Тестовое задание на направление DevSecOps

Задание 1

1. Установить Jenkins, SAST – Semgrep, SCA – Dependency Check, Container Scanning – Trivy, DAST – OWASP ZAP

2. Разработать Dockerfile для сборки веб-приложения (код веб-сервиса можно взять из открытых источников, на любом из языков программирования). Использование приложений на микросервисной архитектуре – приветствуется. При разработке Dockerfile, необходимо ориентироваться на лучшие практики по безопасности: (например, https://sysdig.com/blog/dockerfile-best-practices/).
3. Написать CI/CD Pipeline, который включает шаги:
    - Checkout SCM – выгрузка кода из вашего github/gitlab репозитория.
    - SAST Scan (Semgrep) – проведение статического анализа исходного когда на уязвимости.
    - SCA Scan (Dependency Check) – проведение сканирования зависимостей веб-приложения.
    - Build – сборка веб-приложения через Dockerfile.
    - Container Scanning (Trivy) – проведение сканирования компонентов в файловой системе образа контейнера.
    - Publishing – экспорт собранного образа в public пространство DockerHub.
    - Deployment – запуск полученного контейнера на хостовой машине.
    - DAST Scanning (OWASP ZAP) – проведение динамического сканирования запущенного в контейнера Веб – приложения.

    Полученные отчеты Security сканеров рекомендуется сохранять в workspace Jenkins для дальнейшей ручной загрузки в Github репозиторий.

4. Pipeline необходимо параметризировать в части выбора репозитория и ветки/тег, из которой будет производиться запуск. Использование Scripted либо Declarative style pipeline на Ваш выбор.

Задание 2

1. Установить Minikube на вашу хостовую машину.
2. Установить PodSecurityPolicy для Minikube, согласно мануалу: https://minikube.sigs.k8s.io/docs/tutorials/using_psp/
3. Разработать Kubernetes templates yaml, которые будут запускать веб-сервис из задания 1, включающий в себя:
    - Deployment с Security Contexts из PodSecurityPolicy Restricted: (privileged: false, RunAsUser: 10000, fsGroup: 10000, SeLinux Rules, etc.), работающий на 8080 порту и вытягивающий образ из Dockerhub.
    - Маунт в контейнер Configmap и Secret (опционально).
    - Service по типу ClusterIP.
    - Ingress для публикации сервиса по HTTP.
4. Проверить работу сервиса по адресу https://0.0.0.0:8080/

5. * Доработать Jenkinsfile из задания 1, поменяв этап с деплоем на инсталляцию в Minikube при помощи полученных темплейтов. Использование Jenkins Kubernetes Plugin – приветствуется.

** – задание повышенной сложности, которое дает дополнительный бонус.*

Что необходимо для выполнения: виртуальная/хостовая машина Linux.

Ожидаемый результат выполнения задания: приглашение/URL на приватный репозиторий, в котором находятся:

- Исходный код приложения;
- Dockerfile для сборки образа контейнера Docker;
- Jenkinsfile;
- Kubernetes yaml файлы;
- Результаты отчетов Security сканеров;
- Ссылка на собранный образ в DockerHub;
- Вспомогательные скрипты и файлы;
- Файл с описанием шагов инсталляции сервисов и пошаговая инструкция по запуску Jenkins Pipeline с нуля.