Платформа безопасной разработки «Шерлок»

Функциональные характеристики

Москва

2025

Содержание

1. Аннотация

Документ содержит описание функциональных характеристик ПО «Платформа безопасной разработки "Шерлок"» (далее — Система).

2. Термины и определения

Термин, сокращение	Описание
API	Application Programming Interface. Набор правил, позволяющих одному программному продукту взаимодействовать с другим
CI/CD	Continuous Integration / Continuous Delivery. Технология автоматизации тестирования и доставки новых модулей разрабатываемого проекта заинтересованным сторонам
CI-скрипт	Сценарий, который используется в CI-системе
CI-шаблон	Шаблон с данными, необходимый для настройки конвейера сборки ПО в CI-системе
Image Scan	Сканирование образов контейнеров для идентификации уязвимостей, вредоносного ПО и чувствительной информации (например, секреты)
JSON	JavaScript Object Notation. Текстовый формат обмена данными, который используется для хранения данных и их передачи между различными системами и приложениями
LDAP	Lightweight Directory Access Protocol. Протокол быстрого доступа к каталогам
SARIF	Static Analysis Results Interchange Format. Формат обмена результатами статического анализа на основе JSON для вывода инструментов статического анализа
SAST	Static Application Security Testing, статический анализатор кода Класс решений, которые позволяют анализировать программный код, байт-код и бинарные файлы для идентификации конструкций, которые могут являться ИБ-дефектами. Могут применяться на этапах разработки и тестирования жизненного цикла программного обеспечения. Например, Solar appScreener
SBOM	Software Bill of Materials. Список всех модулей и библиотек, необходимых для сборки программного продукта, а также указателей их связи друг с другом
SCA	Software Composition Analysis, анализ состава ПО Класс решений, которые позволяют идентифицировать ИБ-дефекты (уязвимости, лицензионные ограничения) в зависимостях (компонентах с открытым исходным кодом), используемых при разработке ПО. Могут применяться на этапах разработки и тестирования жизненного цикла программного обеспечения
SCM	Source Code Management, система управления исходными кодами Класс решений, которые позволяют управлять версиями программного кода, отслеживать изменения в программном коде и организовать централизованную работу разработчиков с одним или несколькими проектами (репозиториями)
Security Gate	Этап жизненного цикла программного обеспечения (ПО), на котором проверяется, что все заранее определенные требования по информационной безопасности, предъявляемые к нему, были выполнены. Необходимо для оценки возможности перехода к последующим этапам жизненного цикла ПО
ИБ	Информационная безопасность
ИБ-дефект	Дефект информационной безопасности (ИБ). Ошибка кода или инфраструктуры, которая требует внимания и устранения
Сканер	Средство анализа поддерживаемой Системы (см. SAST, SCA, Image Scan)

3. Цели и назначение

Система предназначена для оптимизации процессов безопасной разработки за счет консолидации информации об ИБ-дефектах, получаемой из различных сканеров, и предоставления возможности централизованной одновременной работы с ними различными пользователями.

4. Основные функции ПБР «Шерлок»

Основными функциями Системы являются:

- Общее
  - Система обладает задокументированным API.
- Установка
  - Система позволяет осуществлять установку локально, во внутренней инфраструктуре без наличия доступа в сеть Интернет.
- Управление учетными записями пользователей
  - Система предоставляет возможность по управлению локальными учетными записями;
  - Система предоставляет возможность по управлению доменными учетными записями за счет интеграции с корпоративным каталогом. В качестве корпоративного каталога реализована поддержка Active Directory.
- Аутентификация
  - Система поддерживает возможность локальной аутентификации пользователей;
  - Система обеспечивает проверку соответствия пароля пользователя требованиям:
    - не менее 8 (восьми) символов;
    - как минимум 1 (одна) строчная и 1 (одна) заглавная буквы;
    - как минимум 1 (одна) цифра;
    - как минимум 1 (один) специальный символ из множества (~ ! ? @ # $ % ^ & * _ - + ( ) [ ] { } > < / \ | " ' . , :).
  - Система поддерживает возможность доменной аутентификации по протоколам LDAP, LDAPs.
- Управление сессиями пользователей
  - Система блокирует сеанс работы пользователя, в случае если он не был активен в течение 15 минут.
- Ролевая модель
  - Управление доступом в Системе реализовано с использованием ролевой модели;

- Система позволяет управлять ролями на уровне:
  - всей Системы в целом;
  - отдельно взятой Группы.

