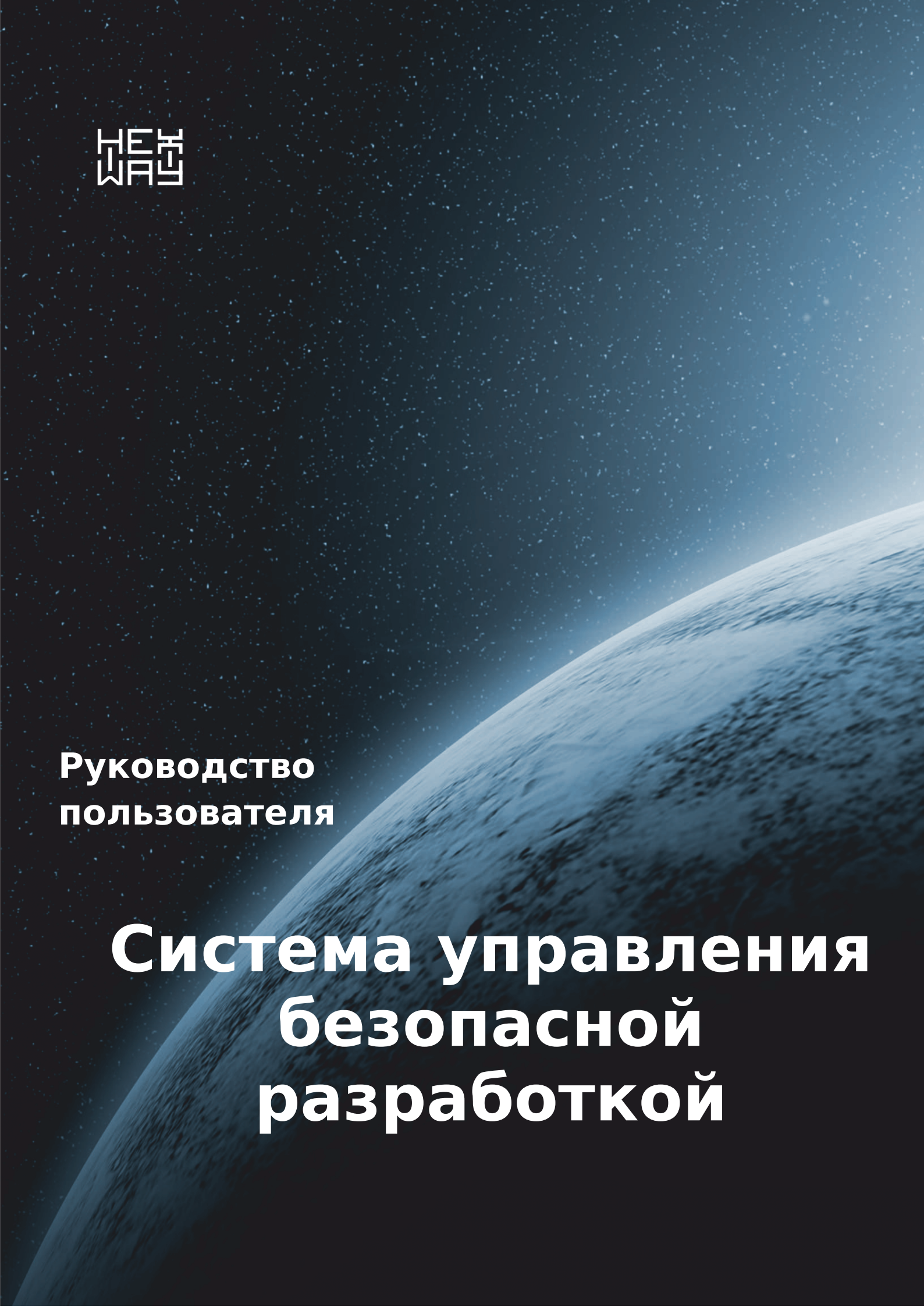

Руководство пользователя

Система управления безопасной разработкой

О платформе

Краткий обзор системы управления безопасной разработкой

Система управления безопасной разработкой — 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 Систему управления безопасной разработкой! Ознакомьтесь со следующими ссылками, чтобы начать свою работу:

1. Создание репозитория.
2. Загрузка результатов сканирований.

Если вы не нашли ответы на свои вопросы, можете задать их сюда: support@hexway.ru.

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 репозиториев, 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 репозиториев, 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 репозиториев, отсортировать их по критичности или добавить новые репозитории в продукт.

При нажатии на один из репозиториев 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 репозиториев. 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 Системы управления безопасной разработкой являются контейнерами для issues, созданных в результате процесса сканирования, завершенного 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 Системе управления безопасной разработкой, просто выберите нужную интеграцию из списка

затем выберите нужный репозиторий из списка:

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

Статус	Описание
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, завершенного в рамках пайплайна.

Фильтры компонентов

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

Список заполняется автоматически в результате процесса сканирования, завершенного 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.

Функции

Список поддерживаем сканеров

Здесь инструкция по загрузке результатов сканирования.

