Министерство науки и высшего образования РФ

ФГАОУ ВО «Омский государственный технический университет»

Кафедра «Автоматизированные системы обработки информации и управления»

**ОТЧЕТ**

по лабораторной работе № 3

на тему «Fuzzing.»

по дисциплине «Качества и надёжность программных систем»

студента Рухлова Максима Валерьевича группы ПИН-192

**Пояснительная записка**

Шифр работы: От-30028999-43-ПИН-36-19 ПЗ

Направление 09.03.04

Старший преподаватель Д.А. Панков

Студент М.В. Рухлов

Омск 2022

На рисунках 1-7 показан ход выполнения лабораторной работы.

На рисунке 1 показан ввод команды docker pull aflplusplus/aflplusplus. docker pull - извлечение образа или репозитория из локального реестра или Docker Hub.

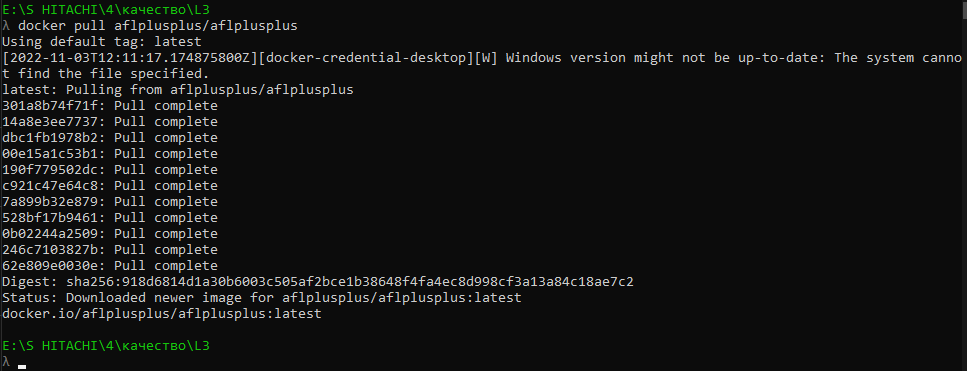


Рисунок 1 - docker pull aflplusplus/aflplusplus

На рисунке 2 показано клонирование репозитория.

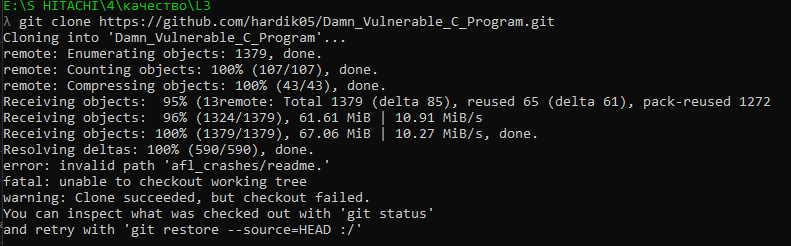


Рисунок 2 - git clone <https://github.com/hardik05/Damn_Vulnerable_C_Program.git>

На рисунке 3 показано содержание репозитория.

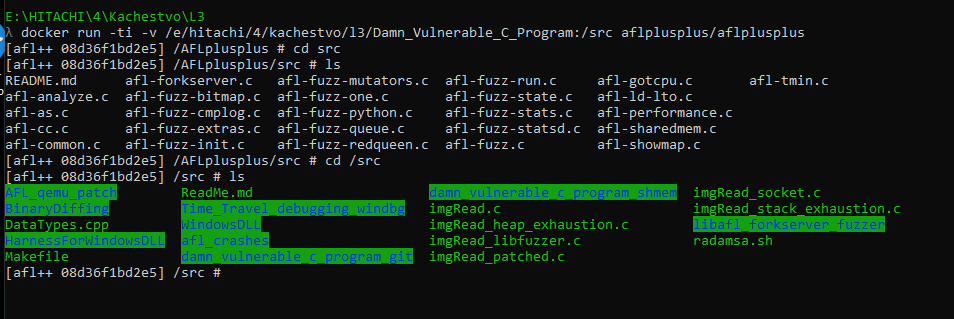


Рисунок 3 – Просмотр директории

На рисунке 4 показана сборка программы для чтения уязвимостей.