- Интеграции (сканеры)
  - Система предоставляет возможность интеграции со:
    - сканерами, автоматизирующими практики статического анализа (Static Application Security Testing, SAST):
      - Solar appScreener;
      - Positive Technologies Application Inspector.
    - сканерами, автоматизирующими практики композиционного анализа (Software Composition Analysis, SCA):
      - CodeScoring.
    - сканерами, автоматизирующими практики анализа образов контейнеров на предмет наличия в них уязвимостей (Image Scan):
      - Trivy;
      - Kaspersky Container Security.
  - Система должна предоставлять возможность единовременной интеграции с несколькими экземплярами сканеров (в случае, если это возможно).

- Интеграции (окружения разработки)
  - Система предоставляет возможность интеграции с:
    - системами управления исходным кодом (Source Code Management, SCM):
      - GitLab;
    - системами управления задачами (Issue Tracker):
      - Jira.

- Загрузка данных об ИБ-дефектах
  - Система должна предоставляет возможность загрузки данных об ИБ-дефектах следующими способами:

- запуск сканирования из графического web-интерфейса Системы (применительно для сканеров, автоматизирующих практики статического и композиционного анализа);
- загрузка результатов путем импорта данных через графический web-интерфейс Системы. Система поддерживает возможность импорта следующих форматов отчетов, получаемых от сканеров:
  - Solar appScreener: SARIF;
  - Positive Technologies Application Inspector: JSON;
  - CodeScoring: SBOM;
  - Trivy: JSON;
  - Kaspersky Container Security: JSON.
- Загрузка результатов сканирования, полученных из систем непрерывной сборки (Continuous Integration, CI);
- Система генерирует шаблоны скриптов, в которых указаны параметры интеграции Системы с следующими системами непрерывной сборки:
  - GitLab CI.
- Система осуществляет дедупликацию ИБ-дефектов:
  - получаемых от одного и того же сканера;
  - получаемых от разных сканеров, принадлежащих к одному типу (например, SAST, SCA, Image Scan).

- История сканирований
  - Система предоставляет информацию о всех сканированиях (как успешных, так и неуспешных), результаты которых импортировались в Систему в независимости от способа получения данных Системой;
  - Система предоставляет возможность фильтрации данных о реализованных сканированиях с учетом следующих критериев:
    - инициатор;
    - сканер;
    - результат.

- Система позволяет осуществлять множественный выбор критериев для формирования выборки данных, необходимой пользователю.

- Отображение информации об ИБ-дефектах
  - Система предоставляет нижеуказанную информацию об ИБ-дефектах, обнаруженных сканерами, автоматизирующими практики статического анализа (SAST):
    - уровень критичности ИБ-дефекта (установленная пользователем);
    - дата обнаружения ИБ-дефекта;
    - актуальность ИБ-дефекта на момент анализа;
    - статус ИБ-дефекта;
    - тип ИБ-дефекта;
    - номер задачи в Jira;
    - язык программирования;
    - информация об исключении ИБ-дефекта;
    - номер и ссылка на CWE. Ссылка должна перенаправлять на описание соответствующего недостатка в ПО, представленном на сайте https://cwe.mitre.org/;
    - исходный код файла, в котором был найден ИБ-дефект с указанием номера строки;
    - ссылка на исходный код файла, в котором был найден ИБ-дефект в системе контроля версии. Ссылка должна перенаправлять на файл в релевантной ветке (та, что выбрана пользователем в графическом web-интерфейсе Системы);
    - комментарии;
    - информация об ИБ-дефекте, получаемая от сканеров, которые его обнаружили:
      - степень критичности (установленная сканером);
      - наименование правила, при помощи которого был найден ИБ-дефект (в случае его наличия);
      - описание ИБ-дефекта (в случае его наличия);
      - рекомендации по устранению ИБ-дефекта (в случае наличия).