Secret Scan

Название сканера	VV_SCAN_TYPE	Поддерживаемые языки	Ссылка
Trufflehog3	TRUFFLEHOG3	All	Github
Gitlab Secret detection	GITLAB_SECRET_DETECT	All	Gitlab
Trufflehog	TRUFFLEHOG	All	Github

SAST

Название сканера	VV_SCAN_TYPE	Поддерживаемые языки	Ссыл
Semgrep	SEMGREP		Github

Название сканера	VV_SCAN_TYPE	Поддерживаемые языки	Ссыл
		C#, Go, Java, JavaScript, JSX, JSON, PHP, Python, Ruby, Scala, Terraform, TypeScript, TSX	
ESLint	ESLINT	ECMAScript, JavaScript	Github
Checkmarx	CHECKMARX	C#, Go, Java, JavaScript, JSX, JSON, PHP, Python, Ruby, Scala, Terraform, TypeScript, TSX	Check
GitLab SAST	GITLAB_SAST	NET Core, .NET Framework, Go, Java, Python, React, JavaScript, TypeScript	Gitlab
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	Snyk.i
SARIF	SARIF	все	SarifW
Sonarqube	sonarqube	все	sonars
PT AI	PT_AI	все	PT AI
AppScreener	SARIF	все	rt-sola solar.r solar_a

Software Composition Analysis (SCA)

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

Название сканера	VV_SCAN_TYPE	Тип сканируемых объе
PT AI	PT_AI	software dependencies
CodeScoring	SARIF	OS packages, software depen

DAST

Название сканера	VV_SCAN_TYPE	Тип сканируемых объектов	Ссылка
Nuclei	NUCLEI	Network, Web	Github
ZAP	ZAP	Network, Web	ZAP
Gitlab DAST	GITLAB_DAST	Network, Web	Gitlab DAST
Acunetix	ACUNETIX	Network, Web	Acunetix
Burp	BURP	WEB	Burp
PT AI	PT_AI	WEB	PT AI

IaC

Название сканера	VV_SCAN_TYPE	Тип сканируемых объектов	Link
kics	KICS	IaC	Github

Алгоритм дедупликации

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

Issue — это сущность в Системе управления безопасной разработкой, которая содержит всю информацию об 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 Системе управления безопасной разработкой 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** Система управления безопасной разработкой 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	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 Системе управления безопасной разработкой 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)	ко (кр
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 Системы управления безопасной разработкой, 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.eyJ1c2VybmFtZSI6ImJvdF91c2VyIiwiaWF0I
\ -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": ""
  }
}
```


Консольная утилита vampy-cli

Об утилите vampi-cli

Описание

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

`vampy-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

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

Доступные команды

- **`upload`**
  Загружает результаты сканирования, которые уже существуют.
- **`scan`**
  Запускает процесс сканирования указаным 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 Системы управления безопасной разработкой (или значение из переменной окружения).	`значение из переменной $VAMPY_API_TOKEN`
`--api-version value`	Версия API Системы управления безопасной разработкой.	`v1`
`--timeout value`	Таймаут в секундах.	`120`
`--vampy-url value`	URL-адрес сервера Системы управления безопасной разработкой (или значение из переменной окружения).	`значение из переменной $VAMPY_URL`

Конфигурация

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

Параметр	Описание	Значение по умолчанию
`--api-token value`	API-токен Системы управления безопасной разработкой (или из переменной `VAMPY_API_TOKEN`).	`API Token from env 'VAMPY_API_TOKEN'`
`--vampy-url value`	URL сервера Системы управления безопасной разработкой (или из переменной `VAMPY_URL`).	`URL from env 'VAMPY_URL'`

Вы можете определить их несколькими способами:

Переменные окружения

```
export VAMPY_API_TOKEN=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZS
export VAMPY_URL=https://vampy.hexway.io

# получим список репозиториев
vampy-cli repositories
```

Параметры командной строки

```
# получим список репозиториев
vampy-cli --vampy-url https://vampy.hexway.io  --api-token eyJhbGciOiJIUzI
```

Установка vampy-cli

Вы можете установить `vampy-cli` с помощью скрипта утилиты `curl` или загрузив предварительно скомпилированный файл со страницы релиза GitHub.

Установка с помощью `curl`

```
$ curl -sSL https://get.vampy.ru | sh
```

Скачать готовые бинарные файлы

На странице Releases на GitHub вы найдете скомпилированные файлы `vampy-cli` для различных платформ

Использовать Docker образ

```
export VAMPY_CLI_VERSION=0.1.0
docker run \
  registry.hexway.io/hexway/vampy-cli:${VAMPY_CLI_VERSION} help
```

Конфигурация

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

Параметр	Описание	Значение по умолчанию
`--api-token value`	API-токен Системы управления безопасной разработкой (или из переменной `VAMPY_API_TOKEN`).	`API Token from env 'VAMPY_API_TOKEN'`
`--vampy-url value`	URL сервера Системы управления безопасной разработкой (или из переменной `VAMPY_URL`).	`URL from env 'VAMPY_URL'`

Вы можете определить их несколькими способами:

Переменные окружения

