ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ SDLPLATFORM

Возможности и характеристики продукта

Содержание

1. Введение

1.1 Назначение платформы

SDLPlatform (Secure Development Lifecycle Platform) — отечественная ASOC-платформа, предназначенная для организации и автоматизации процесса безопасной разработки программного обеспечения. Платформа обеспечивает выполнение технических требований ГОСТ Р 56939-2024 к анализу качества кода, позволяя проводить статический анализ исходного кода (SAST), анализ зависимостей (SCA), поиск секретов, анализ конфигураций и проверку соблюдения правил кодирования.

1.2 Область применения

Платформа предназначена для корпоративной разработки и сопровождения программных продуктов в организациях, внедряющих практики DevSecOps. Использование платформы упрощает выполнение технических требований ГОСТ Р 56939-2024 и Приказа ФСТЭК РФ № 117.

Терминология и сокращения

Термин	Определение
ASOC	Application Security Orchestration and Correlation — оркестрация и корреляция безопасности приложений
SAST	Static Application Security Testing — статический анализ безопасности приложений
SCA	Software Composition Analysis — анализ состава программного обеспечения (зависимостей)
CWE	Common Weakness Enumeration — общий перечень уязвимостей
OWASP	Open Web Application Security Project — открытый проект по безопасности веб-приложений
CI/CD	Continuous Integration / Continuous Delivery — непрерывная интеграция и доставка
VCS	Version Control System — система контроля версий

2. Функциональные возможности

2.1 Управление приложениями

Приложение является основной единицей организации работы в платформе и соответствует программному продукту или проекту организации. Платформа позволяет создавать, редактировать и удалять приложения, указывая для каждого название и описание.

Для каждого приложения доступна гибкая настройка параметров сканирования. Пользователь может определить каталоги, которые следует исключить из анализа, задать минимальный уровень критичности для регистрации сработок, а также указать конкретные правила отдельных сканеров, которые не должны применяться. Поддерживается настройка исключений для Semgrep, Gosec, MobSFScan, Trivy и TruffleHog, а также шаблоны исключений для анализа зависимостей (SCA).

Приложение может быть привязано к репозиторию GitLab для автоматического получения исходного кода. Для работы в изолированных средах без доступа к внешним сетям предусмотрен offline-режим.

2.2 Управление ревизиями исходного кода

Ревизия представляет собой версию исходного кода приложения, загруженную для анализа. Платформа поддерживает несколько способов загрузки кода: загрузка ZIP-архива через веб-интерфейс, загрузка через программный интерфейс (API), а также автоматическое получение кода из GitLab по указанной ветке репозитория.

Для каждой ревизии сохраняется контрольная сумма и размер файла, что позволяет отслеживать изменения и обеспечивает воспроизводимость результатов анализа. При повторных сканированиях платформа автоматически определяет новые и ранее обнаруженные уязвимости, что позволяет отслеживать динамику изменений между ревизиями.

2.3 Процесс сканирования

Сканирование запускается через веб-интерфейс или API. Для интеграции с CI/CD-конвейерами предусмотрен эндпоинт, позволяющий загрузить код и запустить сканирование одним запросом.

Платформа автоматически определяет языки программирования и типы файлов в загруженном коде, после чего подбирает подходящие инструменты анализа. Сканеры запускаются параллельно, что сокращает общее время проверки. Система предотвращает одновременный запуск нескольких сканирований для одного приложения.

Пользователь может отслеживать прогресс сканирования в реальном времени, видеть статус каждого сканера и получать информацию об ошибках через API.

Платформа поддерживает два режима работы: online-режим с доступом к внешним базам уязвимостей (NVD) для получения актуальной информации о CVE, и offline-режим для изолированных сред, где анализ выполняется с использованием локальных баз правил.

2.4 Управление уязвимостями

Каждая обнаруженная уязвимость представлена в виде карточки, содержащей полную информацию о проблеме: описание, классификацию по CWE, источник обнаружения (название сканера), сработавшее правило, путь к файлу с указанием номеров строк, фрагмент исходного кода, уровень критичности и оценку достоверности от ML-модели.

