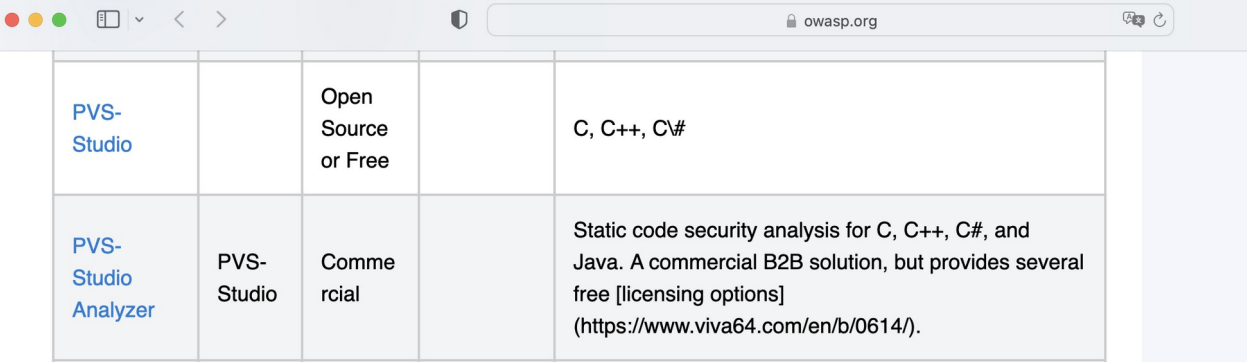**Лабораторная работа 6**

по дисциплине

«Автоматизация процессов жизненного цикла программных средств»

Выполнил:

ст. гр. 2хх-3хх

Иванов И. И.

Проверил:

Красников А. С.

Москва – 2024

<u>План выполнения лабораторной работы</u>

1. Провести исследования доступных статических анализаторов для выбранного командой стека разработки (большой, но не всеобъемлющий [список] (https://owasp.org/www-community/Source_Code_Analysis_Tools))

2. Выбрать минимум два из следующих видов статических анализаторов:

a.   анализатор качества кода (пример - Checkstyle)

b.   анализатор безопасности кода (Пример - Gosec)

c.   средство поиска учетных данных/секретов (Пример - Trufflehog).

Анализатор PVS-Studio выявляет не только опечатки, мёртвый код и другие ошибки, но и потенциальные уязвимости (Static Application Security Testing, SAST). Анализатор отображает предупреждения на Common Weakness Enumeration, SEI CERT Coding Standards, а также поддерживает стандарт MISRA.

3. Добавить в построенном пайплайне TeamCity шаги, на которых код проверяется статическими анализаторами и, в случае обнаружения проблем максимальной критичности для выбранного анализатора (для анализатора безопасности - CRITICAL, для анализатора качества кода - BLOCKER, для средства поиска секретов - любой обнаруженный пароль), билд завершается с ошибкой.

Добавим шаг, в котором будет происходить процесс анализа серверного кода:

Рассмотрим шаг «Analize and ConvertAnalysisResults»:

Разберем подробнее custom script:

#Установим apt-utils и обновим индекс пакетов:

apt install apt-utils -y

apt-get update -y

#Установим Qt и компиляторы

apt-get install qt5-qmake -y

apt-get install cmake -y

apt-get install qt5-default -y

apt-get install qtbase5-dev -y

apt-get install build-essential -y

apt-get install wget -y

#Установим непосредственно pvs-studio

wget -q -O - https://cdn.pvs-studio.com/etc/pubkey.txt | apt-key add - -y

wget -O /etc/apt/sources.list.d/viva64.list https://cdn.pvs-studio.com/etc/viva64.list

apt-get update -y

apt-get install pvs-studio -y

#Добавим лицензию pvs-studio

pvs-studio-analyzer credentials 817601@mail.ru 5DMG-77B4-25N1-2WGF

#Перейдем в папку сервера:

cd ./untitled9

#Удалим старые сборки и старые версии файлов:

rm -f compile_commands.json

rm -f moc*

rm -f Makefile

rm -f *.o

rm -f untitled9.pro

rm -f strace_out

rm -f pvs.log

rm -f pvs*.tasks

#Проведем трассировку компиляции (соберем проект):

qmake -o Makefile untitled.pro

make clean

pvs-studio-analyzer trace -- make

rm -f moc*

#Удалим старую директорию для сохранения отчета и создадим новую:

rm -R ./report

mkdir ./report

#Запустим анализ нашего сервера:

pvs-studio-analyzer analyze -l /root/.config/PVS-Studio/PVS-Studio.lic -o pvs.log –disableLicenseExpirationCheck

