НАЗВАНИЕ

Дата выполнения: дата

Выполнил: ФИО

Оглавление

[Введение 2](#__RefHeading___Toc95_1065274009)

[1 Концепция Shift Left 3](#__RefHeading___Toc97_1065274009)

[2 Структура DevSecOps-пайплайна 5](#__RefHeading___Toc99_1065274009)

[2.1 Pre-commit проверки 5](#__RefHeading___Toc101_1065274009)

[2.2 Pre-build проверки 6](#__RefHeading___Toc103_1065274009)

[2.3 Test-time проверка DAST 10](#__RefHeading___Toc105_1065274009)

[2.4 Post-deploy проверки 12](#__RefHeading___Toc329_367963048)

# 

# Введение

Взят проект сайта \_\_\_\_\_\_\_\_ и сделан форк \_\_\_\_\_\_\_\_\_\_ для дальнейшей работы.

Необходимо провести анализ кода на уязвимости. Настроить pipline для проекта и встроить в него статические (SAST) и динамические (DAST) тесты. Описать процессы тестирования, провести аналитику результатов тестов.

# 

# Этап 1. CI/CD

В проекте задействованы общедоступные сервисы:

* GitHub – хранилище репозитория
* GitLab – построение процессов сбори, тестирования и доставки (CI/CD )
* Snyk – проверка open-source кода и поиск уязвимостей
* Semgrep – инструмент статического анализа с открытым исходным кодом
* Trivy – сканер уязвимостей в контейнерах, Git репозиториях, поиск секретов
* ZAP (Zed Attack Proxy) - OWASP open-source решение, которое в том числе используется в GitLab DAST

## Настройка пайплайн по сборке и доставке программного обеспечения

На данном этапе мы используем сервис Gitlab.com. Используем пробную версию.

Для работы нам необходимо импортировать наш готовый проект расположенный на Githab (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_). В меню Code -> Repository мы увидим клонированный проект.

Следующим этапом идет настройка runners (раннеров или бегунков) для выполнения основных работ в меню Settings -> CI/CD -> Runners. Runner можно настроить на локальной машине, виртуальной или в облаке. В проекте будет использованы ресурсы Yandex.Cloud. Описание настроект будет описано ниже в пункте \_\_\_\_\_\_\_\_\_

После подключения runners необходимо подготовить gitlab.yaml файл, в котором будут описаны все этапы работ, включая этапы тестирования. Пример gitlab.yaml файла будет приведен в приложении

## Настройка облачных сервисов

На данном этапе мы используем сервис Yandex.Cloud. Для проекта будет использоваться грант предоставляемый компанией Яндекс для начального использования облачных сервисов.

Создание проекта. Выбор тестовый период с предоставленным грантом.

Переходим в Yandex Cloud Console. Создаем ресурс -> Виртуальная машина (ВМ).

Операционная система - Ubuntu 22.04

Диски и файловые хранилища – HDD, 18ГБ

Вычислительные ресурсы – Intel IceLake, vCPU -2, 20%, RAM -2ГБ

Сетевые настройки – по умолчанию

Доступ – ввести логин и SSH-ключ (генерация ssh ключа – ssh-keygen –t ed25519; посмотреть ключ – cat ~/.ssh/id\_ed25519.pub )

Запущен процесс создания новой ВМ. После создания ВМ проведем дальнейшие настройки и установим необходимое программное обеспечение. Для этого произведем подключение к ВМ по протоколу ssh:

ssh@<публичный IP адрес> (посмотреть публичный IP можно в настройках ВМ в Yandex Cloud Console)

Обновление системы – apt get update

Установка docker -

Установка gitlab runner -

Проверить доступность runner можно на Gitlab в Settings -> CI/CD -> Runners. Активные и доступные runners будут подсвечены зеленым, неактивные или недоступные подсвечены серым.

# Этап 2. SAST

## Snyk

## Trivy

## Semgrep

## Secret detection

# Этап 3. DAST

## ZAP

# Итог