Текущие задачи УИБ - Задачи #1799

DevSecOps

25 июня 2021 16:10 - Никита Цыкунов

Статус:	В работе	**Дата начала:**	25 июня 2021
Приоритет:	Нормальный	**Срок завершения:**	
Назначена:	Никита Цыкунов	**Готовность:**	0%
Категория:		**Оценка временных затрат:**	0.00 час
Версия:		**Трудозатраты:**	0.00 час
Категория задач:		**Подразделение:**	УИБ

Описание

Определение технологического стека и запуск инструментов в тестовом окружении, для определения и построения процессов безопасной разработки.

История

#1 - 25 июня 2021 16:22 - Никита Цыкунов

Никита Цыкунов писал(а):

> Определение технологического стека и запуск инструментов в тестовом окружении, для определения и построения процессов безопасной разработки.

Коммерческие решения:

ShiftLeft - https://www.shiftleft.io/
AppScreener - https://rt-solar.ru/products/solar_appscreener/capabilities/
CheckMarx - https://www.checkmarx.com/ru/products/static-application-security-testing/

Open source или shareware решения:

Поиск секретов в git – trufflehog

Поиск уязвимых библиотек – owasp dependency-check

Анализ исходного кода:

sonarqube - https://www.sonarqube.org/

SemGrep - https://semgrep.dev/

CodeQL - https://securitylab.github.com/tools/codeql/

DAST:

MobSF - https://github.com/MobSF/Mobile-Security-Framework-MobSF

IAST:

Contrast - https://www.contrastsecurity.com/interactive-application-security-testing-iast

Система управления уязвимостями:

archery - https://www.archerysec.com/

defectdojo - https://github.com/DefectDojo/django-DefectDojo

Обзор инструментов:

https://habr.com/ru/company/swordfish_security/blog/518758/

Полезные ссылки:

https://docs.gitlab.com/runner/register/index.html

https://docs.gitlab.com/runner/install/docker.html

https://docs.sonarqube.org/latest/analysis/gitlab-integration/

https://docs.sonarqube.org/8.5/analysis/gitlab-cicd/

https://coderoad.ru/39875287/%D0%97%D0%B0%D0%BF%D1%83%D1%81%D1%82%D0%B8%D1%82%D0%B5-%D1%81%D0%BA%D0%B0%D0%BD%D0%B5%D1%80-sonarqube-%D1%81-%D0%BF%D0%BE%D0%BC%D0%BE%D1%89%D1%8C%D1%8E-gitlab-ci

https://gitlab.com/gitlab-org/gitlab/-/issues/23911

https://habr.com/ru/post/542676/

https://habr.com/ru/post/429252/

https://habr.com/ru/post/432820/

https://habr.com/ru/company/swordfish_security/blog/541554/

https://habr.com/ru/company/swordfish_security/blog/518758/

https://habr.com/ru/company/swordfish_security/blog/516660/

https://www.youtube.com/watch?v=00EoUvf4cEo&t=205s

https://notsosecure.com/achieving-devsecops-with-open-source-tools/

https://cuckoo-droid.readthedocs.io/en/latest/

https://www.youtube.com/watch?v=dhu5iIQEymU

https://github.com/linkedin/qark

https://www.netguru.com/codestories/android-security-an-overview-of-static-analysis-tools-part-one

https://m.habr.com/ru/company/pt/blog/332904/

https://habr.com/ru/company/jetinfosystems/blog/499762/

https://medium.com/@maxy_ermayank/running-ci-server-jenkins-nexus-sonarqube-at-scale-using-docker-swarm-part-1-7629271f178a

https://github.com/devopssecure/webapp

https://www.youtube.com/watch?v=llQH7R_5JNE

https://nullsweep.com/creating-a-secure-pipeline-jenkins-with-sonarqube-and-dependencycheck/

https://github.com/hysnsec/DevSecOps-Studio

https://docs.fastlane.tools/actions/sh/

https://www.digitalocean.com/community/tutorials/how-to-use-traefik-as-a-reverse-proxy-for-docker-containers-on-ubuntu-18-04-ru

#2 - 25 июня 2021 16:24 - Никита Цыкунов

- Параметр Тема изменился с DevSecOsp на DevSecOps

#3 - 28 июня 2021 09:17 - Никита Цыкунов

Доступ к docker swarm

Проверен сетевой доступ до сервера и отправлена открытая часть ssh ключа для добавления на сервер.