Платформа использует двухуровневую систему статусов. Статус анализа отражает этап работы с уязвимостью: новая (New), на проверке (Check), признана ложным срабатыванием (False Positive) или исправлена (Fixed). Статус устранения показывает фактическое состояние проблемы в коде: открыта и требует устранения (Open), устранена в коде (Fixed), подтверждённое ложное срабатывание (False Positive) или риск принят (Approved).

Уровни критичности уязвимостей варьируются от критического (Critical) до информационного (Info), включая высокий (High), средний (Medium) и низкий (Low) уровни.

Пользователь может изменять статусы и критичность уязвимостей, добавлять комментарии для фиксации решений и обсуждений, а также выполнять массовые операции над группой уязвимостей. Доступны гибкие возможности фильтрации и поиска по различным параметрам.

Платформа автоматически выполняет дедупликацию результатов: если несколько сканеров обнаружили одну и ту же проблему, она будет представлена как единая уязвимость. При этом сохраняется история обнаружений с датами первого и последнего появления.

2.5 Аналитика и отчётность

Платформа предоставляет дашборд с агрегированной статистикой по всем приложениям организации: распределение уязвимостей по уровням критичности и статусам, недельная динамика изменений и статистика триажа (соотношение истинных и ложных срабатываний).

Аналитика доступна как по всем приложениям, так и по каждому в отдельности.

2.6 Интеллектуальная оценка достоверности

Платформа включает ML-компонент, который автоматически оценивает достоверность каждой обнаруженной уязвимости, помогая отличить реальные проблемы безопасности (True Positive) от ложных срабатываний (False Positive).

Для каждой уязвимости рассчитывается показатель ML Confidence в диапазоне от 0% до 100%. Значения от 0% до 30% указывают на высокую вероятность ложного срабатывания, от 30% до 70% — на необходимость проверки специалистом, а значения от 70% до 100% свидетельствуют о высокой вероятности реальной уязвимости.

Модель анализирует фрагмент исходного кода, название сработавшего правила, тип уязвимости по классификации CWE и метрики от сканера.

3. Поддерживаемые инструменты анализа

3.1 SAST (Static Application Security Testing)

Статический анализ исходного кода на наличие уязвимостей безопасности.

Сканер	Языки	Описание
Semgrep	C, C++, C#, Go, Java, JavaScript, JSX, Kotlin, PHP, Pascal, Fortran, Python, Ruby, Rust, Swift, TypeScript, ASP.NET	Универсальный паттерн-ориентированный анализатор
Bandit	Python	Специализированный анализатор безопасности Python
Bearer	Go, Java, JavaScript, PHP, Python, Ruby, TypeScript	Анализатор безопасности и приватности данных
CodeQL	C#, Java, C/C++, Python, JavaScript/TypeScript, Go, Ruby, Swift	Семантический анализатор от GitHub
Gosec	Go	Специализированный анализатор для Go
MobSFScan	Java, Kotlin, Objective-C, Swift	Анализатор для мобильной разработки
PMD	Java, JavaScript	Анализатор качества и безопасности кода
ABAP Code Scanner	ABAP	Анализатор для SAP ABAP

Опционально поддерживается интеграция с коммерческими сканерами PT AI (Positive Technologies Application Inspector) и Solar AppScreener.

3.2 SCA (Software Composition Analysis)

Анализ зависимостей на наличие известных уязвимостей.

Сканер	Назначение
Trivy	Анализ зависимостей, контейнеров, IaC-конфигураций
OWASP Dependency-Check	Анализ JAR-файлов и зависимостей Java-проектов

3.3 Secret Scanning

Поиск секретов, токенов и ключей в исходном коде.

Сканер	Назначение
TruffleHog	Поиск секретов с высокой энтропией и по паттернам
Trivy	Поиск секретов в коде и конфигурациях

3.4 Config Scanning

Анализ конфигурационных файлов на уязвимости и некорректные настройки.

Сканер	Назначение
Trivy	Анализ Dockerfile, Kubernetes YAML, Terraform и других конфигураций

3.5 Linters

Проверка качества кода и соблюдения правил кодирования.

Линтер	Язык
Ruff	Python
golangci-lint	Go
PMD	Java, JavaScript
PHPMD	PHP

4. Поддерживаемые языки программирования

Платформа поддерживает 19 языков программирования с различным набором типов анализа.

