ИНТЕЛЛЕКТУАЛЬНАЯ ПЛАТФОРМА ЗАЩИЩЕННОЙ РАЗРАБОТКИ ПРИЛОЖЕНИЙ SOLIDLAB

Руководство по эксплуатации

Листов 13

2024 г.

АННОТАЦИЯ

Настоящий документ является руководством по эксплуатации интеллектуальной платформы защищённой разработки приложений SolidLab (далее по тексту — SolidLab SDP, Платформа).

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ

1.1 Область применения

Область применения платформы SolidLab SDP – анализ защищённости информационных систем в процессе их разработки и управление выявленными недостатками защищённости.

1.2 Краткое описание возможностей

Интеллектуальная платформа защищенной разработки приложений SolidLab SDP предназначена для поиска и анализа недостатков в исходном коде с целью повышения уровня защищенности разрабатываемых информационных систем. Модули, входящие в состав Платформы, осуществляют инструментальный анализ защищенности приложений в процессе их разработки, а найденные недостатки передаются в единый интерфейс для управления находками. В данном интерфейсе пользователи Платформы могут ознакомиться с информацией о выявленных недостатках защищенности, управлять задачами по их устранению, в том числе приоритизировать данные задачи.

Платформа предоставляет программный интерфейс (API) для автоматизации задач инструментального анализа защищенности приложений.

Сценарии использования:

- Выявление недостатков защищенности в приложениях в процессе их разработки.
- Встраивание проверок защищенности в процессы разработки приложений.
- Оценка защищенности информационных систем на основе результатов инструментального анализа защищенности.
- Валидация обнаруженных недостатков и реализация процесса их устранения.

2 ПОДСИСТЕМЫ SOLIDLAB SDP

Платформа представляет из себя совокупность инструментов, сканеров, сервисов и средств автоматизации, гибко настраиваемых под потребности клиентов и предназначенных для выявления недостатков защищенности в приложениях.

Платформа состоит из набора независимых подсистем, что предоставляет возможность для:

- Использования только необходимых подсистем для решения конкретных задач.
- Изменения состава подсистем в процессе эксплуатации без простоя.
- Оптимизации выделенных для подсистем вычислительных мощностей.

В зависимости от решаемых задач в состав SolidLab SDP могут входить:

- Подсистема централизованной обработки и хранения обнаруженных недостатков защищенности, а также отображения результатов работы других подсистем (Defect Track).
- Подсистема поиска недостатков защищенности методами статического анализа исходного кода приложений.
- Подсистема динамического анализа приложений на уязвимости.
- Подсистема поиска оставленных секретов в исходном коде приложений.
- Подсистема анализа описаний инфраструктуры в виде исходного кода на наличие недостатков защищенности.
- Подсистема анализа используемых программных компонентов на известные уязвимости.
- Подсистема анализа компонентов в образах контейнеров на известные уязвимости.

3 ТРЕБОВАНИЯ К АППАРАТНОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

3.1 Требования к обеспечению рабочего места пользователя

Продукт предоставляется как сервис, который развёрнут в выделенном облаке и доступен при подключении с помощью VPN. SolidLab SDP является программным обеспечением типа «Программное обеспечение как услуга» и не требует установки на рабочие станции пользователей. Для работы с Платформой требуется браузер, клиент для протокола WireGuard и подключение к сети Интернет.

В случае если на рабочей станции пользователя нет доступа к сети Интернет или в операционной системе не установлен браузер или клиент WireGuard, то необходимо обратиться в отдел технической поддержки пользователей компании с запросом на предоставление доступа к сети Интернет и/или установку одного из веб-браузеров.

Рекомендуется устанавливать клиент WireGuard, согласно инструкции с сайта производителя – https://www.wireguard.com/install/

Рекомендуемые браузеры:

- Google Chrome версии 100 и выше.
- Firefox Browser версии 100 и выше.
- Яндекс.Браузер версии 21 и выше.
- Safari версии 14 и выше.

3.2 Требования к инфраструктуре

Для обеспечения корректной работы подсистем инструментального анализа Платформы (всех подсистем, кроме подсистемы централизованной обработки и хранения обнаруженных недостатков) необходимо предоставить доступ к соответствующим объектам анализа:

- К репозиторию исходного кода приложений для следующих подсистем:
  - Поиска недостатков защищенности методами статического анализа исходного кода приложений.
  - Поиска оставленных секретов в исходном коде приложений.
  - Анализа описаний инфраструктуры в виде исходного кода на наличие недостатков защищенности.
  - Анализа используемых программных компонентов на известные уязвимости.
- К экземпляру запущенного приложения для подсистемы динамического анализа приложений на уязвимости.
- К реестру образов Docker-контейнеров для подсистемы анализа компонентов в образах контейнеров на известные уязвимости.