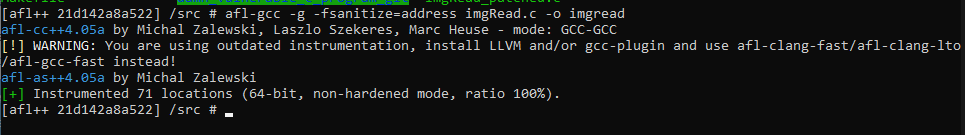


Рисунок 4 – Сборка программы

На рисунке 5 показано создание новых папок

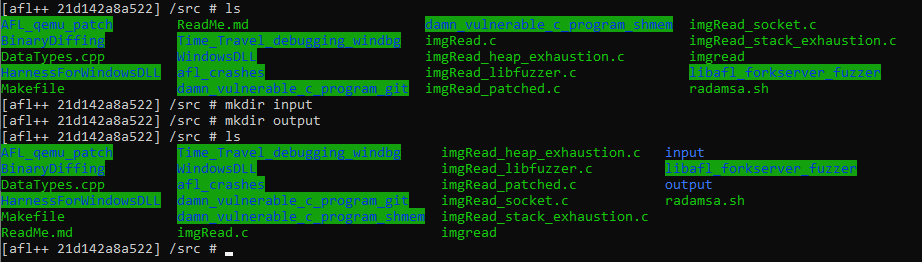


Рисунок 5 - Добавление директории input и output

На рисунке 6 можно увидеть файл текстовый который был создан с помощью echo "IMG" >input/1.txt



Рисунок 6 – 1.txt

На рисунке 7 показан фаззинг.

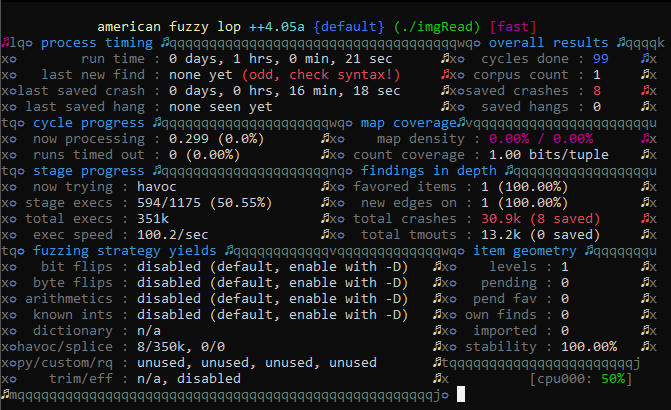


Рисунок 7 – Фаззинг

В ходе фаззинга было обнаружено 8 уязвимостей.

**Уязвимость первая:**

На рисунке 8 показано описание первой уязвимости.

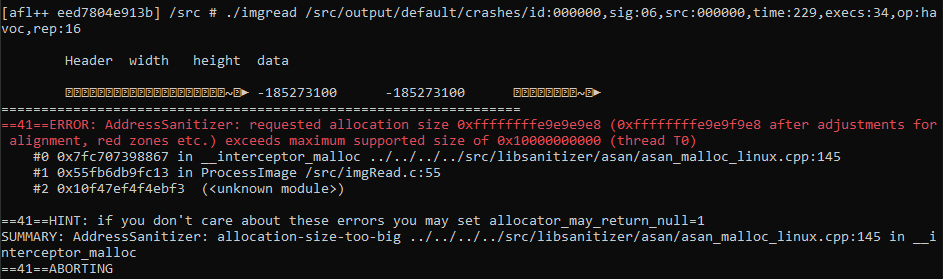


Рисунок 8 – Первая уязвимость

На рисунке 8 показано, что уязвимость заключается в превышении допустимого размера памяти.

**Уязвимость вторая:**

По аналогии с первой.

**Уязвимость третья:**

На рисунке 9 показано описание третьей уязвимости.