- Система предоставляет нижеуказанную информацию об ИБ-дефектах, обнаруженных сканерами, автоматизирующими практики композиционного анализа (SCA):
  - описание ИБ-дефекта;
  - уровень критичности ИБ-дефекта (установленная пользователем);
  - дата обнаружения ИБ-дефекта;
  - актуальность ИБ-дефекта на момент анализа;
  - статус ИБ-дефекта;
  - наличие exploit у ИБ-дефекта;
  - номер задачи в Jira;
  - технология (язык программирования или пакетный менеджер операционной системы);
  - информация об исключении ИБ-дефекта;
  - номер и ссылка на CWE. Ссылка должна перенаправлять на описание соответствующего недостатка в ПО, представленном на сайте https://cwe.mitre.org/;
  - наименование зависимости;
  - версия зависимости;
  - тип зависимости (прямая или транзитивная);
  - информация об уязвимости:
    - номер и ссылка на CVE. Ссылка должна перенаправлять на описание соответствующей уязвимости;
    - версия зависимости, в которой указанная уязвимость устранена;
    - дата публикации уязвимости;
    - дата изменения уязвимости.
  - CVSS-оценка и вектор уязвимости. Система должна предоставлять информацию согласно CVSS версии v2 и v3 (там, где это возможно);
  - Комментарии
  - Информация об ИБ-дефекте, получаемая от сканеров, которые его обнаружили:
    - степень критичности (установленная сканером);

- перечень ссылок (референсов), которые содержат информацию о рассматриваемой уязвимости.

- Система предоставляет нижеуказанную информацию об ИБ-дефектах, обнаруженных сканерами, автоматизирующими практики анализа образов контейнеров на предмет наличия в них уязвимостей (Image Scan):
  - описание ИБ-дефекта;
  - уровень критичности ИБ-дефекта (установленная пользователем);
  - наименование образа контейнера;
  - дата обнаружения ИБ-дефекта;
  - актуальность ИБ-дефекта на момент анализа;
  - статус ИБ-дефекта;
  - наличие exploit у ИБ-дефекта;
  - номер задачи в Jira;
  - технология (язык программирования или пакетный менеджер операционной системы);
  - информация об исключении ИБ-дефекта;
  - номер и ссылка на CWE. Ссылка должна перенаправлять на описание соответствующего недостатка в ПО, представленном на сайте https://cwe.mitre.org/;
  - наименование зависимости;
  - версия зависимости;
  - тип зависимости (прямая или транзитивная).
  - Информация об уязвимости:
    - номер и ссылка на CVE. Ссылка должна перенаправлять на описание соответствующей уязвимости;
    - версия зависимости, в которой указанная уязвимость устранена;
    - дата публикации уязвимости;
    - дата изменения уязвимости.
  - CVSS-оценка и вектор уязвимости. Система должна предоставлять информацию согласно CVSS версии v2 и v3 (там, где это возможно);
  - Комментарии;

- Информация об ИБ-дефекте, получаемая от сканеров, которые его обнаружили:
  - степень критичности (установленная сканером);
  - перечень ссылок (референсов), которые содержат информацию о рассматриваемой уязвимости.

- Управление ИБ-дефектами
  - Система предоставляет возможность получения информации об ИБ-дефектах для определенной ветки проекта без необходимости повторной настройки анализируемого проекта;
  - Система предоставляет возможность фильтрации выборки ИБ-дефектов на основе критериев:
    - ИБ-дефекты, обнаруженные сканерами, автоматизирующими практики статического анализа (SAST):
      - критичность;
      - тип;
      - статус;
      - сканер;
      - язык.
    - ИБ-дефекты, обнаруженные сканерами, автоматизирующими практики композиционного анализа (SCA):
      - критичность;
      - тип зависимости;
      - статус;
      - сканер;
      - технология.
    - ИБ-дефекты, обнаруженные сканерами, автоматизирующими практики анализа образов контейнеров на предмет наличия в них уязвимостей (Image Scan):
      - критичность;
      - статус;
      - сканер.