#Сохраним отчет в формате .html в созданной папке:

plog-converter -a GA:1,2 --renderTypes fullhtml -t tasklist -o ./report/ pvs.log

#Заменим «фальшивые» err просто на 1, чтобы корректно найти критические ошибки:

sed "s|pvs-studio.com/en/docs/warnings/   1   err   Help: The documentation for all analyzer warnings is available here: https://pvs-studio.com/en/docs/warnings/.|1|" ./report/pvs.tasks > ./report/pvs1.tasks

sed "s|err   Renew|1|" ./report/pvs1.tasks > ./report/pvs2.tasks

#Найдем err в отчете (если err будет найден, то код – 0, иначе – 1):

grep -r 'err' ./report/pvs2.tasks

#Выведем в случае обнаружения err в отчете код 1 (если ошибок нет, то 0):
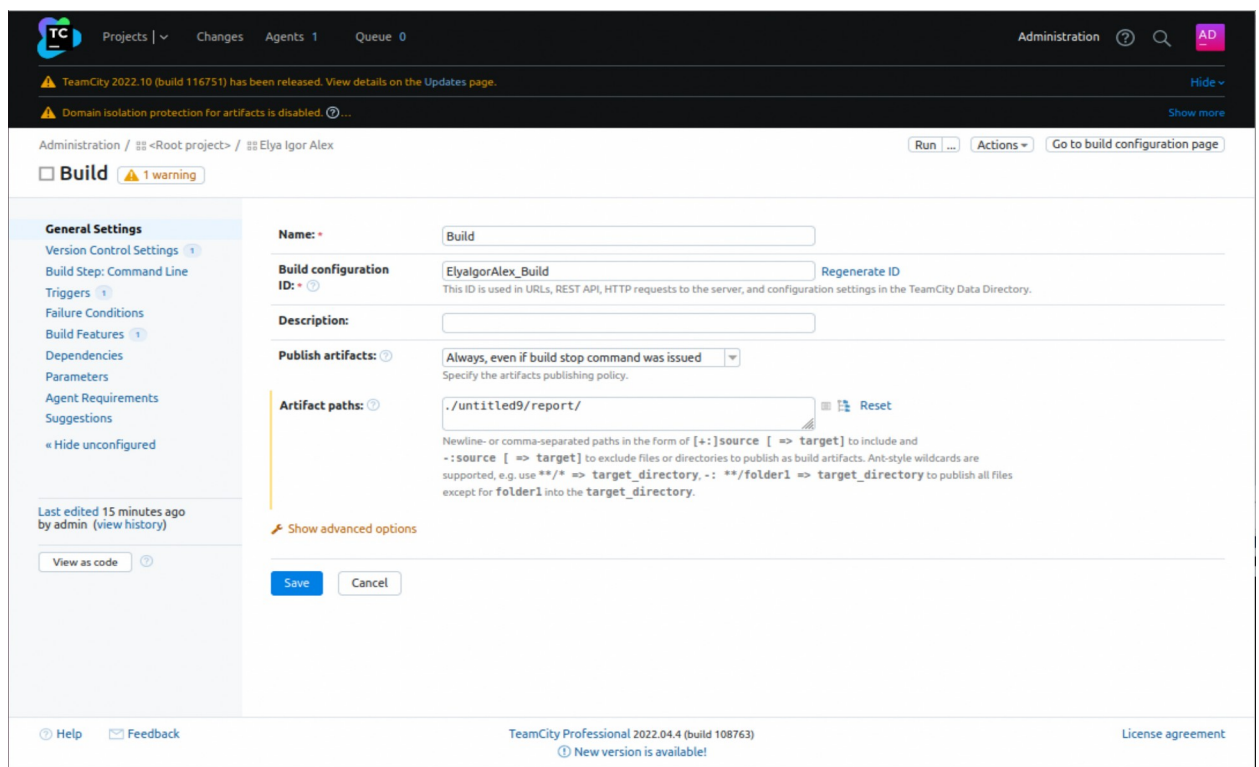
if [$? -eq 0]

      then

            exit 1

      else

            exit 0

fi

После создания шага настроим пути Arthifact, то есть пропишем в них путь созданной папки report, чтобы после успешного выполнения сборки отчёт анализатора будет доступен в артефактах:

Также можно сделать так, чтобы отчёт анализатора отображался на специальной вкладке отчёта о сеансе сборки. Для этого необходимо перейти в настройки проекта, открыть 'Report Tabs' и добавить новую вкладку отчёта сборки ('Create new build report tab').
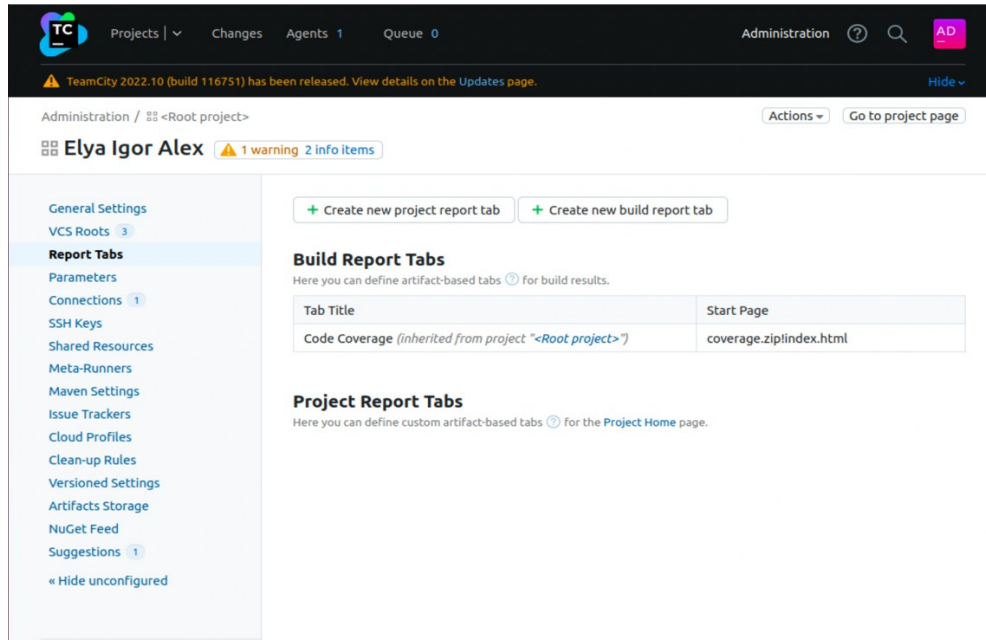
В окне добавления вкладки в поле 'Start page' необходимо указать путь к файлу 'index.html' относительно корневой папки артефактов:

Зафиксируем успешную сборку из feature1:

Посмотрим созданные отчеты с анализом кода сервера:

Теперь зафиксируем сборку с ошибками из ветки feature4:

⚠ TeamCity 2022.10 (build 116751) has been released. View details on the Updates page.   Hide ⌄

⚠ Domain isolation protection for artifacts is disabled. ⑦…   Show more

⊞ Elya Igor Alex   👁   Edit Project Settings

Overview   Change Log   Statistics   Current Problems   Investigations   Muted Problems   Flaky Tests 0   Pvs-Studio-REPORT

Data was taken from Elya Igor Alex / Build, build 164 | ▽ started at 17 Nov 22 17:47

PVS-Studio Analysis Results

Date:	Thu Nov 17 17:48:37 2022
PVS-Studio Version:	7.21.64848.262
Command Line:	plog-converter -a GA\:1,2 --renderTypes fullhtml -t tasklist -o ./report/ pvs.log
Total Warnings (GA):	3
Fails/Info:	1

Group ▲	⬍	Location	⬍	Level	⬍	Code	⬍	Message
		:1				Renew		Your license will expire in 6 days. Click 'Renew' to learn more or contact us at support@viva64.com.
General Analysis		functions.cpp:21		Medium		V640		The code's operational logic does not correspond with its formatting. The statement is indented to the right, but it is always executed. It is possible that curly brackets are missing.
General Analysis		mytcpserver.cpp:25		High		V611		The memory was allocated using 'new T' operator but was released using the 'delete []' operator. Consider inspecting this code. It's probably better to use 'delete server;'.
General Analysis		mytcpserver.cpp:24		Medium		V799		The 'server' variable is not used after memory has been allocated for it. Consider checking the use of this variable.