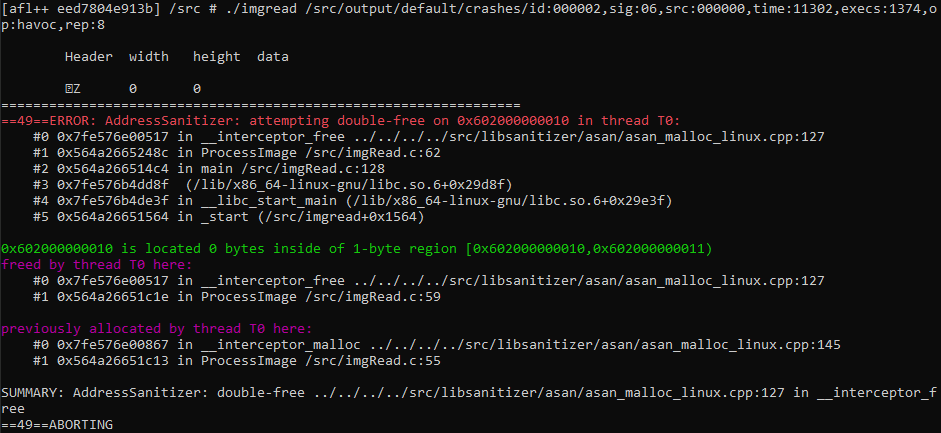


Рисунок 9 – Третья уязвимость

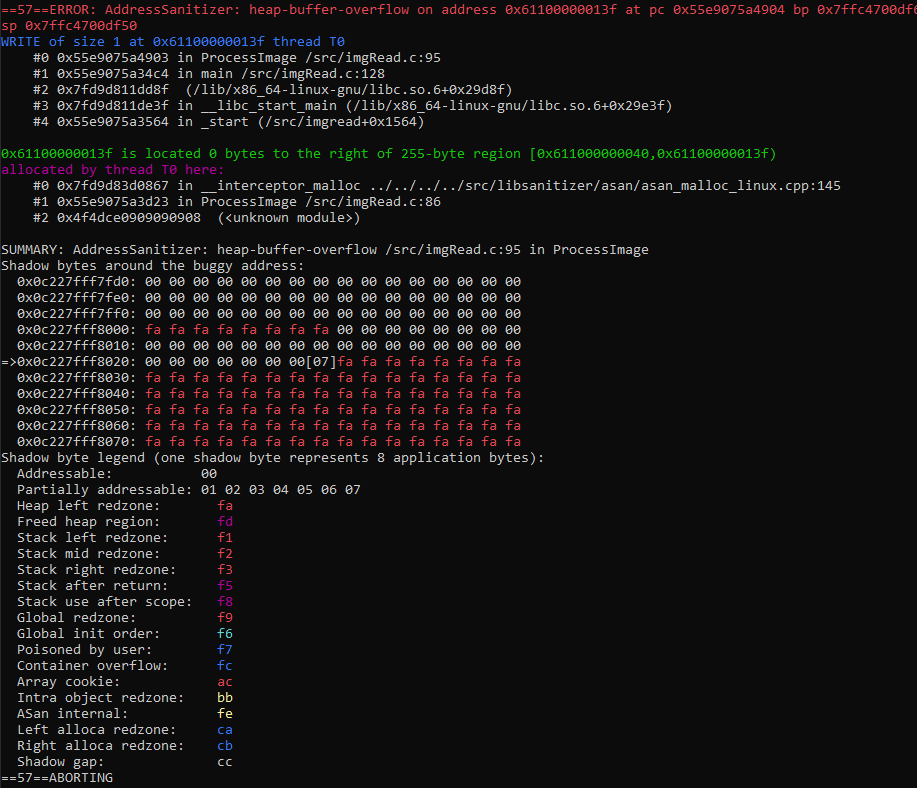
На рисунке 9 показано, что уязвимость заключается в двойном очищении, то есть освобождение памяти от переменной, от которой память уже была освобождена.

**Уязвимость четвёртая:**

По аналогии с первой.

**Уязвимость пятая:**

На рисунке 10 показано описание пятой уязвимости.

Рисунок 10 – Пятая уязвимость

На рисунке 10 показано, что уязвимость заключается в том, что по адресу 0x61100000013f был переполнен heap буфер.

**Уязвимость шестая:**

На рисунке 11 показано описание шестой уязвимости.

