vampy
```

```
-----
```

```
Getting list or repositoryes...
```

```
-----
```

```
Repositories (10 of 10):
```

| NAME                    | SLUG                                | CRITICALITY |
|-------------------------|-------------------------------------|-------------|
| Vampy CLI               | vmp/product/vampy-cli               | HIGH        |
| Vampy k8s               | vmp/deploy/vampy-k8s                |             |
| Vampy Scan Images       | vmp/product/vampy-scan-images       | HIGH        |
| Vampy Docker Proxy      | vmp/product/vampy-docker-proxy      | CRITICAL    |
| Vampy manuals           | vmp/docs/vmp-manuals                | MEDIUM      |
| Vampy Install Generator | vmp/product/vampy-install-generator | MEDIUM      |
| Vampy Orchestration     | vmp/product/vampy-orchestration     | CRITICAL    |
| Vampy Files             | vmp/product/vampy-files             | HIGH        |
| Vampy Engine            | vmp/product/vampy-engine            | CRITICAL    |
| Vampy Deck              | vmp/product/vampy-deck              | CRITICAL    |

## Получение списка продуктов

Для того чтобы получить список доступных продуктов, выполните:

```
vampy-cli products
```

```
-----
```

```
Getting list or products...
```

```
-----
```

```
Products (6 of 6):
```

| NAME            | SLUG          | CRITICALITY | LINK                                                            |
|-----------------|---------------|-------------|-----------------------------------------------------------------|
| Base Images     | base-images   | CRITICAL    | <a href="https://vampy.hexway.io/">https://vampy.hexway.io/</a> |
| Default product | default       | LOW         | <a href="https://vampy.hexway.io/">https://vampy.hexway.io/</a> |
| Side products   | side-products | MEDIUM      | <a href="https://vampy.hexway.io/">https://vampy.hexway.io/</a> |
| Vampy           | vampy         | CRITICAL    | <a href="https://vampy.hexway.io/">https://vampy.hexway.io/</a> |
| Apiary          | apiary        | CRITICAL    | <a href="https://vampy.hexway.io/">https://vampy.hexway.io/</a> |
| Hive            | hive          | CRITICAL    | <a href="https://vampy.hexway.io/">https://vampy.hexway.io/</a> |

## Поиск продукта по его названию

```
vampy-cli products --search Vampy
```

```
-----
```

```
Getting list of products...
```

```
-----
```

```
Products (1 of 1):
```

| NAME  | SLUG  | CRITICALITY | LINK                                                                                                |
|-------|-------|-------------|-----------------------------------------------------------------------------------------------------|
| Vampy | vampy | CRITICAL    | <a href="https://vampy.hexway.io/products/455416e5-">https://vampy.hexway.io/products/455416e5-</a> |

## Получение списка фоновых задач

### Проверка статусов фоновых задач

Большинство задач в Vampy выполняются в фоне. При запуске задачи ей присваивается id

Для того чтобы проверить текущий статус фоновой задачи, необходимо использовать команду `bg-task` и указать идентификатор задачи

Например

```
vampy-cli bg-task --task-id 353f5c26-4f06-4ae1-8999-321e2932b49c --details
```

```
-----
```

```
Checking Background Task 353f5c26-4f06-4ae1-8999-321e2932b49c
```

```
-----
```

```
Background Task: 'Parsing SEMGREP scan results'
```

```
Link: https://vampy.hexway.io/admin/bgTasks?id=353f5c26-4f06-4ae1-8999-321e2932b49c
```

```
Status: SUCCESS
```

```
SubTasks:
```

| NAME                                    | STARTED                     |
|-----------------------------------------|-----------------------------|
| Parsing process                         | 2024-12-18T05:23:32.193128Z |
| Linking related items                   | 2024-12-18T05:23:32.256742Z |
| Process finding's duplications          | 2024-12-18T05:23:37.342201Z |
| AutoResolving or re-opening scan issues | 2024-12-18T05:23:39.505292Z |
| Fetch git blames for ScanIssues         | 2024-12-18T05:23:39.720654Z |
| Scan upload events sending              | 2024-12-18T05:23:39.806415Z |

## Настройка параметров вывода

### Настройка параметров вывода

В `vampy_cli` поддерживается 3 режима вывода:

## Стандартный

Используется по умолчанию. Минимальная информация. Возвращается только статус код работы и минимальный набор ответов.

Пример. Загрузка результатов сканирования TRIVY на платформу:

```
vampy-cli upload --repository vmp/product/vampy-engine --file ~/scans_trivy
ScanUpload: e75c4c39-8c15-4166-8711-f5fefc398816
QG: FAILED
```

## Детализированный

Расширенный формат вывода результатов работы утилиты. Для его использования необходимо передать параметр `--details`

Пример. Загрузка результатов сканирования TRIVY на платформу:

```
vampy-cli upload --repository vmp/product/vampy-engine --file ~/scans_trivy
-----
Run scan result uploading for repository vmp/product/vampy-engine | scanner
-----
  Scan successful uploaded
Parser: TRIVY_IMAGE
Repository: [CRITICAL]Vampy Engine
Link: https://vampy.hexway.io/repositories/123d4494-738c-404b-aa1b-14a8959
-----
Checking SQG for repository 'vmp/product/vampy-engine' ...
-----

  QualityGate: 'Critical'
Repository: 'Vampy Engine'
Link: https://vampy.hexway.io/repositories/123d4494-738c-404b-aa1b-14a8959
Status: FAILED
Results:
```

| STATUS | SCANNER TYPE | SEVERITIES | MAX VALUE | CURRENT VALUE |
|--------|--------------|------------|-----------|---------------|
| PASSED | SAST         | CRITICAL   | 0         | 0             |
| PASSED | DAST         | CRITICAL   | 0         | 0             |
| FAILED | SCA          | CRITICAL   | 0         | 1             |
| PASSED | SAST         | HIGH       | 0         | 0             |
| PASSED | DAST         | HIGH       | 0         | 0             |
| FAILED | SCA          | HIGH       | 10        | 20            |
| PASSED | SAST         | MEDIUM     | 5         | 0             |
| PASSED | DAST         | MEDIUM     | 5         | 0             |
| FAILED | SCA          | MEDIUM     | 40        | 62            |
| PASSED | SAST         | LOW        | 10        | 0             |
| PASSED | DAST         | LOW        | 10        | 0             |
| FAILED | SCA          | LOW        | 100       | 137           |



|        |      |          |     |     |
|--------|------|----------|-----|-----|
| PASSED | SAST | CRITICAL | 0   | 0   |
| PASSED | DAST | CRITICAL | 0   | 0   |
| FAILED | SCA  | CRITICAL | 0   | 1   |
| PASSED | SAST | HIGH     | 0   | 0   |
| PASSED | DAST | HIGH     | 0   | 0   |
| FAILED | SCA  | HIGH     | 10  | 20  |
| PASSED | SAST | MEDIUM   | 5   | 0   |
| PASSED | DAST | MEDIUM   | 5   | 0   |
| FAILED | SCA  | MEDIUM   | 40  | 62  |
| PASSED | SAST | LOW      | 10  | 0   |
| PASSED | DAST | LOW      | 10  | 0   |
| FAILED | SCA  | LOW      | 100 | 137 |

# Оркестрация

## Предустановленные сканеры

---

Платформа поставляется с несколькими сканерами, которые могут быть запущены как в рамках одного [репозитория](#) или [продукта](#).

| Name                    | Type |
|-------------------------|------|
| <a href="#">semgrep</a> | SAST |
| <a href="#">trivy</a>   | SCA  |

### **semgrep**

---

By default, the scanner starts with the following parameters:

```
semgrep scan --json --output "${RESULTS_FILE}" "${CURRENT_DIR}
```

### **Trivy**

---

By default, the scanner starts with the following parameters:

```
trivy fs
--scanners vuln,secret,misconfig ${SRC_DIR}
-f json --output ${RESULTS_FILE}
--cache-dir ${CACHE_DIR}
--db-repository="${HW_REGISTRY}/trivy/trivy-db:2
```

## Запуск сканирования

---

:warning: This section is temporary. In the next releases, orchestration setup and configuration will be implemented through the platform web interface.

Scanning can be run for individual repositories as well as for products.

## **Starting a repository scan**

To start a repository scan, you need to:

- Open the "right panel" with information about the repository
- Start the scan by clicking the "Start" button
- The scan will be run in the background **for the selected repository**. The scan results will be automatically uploaded to the platform.

## **Starting a product scan**

To start a product scan, you need to:

- Open the "right panel" with information about the product
- Start the scan by clicking the "Start" button
- The scan will be run in the background **for all repositories of the selected product**. The scan results will be automatically uploaded to the platform.

## **Изменение параметров сканирований**

:warning: This section is temporary. In the next releases, orchestration setup and configuration will be implemented through the platform web interface.

### **Show scanner run options**

To view the run parameters of scanners, you need to hover your mouse over their names in the "Scans" section

### **Change scanner run options**

If you want to change the parameters or the order in which the scanners are launched, you must:

#### **For Docker installation**

- Change the settings in the user `.ini` file. It is located at `/opt/hw-vmp/config/`

#### **Change the list of scanners to run**

Set the `v.orchestration.scanners` parameter to a list of scanners that should be launched:

Example:

```
v.orchestration.scanners = SEMGREP,TRIVY_FS
```

### Change the order and sequence of launching scanners

Add line:

- `v.orchestration.run_mode = CONCURRENT` in case the scanners must work in parallel with each other.
- `v.orchestration.run_mode = CHAIN` in case the scanners must work sequentially (one after the other).

### Change run options for semgrep scanner

Set the necessary scanner launch options in the `v.orchestration.command.semgrep` parameter

### Change run options for trivy scanner

Set the necessary scanner launch options in the `v.orchestration.command.trivy_fs` parameter

### Environment variables

`${RESULTS_FILE}` - path to the file where we expect the scan result to be located

`${SRC_DIR}` - path to the sources

`${CACHE_DIR}` - directory with caches

`${HW_REGISTRY}` - our registry (in particular, a public one is used to download the database from trivi)

### For Kubernetes intallation

To change the parameters for running scanners, use the corresponding parameters in the `values.yaml` file.

## Загрузка результатов сканирований

Вы можете загружать результаты сканирования с помощью API.

Чтобы загрузить данные на платформу:

- [Создайте бота](#). Для корректной работы бота ему необходимо присвоить роль **Editor** или **Admin**. Запомните сгенерированный API токен.

- Выполните следующий запрос CURL:

```
bash curl -X POST "$SEC_VV_URL/scan_uploads/" \ -H "accept: */*" \
\ -H "Content-Type: multipart/form-data" \ -H "Authentication:
$SEC_VV_KEY" \ -F "repository=$VAMPY_REPO_SLUG" \ -F
"repositoryBranch=$VAMPY_REPO_BRANCH" \ -F
"scannerResultFile=@$VV_REPORT_FILE_NAME" \ -F
"scannerType=$VV_SCAN_TYPE" #(1)!
```

Примечания:

- (1) Имя сканера. Поддерживаемые сканеры:
  - TRIVY\_FS
  - RIVY\_IMAGE
  - DEPENDENCY\_CHECK
  - ESLINT
  - SEMGREP
  - GRYPE
  - TRUFFLEHOG

где

- **SEC\_VV\_UR** - путь к серверу Vampr, напр. `http://vmp-demo.corp.hexway.io/api/`;
- **SEC\_VV\_KEY** - токен API для бота;
- **VAMPY\_REPO\_SLUG** - путь к проекту репозитория, для которого работает пайплайн. Эквивалентно `CI_PROJECT_PATH`;
- **VAMPY\_REPO\_BRANCH** - ветка, для которой работает пайплайн. Эквивалентно `CI_COMMIT_REF_NAME`;
- **VV\_REPORT\_FILE\_NAME** - имя файла с результатами скана;
- **VV\_SCAN\_TYPE** - название сканера. Поддерживаемые сканеры можно посмотреть ([на этой странице](#)).