#4 - 28 июня 2021 17:45 - Никита Цыкунов

Развернуть gitlab-ce

Источники:
https://lunar.computer/posts/gitlab-docker-swarm/
https://mcs.mail.ru/help/ru_RU/cases-gitlab/case-gitlab
https://docs.gitlab.com/omnibus/docker/
https://medium.com/@elinah/%D0%BB%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D0%B7%D0%B0%D0%BF%D1%83%D1%81%D0%BA-gitlab-%D1%81-docker-compose-5c2ede6cbe2b
https://docs.gitlab.com/runner/configuration/tls-self-signed.html#supported-options-for-self-signed-certificates-targeting-the-gitlab-server

https://docs.gitlab.com/ee/security/reset_user_password.html
https://qna.habr.com/q/514971
https://stackoverflow.com/questions/23885449/unable-to-resolve-unable-to-get-local-issuer-certificate-using-git-on-windows
https://docs.gitlab.com/ee/administration/troubleshooting/ssl.html
https://docs.gitlab.com/omnibus/settings/environment-variables.html

##===

Созданы папки и файлы на общем ресурсе:

/mnt/docker/devsecops/gitlab-test/data (данные gitlab)
/mnt/docker/devsecops/gitlab/logs (логи gitlab)
/mnt/docker/devsecops/gitlab/config (настройки gitlab и omnibus)
/mnt/docker/devsecops/gitlab/ssl (ssl сертификат для 192.168.6.141)
/mnt/docker/devsecops/gitlab-runner/data (данные gitlab-runner)
/mnt/docker/devsecops/gitlab/config (настройки gitlab-runner)
/mnt/docker/devsecops/gitlab.rb (настройки gitlab)
/mnt/docker/devsecops/swarm-gitlab.yml (запуск gitlab и gitlab-runner)
/mnt/docker/devsecops/gitlab_admin_pass.txt (пароль администратора gitlab)

Проброшены порты:

4822:22, 4880:80, 48443:443

UPD: заменены 80:80, 443:443, в связи с проблемами адресации gitlab-runner запущенного в docker и образов запускаемых внутри runner из образов docker :)

Выпущен ssl сертификат и установлен на рабочее место:

Создан конфиг для openssl
"sudo cp /etc/ssl/openssl.cnf /mnt/docker/devsecops/gitlab-test/ssl/gitlab.local.cnf"

В конфиг внесены следующие изменения:

[req]

req_extensions = v3_req # раскоментирована строка

[v3_req]

subjectAltName = @alt_names # добавлена строка

[alt_names] # Добавлено в конце файла

DNS.0 = gitlab
DNS.1 = *.gitlab
DNS.2 = www.gitlab
IP.1 = 192.168.6.141

Команда для формирования сертификата:

"sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /mnt/docker/devsecops/gitlab/ssl/gitlab.local.key -out
/mnt/docker/devsecops/gitlab/ssl/gitlab.local.crt -extensions v3_req -config /mnt/docker/devsecops/gitlab/ssl/gitlab.local.cnf"

Права на ключ сертификата:

sudo chmod 400 /mnt/docker/devsecops/gitlab/ssl/gitlab.local.key

P.S:

Манипуляции с alt_names необходимы для того, чтобы браузеры принимали самоподписанные сертификаты.

Начиная с Chrome 58, самоподписанный сертификат должен иметь правильное доменное имя в поле Subject Alternative Name (SAN).