Язык	SAST	SCA	Linters	Secret	Config
ABAP	ABAP Code Scanner	—	—	TruffleHog, Trivy	Trivy
ASP.NET	Semgrep	—	—	TruffleHog, Trivy	Trivy
C	Semgrep, CodeQL	—	—	TruffleHog, Trivy	Trivy
C#	Semgrep, CodeQL	—	—	TruffleHog, Trivy	Trivy
C++	Semgrep, CodeQL	—	—	TruffleHog, Trivy	Trivy
Fortran	Semgrep	—	—	TruffleHog, Trivy	Trivy
Go	Semgrep, Gosec, Bearer, CodeQL	Trivy	golangci-lint	TruffleHog, Trivy	Trivy
Java	Semgrep, MobSFScan, Bearer, PMD, CodeQL	OWASP DC, Trivy	PMD	TruffleHog, Trivy	Trivy
JavaScript	Semgrep, Bearer, PMD, CodeQL	Trivy	PMD	TruffleHog, Trivy	Trivy
JSX	Semgrep	Trivy	—	TruffleHog, Trivy	Trivy
Kotlin	Semgrep, MobSFScan	Trivy	—	TruffleHog, Trivy	Trivy
Objective-C	MobSFScan	—	—	TruffleHog, Trivy	Trivy
Pascal	Semgrep	—	—	TruffleHog, Trivy	Trivy
PHP	Semgrep, Bearer	Trivy	PHPMD	TruffleHog, Trivy	Trivy
Python	Semgrep, Bandit, Bearer, CodeQL	Trivy	Ruff	TruffleHog, Trivy	Trivy
Ruby	Semgrep, Bearer, CodeQL	Trivy	—	TruffleHog, Trivy	Trivy
Rust	Semgrep	Trivy	—	TruffleHog, Trivy	Trivy
Swift	Semgrep, MobSFScan, CodeQL	—	—	TruffleHog, Trivy	Trivy
TypeScript	Semgrep, Bearer, CodeQL	Trivy	—	TruffleHog, Trivy	Trivy

Сканеры Trivy и TruffleHog являются универсальными и работают для всех языков программирования.

5. Веб-интерфейс

5.1 Основные разделы

Веб-интерфейс платформы включает несколько основных разделов. Раздел «Аналитика» представляет собой дашборд с общей статистикой по всем приложениям организации. В разделе «Приложения» отображается список приложений с возможностью создания новых и управления существующими. Раздел «Уязвимости» содержит глобальный список уязвимостей со всех приложений с возможностью фильтрации и поиска. «Рабочая область» предназначена для работы с уязвимостями конкретного выбранного приложения. Раздел «Инструменты» позволяет управлять интеграциями с GitLab и внешними сканерами. В разделе «Настройки приложения» выполняется конфигурация параметров сканирования.

5.2 Работа с приложениями

Список приложений отображается в табличном виде с индикаторами статуса сканирования. Для каждого приложения показывается количество уязвимостей по уровням критичности, индикатор прогресса текущего сканирования и быстрый переход к списку уязвимостей.

5.3 Работа с уязвимостями

Уязвимости представлены в виде таблицы с возможностью сортировки по любому столбцу. Доступна многокритериальная фильтрация по критичности, статусу анализа, статусу устранения, типу (SAST или SCA) и принадлежности к приложению. Поддерживается полнотекстовый поиск по описанию и другим полям.

При выборе уязвимости открывается детальная карточка с полной информацией о проблеме. Пользователь может изменить статус и критичность, добавить комментарий или выполнить массовую операцию над несколькими уязвимостями одновременно.

5.4 Настройки сканирования

В настройках приложения можно выбрать источник кода (загрузка ZIP-архива или получение из GitLab), настроить параметры SAST-анализа (минимальная критичность, исключаемые каталоги и правила) и SCA-анализа (шаблоны исключений), а также включить режим offline-сканирования для работы в изолированных средах.

5.5 Дополнительные возможности

Интерфейс поддерживает переключение между светлой и тёмной темой оформления, локализацию на русский и английский языки, а также генерацию API-токенов для интеграции с внешними системами.