```
export VAMPY_API_TOKEN=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZS
export VAMPY_URL=https://vampy.hexway.io

# получим список репозиториев
./vampy_cli repositories
```

Параметры командной строки

```
# получим список репозиториев
./vampy_cli --vampy-url https://vampy.hexway.io  --api-token eyJhbGciOiJIU
```

Запуск сканирований

С помощью команды `scan` вы можете заускать на сканирование репозитории (`--repository`) и продукты (`--products`)

Система управления безопасной разработкой автоматически запустит сканер или группу сканеров, а после завершения сканирования загрузит результаты в Систему управления безопасной разработкой

Запуск сканировани дефольным профилем сканирования

```
vampy-cli scan --repository vmp/product/vampy-engine --details

--------
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/123d4494-738c-404b-aa1b-14a8959
```

```
ScanProfile "Default" | RunMode: "CONCURRENT"
Commands:
---------
┌─ Semgrep
│   ├─ Scanner: SEMGREP
│   ├─ DockerImage: regestry.hexway.io/hexway/vampy-scan-images/semgrep:lat
│   └─ Command: semgrep scan --json --output "${RESULTS_FILE}" --config=p/o
└─ Trivy: FileSystem
    ├─ Scanner: TRIVY_FS
    ├─ DockerImage: regestry.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: regestry.hexway.io/hexway/vampy-scan-images/semgrep:lat
│   └─ Command: semgrep scan --json --output "${RESULTS_FILE}" --config=p/o
└─ Trivy: FileSystem
    ├─ Scanner: TRIVY_FS
    ├─ DockerImage: regestry.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 reposito
--------
```

```
  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 соответсвие выбранным критериям качества (Security Quality Gates)

Для этого запустите `vampy_cli` с параметрами `--check-qg` 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: regestry.hexway.io/hexway/vampy-scan-images/semgrep:lat
│   └─ Command: semgrep scan --json --output "${RESULTS_FILE}" --config=p/o
└─ Trivy: FileSystem
    ├─ Scanner: TRIVY_FS
    ├─ DockerImage: regestry.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/1efd4494-738c-404b-aa1b-14a8959
Status: FAILED
Results:
```

STATUS	SCANNER TYPE	SEVERITIES	MAX VALUE	CURRENT VALUE
PASSED	SAST	CRITICAL	0	0
PASSED	DAST	CRITICAL	0	0
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/1efd1234-738c-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/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 reposito
--------

  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: regestry.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 reposito
--------

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

NAME	STARTED	FIN
Prepare data for orchestration	2024-12-18T05:16:52.930801Z	202
Execute orchestration task	2024-12-18T05:16:53.727320Z	202
Await orchestration service callback		

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 парметрам QG

В результате выполнения команды вы не только полчите 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-aa1b
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	CRITICAL
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	https://vampy.hexway.io/
Default product	default	LOW	https://vampy.hexway.io/
Side products	side-products	MEDIUM	https://vampy.hexway.io/
Vampy	vampy	CRITICAL	https://vampy.hexway.io/
Apiary	apiary	CRITICAL	https://vampy.hexway.io/
Hive	hive	CRITICAL	https://vampy.hexway.io/

Поиск продукта по его названию

```
vampy-cli products --search Vampy
--------
Getting list or products...
--------
Products (1 of 1):
```

NAME	SLUG	CRITICALITY	LINK
Vampy	vampy	CRITICAL	https://vampy.hexway.io/products/455416e5-

Получение списка фоновых задач

Проверка статусов фоновых задач

Большинство задач в Системе управления безопасной разработкой 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=353f5c26-4f06-4ae1-8999-321
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_triv
ScanUpload: e75c4c39-8c15-4166-8711-f5fefc398816
QG: FAILED
```

Детализированый

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_triv
--------
Run scan result uploading for repository vmp/product/vampy-engine | scanne
--------
  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

Расширенный

Выводит дополнительную информацию о коммуникации `vampy_cli` с Системой управления безопасной разработкой Для его использования необходимо передать параметр `--verbose`

Пример. Загрузка результатов сканирования TRIVY на платформу:

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

```
Run scan result uploading for repository vmp/product/vampy-engine | scanne
--------
  Scan successful uploaded
Parser: TRIVY_IMAGE
Repository:  [CRITICAL]Vampy Engine
Link: https://vampy.hexway.io/repositories/123d4494-7123-404b-aa1b-14a8959

--------
Checking SQG for repository 'vmp/product/vampy-engine' ...
--------
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af
Repository: [CRITICAL] 'Vampy Engine' ... Status: PROGRESS | will retry af

  QualityGate: 'Critical'
Repository: 'Vampy Engine'
Link: https://vampy.hexway.io/repositories/1efd4494-7123-404b-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 |
```

Оркестрация

Предустановленные сканеры

Платформа поставляется с несколькими сканерами, которые могут быть запущены как в рамках одного репозитория или продукта.

Name	Type
semgrep	SAST
trivy	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 Системы управления безопасной разработкой, 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).