##===

Конфигурация gitlab.rb

external_url 'https://192.168.6.141'
letsencrypt['enable'] = false
nginx['ssl_certificate'] = "/etc/gitlab/ssl/gitlab.local.crt"
nginx['ssl_certificate_key'] = "/etc/gitlab/ssl/gitlab.local.key"
gitlab_rails['initial_root_password'] = File.read('/run/secrets/gitlab_root_password')

##===

Зарегистрирован 1 раннер:

подключение к контейнеру: docker exec -it <Имя контейнера>

регистрация: gitlab-runner register --tls-ca-file=/home/gitlab_ci_multi_runner/data/gitlab.local.crt

~~нужно в файл /etc/gitlab-runner/config.toml, добавить extra_hosts=["домен днс:ip dns"]. Так как контейнер с runner-ом и контейнер запускаемый внутри контейнера с runner-ом используют разные dns.~~

Были проблемы с сетью, решено.

#5 - 30 июня 2021 14:31 - Никита Цыкунов

- Файл gitlab-test.yml добавлен

#6 - 30 июня 2021 16:00 - Никита Цыкунов

- Файл swarm-gitlab.yml добавлен

#7 - 01 июля 2021 09:53 - Никита Цыкунов

- Параметр Статус изменился с Новая на В работе

#8 - 02 июля 2021 17:47 - Никита Цыкунов

Решение проблемы с доступом в интернет на контейнерах

Не было доступа в интернет в стеке devsecops.

Решение:

Удаление всех стеков в кластере и перезагрузка сервиса docker:

sudo docker stack ls (Просмотр стеков)
sudo docker stack rm <имя стека> (Удаление стека)
sudo service docker restart (Перезагрузка службы docker, необходимо выполнить на всех нодах)
sudo docker stack deploy -c <путь до yml файла стека> <имя стека> (Создание стеков)

#9 - 02 июля 2021 17:54 - Никита Цыкунов

Создание репозиториев с учебными приложениями

Созданы и перенесены данные из 2 репозиториев:

webapp
https://github.com/cehkunal/webapp

WebGoat
https://github.com/WebGoat/WebGoat

Команды для создания и переноса проекта:

- Создать папку на компьютере, для проектов
- git clone https://192.168.6.141/root/webgoat.git (скачать репозиторий)
- cd webgoat
- Поместить файлы приложения в папку
- git switch -c main (переключение ветки)
- git add . (добавление всех файлов из папки в репозиторий)
- git commit -m "add project WebGoat" (создание комита)
- git push -u origin main (загрузка данных в репозиторий)

#10 - 02 июля 2021 17:55 - Никита Цыкунов

Сборка учебных приложений

~~Необходим доступ к~~ https://repo.maven.apache.org/maven2

Добавил настройки прокси в gitlab.rb, так как не использует системные настройки прокси.

https://docs.gitlab.com/omnibus/settings/environment-variables.html

Добавил настройки прокси в конфигурацию стэка. (на всякий случай)

Добавил настройки прокси в gitlab-runner.

environment = ["https_proxy=http://192.168.6.15:3128", "http_proxy=http://192.168.6.15:3128", "HTTPS_PROXY=http://192.168.6.15:3128", "HTTP_PROXY=http://192.168.6.15:3128",
"no_proxy=localhost,hids-elk,hids-elk-1,hids-elk-2,hids-elk-3,127.0.0.1,192.168.206.50,192.168.206.51,192.168.206.52,*.mbrd.ru,* .mtsbank.ru,gitlab,gitlab-runner,repo.maven.apache.org",
"NO_PROXY=localhost,hids-elk,hids-elk-1,hids-elk-2,hids-elk-3,127.0.0.1,192.168.206.50,192.168.206.51,192.168.206.52,*.mbrd.ru,* .mtsbank.ru,gitlab,gitlab-runner,repo.maven.apache.org"]

Добавил настройки прокси в при вызове mvn, так как этот инструмент не использует системные настройки прокси.

mvn -Dhttp.proxyHost=192.168.6.15 -Dhttp.proxyPort=3128 -Dhttps.proxyHost=192.168.6.15 -Dhttps.proxyPort=3128 clean package

##===

Сборка приложения из примера запустилась на более старой версии maven (3.8.1-jdk-11-slim)

#11 - 05 июля 2021 15:38 - Никита Цыкунов

Загрузка приложения на "сервер"

Реализован шаг Deploy с загрузкой .war файла в промотированную к нескольким контейнерам папку.