6. Интеграционные возможности

6.1 Интеграция с CI/CD

Платформа предоставляет HTTP API для интеграции с системами непрерывной интеграции и доставки. Результаты сканирования доступны через API, что позволяет CI/CD-конвейеру принимать решение о продолжении или блокировке релиза на основе количества и критичности обнаруженных уязвимостей.

Имеются готовые интеграции для GitLab CI и Jenkins. Благодаря универсальному API платформа может быть интегрирована с любой другой системой CI/CD.

6.2 Интеграция с системами контроля версий

Платформа поддерживает интеграцию с GitLab. После подключения по URL и токену доступа пользователь может просматривать список репозиториев и веток, привязывать приложения к конкретным репозиториям и автоматически получать код для сканирования.

6.3 Интеграция с коммерческими сканерами

Помимо встроенных инструментов анализа, платформа поддерживает интеграцию с коммерческими SAST-решениями: PT AI (Positive Technologies Application Inspector) и Solar AppScreener. Код отправляется на анализ во внешний сканер, а результаты автоматически импортируются и нормализуются наравне с результатами встроенных инструментов.

6.4 HTTP API

Платформа предоставляет полнофункциональный REST API для автоматизации всех операций: управления приложениями и ревизиями, запуска сканирований, получения и обработки уязвимостей, работы с комментариями и интеграциями. API документирован в формате OpenAPI/Swagger.

Для аутентификации используется OAuth 2.0 с выдачей JWT-токена. Для интеграций с внешними системами можно сгенерировать долгоживущий API-токен.

7. Безопасность и доступ

7.1 Аутентификация и авторизация

Доступ к платформе защищён аутентификацией по протоколу OAuth 2.0 с выдачей JWT-токена. Поддерживается разграничение доступа по владельцу приложения: пользователь видит только свои приложения и связанные с ними уязвимости. Для администрирования платформы предусмотрена роль суперпользователя.

7.2 Защита данных

Пароли пользователей хранятся в виде криптографических хешей. Передача данных между клиентом и сервером осуществляется по защищённому протоколу HTTPS. Загруженный исходный код хранится в изолированном объектном хранилище.

7.3 Лицензирование

Платформа использует лицензирование на основе JWT-токена. Лицензия определяет срок действия, максимальное количество приложений и доступные функции. Импорт лицензии выполняется через веб-интерфейс или API.

8. Варианты развёртывания

8.1 Docker Compose

Основной способ развёртывания, подходящий для большинства сценариев использования. Все компоненты платформы поставляются в виде Docker-контейнеров и запускаются с помощью единого файла конфигурации. Данные сохраняются в персистентных томах.

Минимальные требования: Docker Engine 20.10 или выше, Docker Compose v2 или выше, 8 ГБ оперативной памяти, 50 ГБ дискового пространства.

8.2 Kubernetes / Helm

Масштабируемый вариант развёртывания для высоконагруженных сред и крупных организаций. Установка автоматизирована с помощью Helm Chart. Поддерживается горизонтальное масштабирование компонентов сканирования, интеграция с Kubernetes Secrets для управления секретами и настройка Ingress для внешнего доступа.

8.3 Изолированные среды (Air-gap)

Платформа поддерживает развёртывание в полностью изолированных средах без доступа к интернету. Все образы поставляются из приватного Docker Registry организации. Режим offline-сканирования позволяет выполнять анализ с использованием локальных баз правил и уязвимостей без обращения к внешним источникам.

8.4 SaaS

Платформа доступна в режиме SaaS (Software as a Service). В этом случае инфраструктура размещается и обслуживается поставщиком, а заказчик получает доступ к платформе через веб-интерфейс и API без необходимости развёртывания собственных серверов. Режим SaaS подходит для организаций, которые хотят начать использовать платформу без затрат на инфраструктуру и администрирование.

9. Технические характеристики

9.1 Требования к системе

Параметр	Значение
Операционная система	Linux (Ubuntu 24.04.1 LTS или выше)
Процессор	4 ядра
Оперативная память	8 ГБ
Свободное место на диске	50 ГБ
Сетевое подключение	Требуется для установки и получения обновлений

9.2 Масштабирование

Платформа поддерживает горизонтальное масштабирование путём увеличения количества экземпляров компонентов сканирования, а также вертикальное масштабирование путём выделения дополнительных ресурсов отдельным компонентам.