При наличии аутентификации для доступа к объектам анализа необходимо также предоставить необходимые аутентификационные данные, например, логин и пароль или токен доступа. В том числе для обеспечения полноты работы подсистемы динамического анализа рекомендуется предоставить учетные данные пользователя для аутентификации в анализируемом приложении.

Влияние работы Платформы на системы хранения исходного кода или образов контейнеров минимально и не снижает их работоспособности. В ходе работы компонента динамического сканирования возможно влияние на анализируемые информационные системы. В связи с этим динамическое сканирование рекомендуется выполнять на тестовых стендах, планировать заранее и проводить в периоды минимальной загрузки сервисов во избежание их перегрузки.

Для автоматизации запуска подсистем инструментального анализа Платформы в рамках CI/CD на машине, автоматизирующей запуск, необходима установка клиента протокола WireGuard и наличие HTTP-клиента для взаимодействия с API Платформы. В качестве HTTP-клиента рекомендуется программа cURL.

4 ДОСТУП К ПЛАТФОРМЕ

Развёртывание и настройку Платформы осуществляет инженер компании SolidLab (далее – инженер). На этапе развёртывания SolidLab SDP осуществляется подготовка инфраструктуры (виртуальных машин) для подсистем Платформы.

Инженер создаёт клиенту учетную запись для взаимодействия с Платформой, а именно для запуска подсистем анализа и просмотра их результатов в подсистеме централизованной обработки и хранения обнаруженных недостатков защищенности.

Клиенты отслеживают результаты работы подсистем Платформы через интерфейсы подсистемы централизованной обработки и хранения обнаруженных недостатков защищенности, и на основании обнаруженных недостатков принимают решения об изменениях в информационных системах, анализируемых Платформой.

Для доступа к веб-интерфейсу подсистемы централизованной обработки и хранения обнаруженных недостатков защищенности необходимо в адресной строке браузера ввести путь, который имеет вид https://defecttrack.<platform name>, где <platform name> – наименование Платформы, которое передается клиенту вместе с учётными данными для входа.

Для аутентификации в этом веб-интерфейсе необходимо ввести в поля формы «Имя пользователя» и «Пароль» данные переданной учетной записи.

5 РАБОТА С ПЛАТФОРМОЙ

Для запуска сканирования необходимо направить HTTP-запрос с телом в формате JSON на API-интерфейс Платформы, расположенный по адресу: https://sdp.<platform_name>/api/v3/scan/start, где <platform_name> – наименование Платформы, которое передается клиенту вместе с учётными данными для входа.

При взаимодействии с подсистемами Платформы клиенты могут использовать три типовых сценария:

1. Первый сценарий рекомендуется выполнять для анализа исходного кода приложения:

1.1. Получение исходного кода из репозитория исходного кода.
1.2. Параллельный анализ выбранными подсистемами.
1.3. Выгрузка результатов анализа в подсистему централизованной обработки и хранения обнаруженных недостатков защищенности.

Данный сценарий распространяется на следующие подсистемы:

- Поиска недостатков защищенности методами статического анализа исходного кода приложений.
- Поиска оставленных секретов в исходном коде приложений.
- Анализа описаний инфраструктуры в виде исходного кода на наличие недостатков защищенности.
- Анализа используемых программных компонентов на известные уязвимости.

2. Второй сценарий рекомендуется выполнять для анализа Docker-образов:

2.1. Получение Docker-образа кода из репозитория образов.
2.2. Выполнение анализа компонентов в образе Docker-контейнера на известные уязвимости.
2.3. Выгрузка результатов анализа в подсистему централизованной обработки и хранения обнаруженных недостатков защищенности.

Данный сценарий распространяется на подсистему анализа компонентов в образах контейнеров на известные уязвимости.

3. Третий сценарий рекомендуется выполнять для анализа запущенных экземпляров приложений:

3.1. Выполнение динамического анализа приложений на уязвимости.
3.2. Выгрузка результатов анализа в подсистему централизованной обработки и хранения обнаруженных недостатков защищенности.

Данный сценарий распространяется на подсистему динамического анализа приложений на уязвимости.

Пример входного JSON-файла для запуска сканирования по первому сценарию (подсистемой статического анализа и подсистемой поиска секретов в исходной коде) для приложения VulnerableApp, расположенного в репозитории исходного кода по адресу: https://github.com/SasanLabs/VulnerableApp.git, представлен ниже:

```
{
    "template": "global-tasks",
    "user": "<имя-клиента>",
    "client_name": "<имя-клиента>",
    "project_name": "vulnerableapp",
    "tariff": "t2",
    "scan_plan":  {
        "scenario": "sast/mast/secret/infra/ost",
        "scanners": [ "semgrep", "gitleaks" ],
        "source": {
            "type": "git",
            "url": "https://github.com/SasanLabs/VulnerableApp.git"
        },
        "export": {
            "type": "defecttrack",
            "host": "https://defecttrack.<platform name>"
        }
    }
}
```

Примеры HTTP запросов с JSON-телом для запуска каждой подсистемы Платформы представлены в руководстве пользователя.