Имитация загрузки кода на сервер.

#12 - 06 июля 2021 16:58 - Никита Цыкунов

Реализация шага поиска секретов в комитах

Реализован поиск секретов в комитах gitlab/github.

В шаге используется образ python:3-alpine

и модуль python "trufflehog":

- apk add --no-cache git && pip install gitdb2==3.0.0 trufflehog

В виду самоподписанного сертификата отключена проверка ssl в git:

- git config --global http.sslVerify false

Запуск сканирования:

- trufflehog --regex --entropy=False **--status_on_failures=True** --json https://gitlab/root/webapp.git > /mnt/devsecops/trufflehog_log
- cat /mnt/devsecops/trufflehog_log

P.S.

По задумке авторов trufflehog, если находятся секреты, то программа возвращает exit 1, в связи с этим в тестовом окружении шаг пропускается "allow_failure: true"

https://github.com/trufflesecurity/truffleHog/issues/87

##===

P. S. S
Доработал Trufflehog для возможности выбора кода выхода после того как он находит серкеты.

За основу брал данные репозитории:

https://github.com/calve/truffleHog
https://github.com/tmsteen/truffleHog

Исходный код доработанной версии во вложении.

#13 - 06 июля 2021 17:18 - Никита Цыкунов

- Файл gitlab-ci.yml добавлен

#14 - 07 июля 2021 12:41 - Никита Цыкунов

- Файл trufflehog-main.zip добавлен

#15 - 07 июля 2021 14:56 - Никита Цыкунов

Реализация шага поиска уязвимых зависимостей

Источники:

https://gitlab.com/gitlab-ci-utils/docker-dependency-check/-/tree/master (пример использования)
https://jeremylong.github.io/DependencyCheck/dependency-check-cli/index.html (cli утилита)
https://docs.gitlab.com/ee/ci/docker/using_docker_images.html (entrypoint)
https://jeremylong.github.io/DependencyCheck/dependency-check-cli/arguments.html (аргументы cli)

##==========

Использовал образ и docker hub "owasp/dependency-check:latest" с пустой точкой монтирования entrypoint: [""], для возможности запуска с proxy(необходимо для скачивания и обновления правил с CVE).

Команда для запуска:

/usr/share/dependency-check/bin/dependency-check.sh --scan "./" --format ALL --project "$CI_PROJECT_NAME" --proxyserver 192.168.6.15 --proxyport 3128 --format JSON --out /builds/root/webapp/

#16 - 07 июля 2021 17:24 - Никита Цыкунов

Реализация шага статического анализа кода

Интеграция Appscreener

Необходим доступ до appscreener.mbrd.ru по 80 и 443 портам

##==========

Интеграция SonarQube

https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-maven/

#17 - 07 июля 2021 17:25 - Никита Цыкунов

Развернуть и настроить sonarqube

Источники:

https://www.digitalocean.com/community/tutorials/how-to-ensure-code-quality-with-sonarqube-on-ubuntu-18-04
https://qna.habr.com/q/542064
https://stackoverflow.com/questions/24288616/permission-denied-on-accessing-host-directory-in-docker

##==========

Создание пользователя sonarqube для доступа к ресурсам на хосте

sudo adduser --system --no-create-home --group --disabled-login sonarqube

id sonarqube (выводит id пользователя и его группы), чтобы использовать кастомный id в контейнере, его нужно пересобирать.

Так как id пользователя sonarqube в контейнере 999, а на хосте 112 и он не может получать доступ к папкам внутри "permission denied".