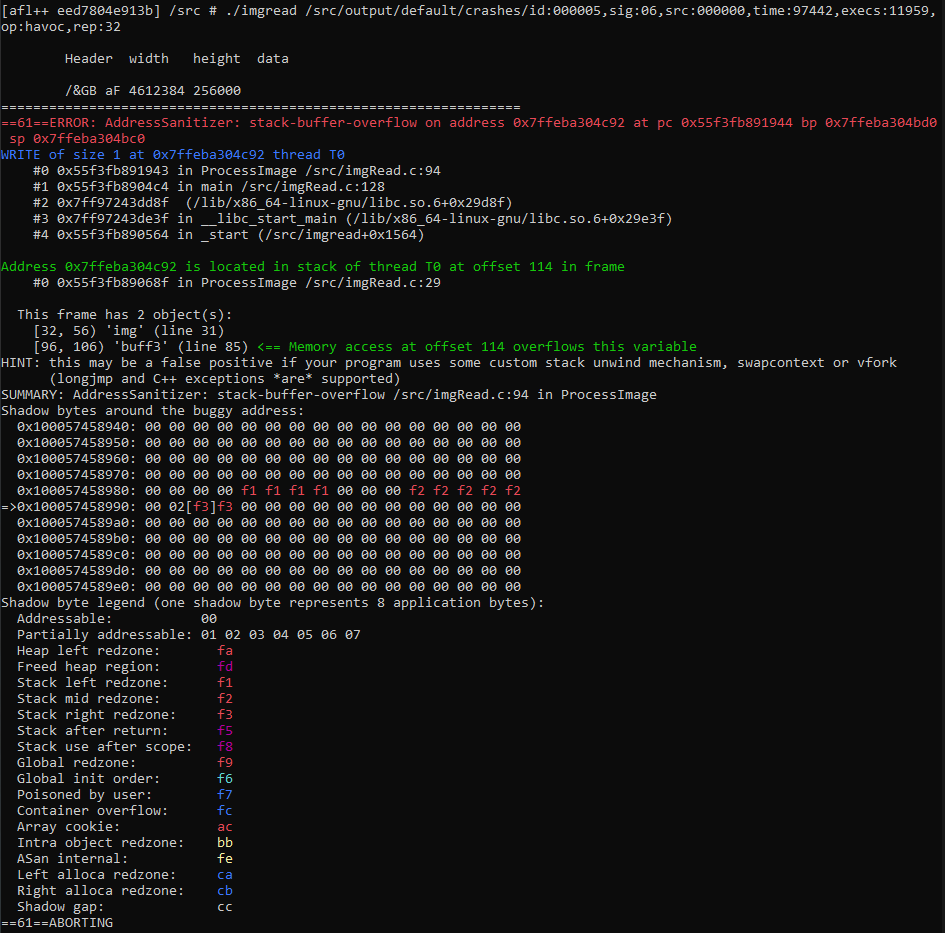


Рисунок 11 – Шестая уязвимость

На рисунке 11 показано, что уязвимость заключается в том, что по адресу 0x7ffeba304c92 был переполнен stack буфер.

**Уязвимость седьмая:**

На рисунке 12 показано описание седьмая уязвимости.

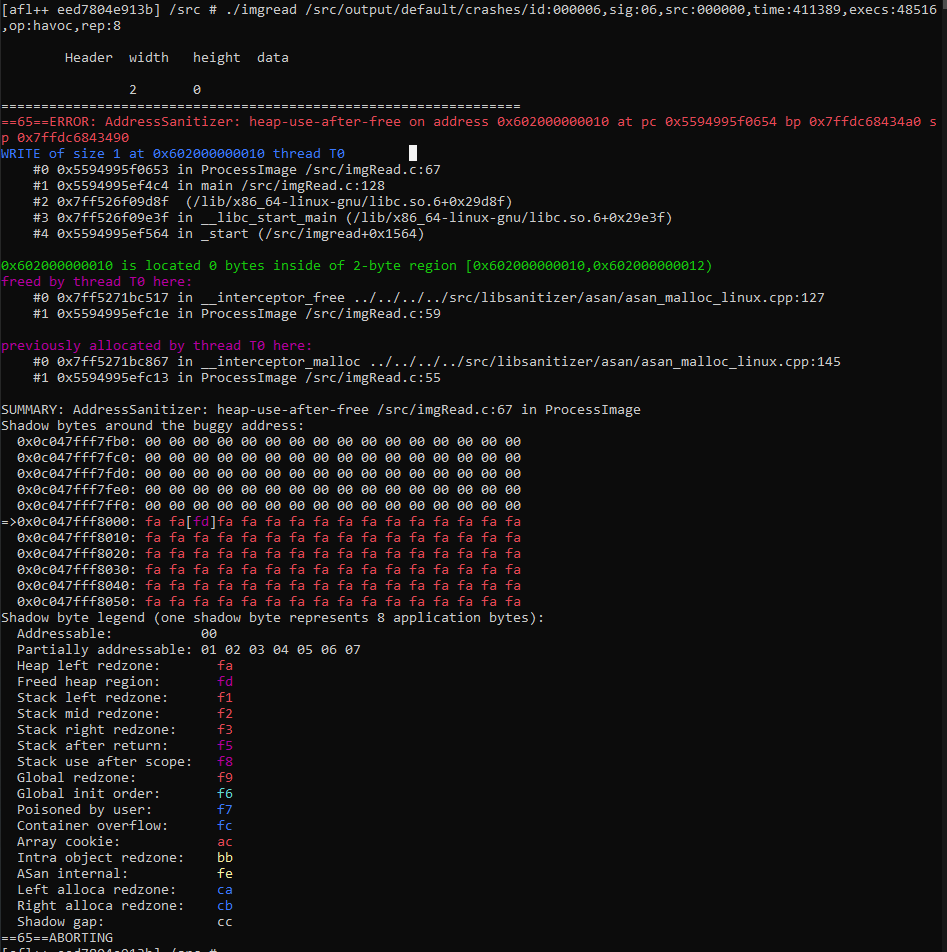


Рисунок 12 – Седьмая уязвимость

На рисунке 12 показано, что уязвимость заключается в том, что по адресу 0x602000000010 была попытка использовать очищенный heap буфер.

**Уязвимость восьмая:**

По аналогии с первой.

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторной работы был произведён фаззинг, обнаружены и описаны уязвимости.