Адрес репозитория исходного кода для анализа может быть указан в значении поля scan_plan.source.url.

Стоит учесть, что для корректного завершения этапа клонирования исходного кода для сканирования и корректной работы компонента анализа образов контейнеров, необходимо предоставить соответствующие токены доступа к репозиторию исходного кода или к реестру образов контейнеров.

Подсистемы, требуемые для анализа приложения Платформы, указываются в массиве scan_plan.scanners.

В примере запуска SolidLab SDP выше выполняется одновременное обращение к подсистемам поиска недостатков защищенности методами статического анализа исходного кода приложений и поиска оставленных секретов в исходном коде приложений.

Функция выполнения нескольких сканирований реализуется с помощью очереди задач. При отправке нескольких HTTP-запросов к API Платформы первый запрос на сканирования будет взят в работу, а остальные попадут в очередь и будут ожидать освобождения ресурсов.

Пример входного тела запроса, который запускает подсистемы динамического анализа приложений на уязвимости для тестового веб-приложения juice-shop представлен ниже.

```
{
    "template": "global-tasks",
    "user": "<имя-клиента>",
    "client_name": "<имя-клиента>",
    "url_for_scan": "https://juice-shop.herokuapp.com/",
    "project_name": "juice-shop",
    "tariff": "t2",
    "scan_plan":  {
        "scenario": "dast/image",
        "scanners": [ "zap" ],
        "export": {
            "type": "defecttrack",
            host": "https://defecttrack.<platform name>"
        }
    }
}
```

По окончании сканирования осуществляется выгрузка результатов в подсистему обработки и хранения обнаруженных недостатков. Для разных запусков подсистем Платформы можно указывать одно и тоже имя проекта в параметре "project-name": в таком случае результаты анализа нескольких подсистем будут объединены в одну группу (проект) в подсистеме обработки и хранения обнаруженных недостатков. Это подсистема предоставляет следующие возможности:

- Выполнения ручной верификации найденных недостатков и анализа критичности недостатков.
- Реализации процесса отслеживания исправления выявленных недостатков защищенности.
- Настройки гранулярных прав доступа к результатам.
- Отображения статистики по выявленным недостаткам защищенности.
- Создания отчетов по результатам анализа и верификации выявленных недостатков защищенности.

Характеристики анализа защищенности, выполняемого подсистемой поиска недостатков защищенности методами статического анализа исходного кода приложений:

- Поиск недостатков защищенности методами статического анализа без компиляции исходного кода приложений.
- Поиск недостатков защищенности методами статического анализа для различных языков программирования, включая: C, C++, C#, Go, Java, JavaScript, Kotlin, TypeScript, Ruby, Rust, PHP, Python, Scala, Swift.
- Поиск различных недостатков защищенности методами статического анализа, включая: инъекции (в том числе XSS, SQLi, SSTI), недостатки десериализации пользовательских данных, SSRF, XXE, RCE.

Характеристики анализа защищенности, выполняемого подсистемой динамического анализа приложений на уязвимости:

- Составление поверхности атаки с помощью различных методов, включая поиск входных точек с помощью статического и динамического обхода веб-приложений.
- Поиск различных недостатков защищенности, включая: Reflected XSS, DOM-based XSS, SQL-инъекция, XXE, Path Traversal.

Характеристики анализа защищенности, выполняемого подсистемой поиска оставленных секретов в исходном коде приложений:

- Поиск оставленных секретов в исходном коде приложений и в истории коммитов.

Характеристики анализа защищенности, выполняемого подсистемой анализа описаний инфраструктуры в виде исходного кода на наличие недостатков защищенности:

- Анализ защищенности описаний инфраструктуры в виде исходного кода, включая описания инфраструктуры для следующих систем: Terraform, Kubernetes, Docker, Ansible.

Характеристики анализа защищенности, выполняемого подсистемой анализа используемых программных компонентов на известные уязвимости:

- Поиск известных уязвимостей в программных компонентах, используемых в приложениях, собираемых с помощью систем, включая: npm, maven, gradle, pypi, composer, go.

Характеристики анализа защищенности, выполняемого подсистемой анализа компонентов в образах контейнеров на известные уязвимости:

- Анализ компонентов в образах Docker-контейнеров на известные уязвимости.

6 КОНТАКТЫ

Для получения руководств пользователя и пользователя Defect Track, в которых детально описано взаимодействие с Платформой, необходимо направить обращение на электронный адрес info@solidlab.ru.