На текущий момент на папку /mnt/docker/devsecops/sonar сменен владелец на 999:999, что соответствует пользователю systemd-coredump

Созданы следующие директории:

/mnt/docker/devsecops/sonar/data

/mnt/docker/devsecops/sonar/extensions
UPD: прим монтировании папки extensions перестают скачиваться плагины
https://community.sonarsource.com/t/cant-create-a-quality-profile-because-there-are-no-languages-available/30674/3

/mnt/docker/devsecops/sonar/logs
/mnt/docker/devsecops/sonar/postgresql
/mnt/docker/devsecops/sonar/temp

Владелец каталогов на хосте systemd-coredump в контейнере sonarqube.

Проброшен порт 4890:9000

##==========

Изменен стандартный пароль, сохранен в teampass.

Сгенерирован токен для запуска сканирования "devsecops", токен сохранен в teampass.

Создан проект для сканирования devsecops, команда для запуска:

mvn sonar:sonar -Dsonar.projectKey=devsecops -Dsonar.host.url=http://192.168.6.141:4890 -Dsonar.login=TOKEN

#18 - 08 июля 2021 11:37 - Никита Цыкунов

- Файл swarm-sonar.yml добавлен

- Файл tomcat-metasploitable.yml добавлен

#19 - 08 июля 2021 14:59 - Никита Цыкунов

- Файл gitlab-ci (1).yml добавлен

#20 - 08 июля 2021 17:01 - Никита Цыкунов

Развернуть и интегрировать defectdojo

Источник:

https://github.com/DefectDojo/django-DefectDojo
https://github.com/cloudchefs/docker-defect-dojo

#21 - 08 июля 2021 17:02 - Никита Цыкунов

Развернуть и интегрировать archery

Источник:

https://github.com/archerysec/archerysec

#22 - 08 июля 2021 17:03 - Никита Цыкунов

Развернуть и интегрировать dependencytrack

Источник:

https://docs.dependencytrack.org/getting-started/deploy-docker/

##==========

Dependency-Track - это интеллектуальная платформа анализа компонентов, которая позволяет организациям выявлять и снижать риски в цепочке поставок программного обеспечения. Dependency-Track использует уникальный и очень полезный подход, используя возможности Software Bill of Materials (SBOM). Этот подход предоставляет возможности, которых не могут достичь традиционные решения Software Composition Analysis (SCA).

##==========

~~Не запустилась на тестовом стенде из за прокси(не может скачать зависимости, прописал http_proxy, не помогло).~~

```
# - ALPINE_HTTP_PROXY_ADDRESS=proxy.example.com
# - ALPINE_HTTP_PROXY_PORT=8888
```

Необходимо настроить БД.

#23 - 08 июля 2021 17:04 - Никита Цыкунов

Интегрировать semgrep

Источник:

https://semgrep.dev/docs/getting-started/

#24 - 09 июля 2021 12:11 - Никита Цыкунов

Интегрировать sheftleft scan

https://slscan.io/en/latest/

##==========

```
image: shiftleft/sast-scan:latest # официальный образ
script:
    - mvn -Dhttp.proxyHost=192.168.6.15 -Dhttp.proxyPort=3128 -Dhttps.proxyHost=192.168.6.15 -Dhttps.proxyPort
=3128 clean install # сборка приложения + class файлы
    - scan --build --no-error # сканирование кода с продолжением pipeline в случае нахождения уязвимостей
```

Файлы

gitlab-test.yml	936 байта	30 июня 2021	Никита Цыкунов
swarm-gitlab.yml	1,19 КБ	30 июня 2021	Никита Цыкунов
gitlab-ci.yml	1,71 КБ	06 июля 2021	Никита Цыкунов
trufflehog-main.zip	16,7 КБ	07 июля 2021	Никита Цыкунов
swarm-sonar.yml	974 байта	08 июля 2021	Никита Цыкунов
tomcat-metasploitable.yml	409 байта	08 июля 2021	Никита Цыкунов
gitlab-ci (1).yml	2,81 КБ	08 июля 2021	Никита Цыкунов