- Система позволяет осуществлять множественный выбор фильтров и/или критериев для формирования выборки данных, необходимой пользователю;
- Система позволяет пользователю изменять уровень критичности ИБ-дефекта;
- Система позволяет пользователю изменять статус ИБ-дефекта. В Системе должны присутствовать следующие статусы ИБ-дефекта:
  - Новый;
  - Подтвержден;
  - Ложное срабатывание;
  - В работе;
  - Риск принят;
  - Устранен;
  - Повторно найден;
  - Проигнорирован.
- Система позволяет добавлять ИБ-дефект в исключения на определенный пользователем срок;
- Система предоставляет пользователю возможность создания задач в подключенной системе управления задачами (Issue Tracker);
- Система позволяет выбирать пользователю несколько ИБ-дефектов (множественный выбор) для реализации следующих действий:
  - создание общей задачи в подключенной системе управления задачами (Issue Tracker);
  - создание нескольких задач в подключенной системе управления задачами (Issue Tracker);
  - изменение статуса;
  - исключение.
- Система предоставляет возможность создания Security Gate — механизма, позволяющего влиять на процесс сборки ПО в системах непрерывной сборки (Continuous Integration, CI);

- Дашборды и аналитика
  - Система предоставляет аналитическую информацию, представленную на следующих дашбордах:

- количество ИБ-дефектов (Подтвержден, Устранен, В работе);
- динамика по ИБ-дефектам;
- динамика отношения «Устраненных» ИБ-дефектов к «Новым и повторно найденным»;
- отношение «Подтвержденных» к «В работе»;
- динамика отношения «В работе» к «Устраненным»;
- время устранения ИБ-дефекта в днях;
- топ-3 повторно найденные группы ИБ-дефектов.

- Система предоставляет возможность выбора типов сканеров, данные которых будут использоваться при построении аналитических графиков:
  - SAST;
  - SCA;
  - Image Scan.
- Система предоставляет возможность выбора временного интервала, который будет использоваться при построении аналитических графиков:
  - неделя;
  - месяц;
  - квартал;
  - год.
- Система предоставляет возможность выбора ветки анализируемого проекта, для которой будут построены аналитические графики;

- Журналы событий
  - В системе реализовано логирование как минимум следующих действий:
    - Вход в Систему;
    - Выход из Системы;
    - Действия с Группами (создание, изменение, удаление).

5. Программно-аппаратные требования для установки и функционирования ПБР «Шерлок»

5.1. Аппаратные требования

5.1.1. Вариант установки Docker Compose

Для запуска ПБР «Шерлок» в данном варианте установки требуется минимум:

– ОС Linux. Рекомендуемая ОС — Debian 12;

– Docker;

– Docker Compose;

– CPU: 2 vCPU;

– Memory: 8 GB;

– Disk SSD: 50 GB.

5.1.2. Вариант установки Helm

Для запуска ПБР «Шерлок» в данном варианте установки требуется минимум:

– ОС Linux. Рекомендуемая ОС — Debian 12;

– Kubernetes;

– Helm;

– Утилита kubectl;

– CPU: 2 vCPU;

– Memory: 8 GB;

– Disk SSD: 50 GB.

5.2. Требования к установленному ПО

Для работы с Системой пользователь должен обладать:

– автоматизированным рабочим местом. Минимальные требования к автоматизированному рабочему месту:

– процессор с тактовой частотой не менее 2000 МГц;

– оперативная память не менее 4096 Мбайт.

– веб-браузером. Рекомендуется использовать следующие веб-браузеры (версия — актуальная на декабрь 2024 года):

- Google Chrome;
- Microsoft Edge;
- Mozilla Firefox;
- Яндекс Браузер.

6. Программное обеспечение, используемое для создания ПБР «Шерлок»

6.1. Языки программирования

- Java;
- JavaScript;
- Shell;
- SQL.

6.2. Технологии

- React, TypeScript, Ant Design, RTK Query;
- Spring Framework, MongoDB, PostgreSQL, ClickHouse, Minio;
- Docker, Kubernetes, Sonatype Nexus Repository Manager.

6.3. Интегрированные среды разработки

- JetBrains IntelliJ Idea;
- JetBrains WebStorm;
- Visual Studio Code.