Вариант 4

Задание 1

Денис Сутягин

Исходные данные

Проект https://github.com/jweyrich/imgify, ветка master

Отчет

1. Порядок сборки релизной и отладочной версии

Сборка производится с использованием CI Jenkins, развернутой на ВМ Исходный код Jenkinsfile размещается проекте github: https://github.com/drJabber/ispras-fuzz.git, в ветке imgify-build/dev01- для сборки отладочной версии, imgify-build/rel01 - для сборки релизной версии

Сборка происходит в образе doker - aflplusplus/aflpusplus:stable, в который в процессе сборки добавляется установка зависимостей - libpng-dev

```
FROM aflplusplus/aflplusplus:stable

RUN cat /etc/os-release && \
    apt update && \
    apt install -y libpng-dev
```

Далее - данный докерфайл используется в дженкинсе для сборки образа и последующей сборки imgify:

```
    stage("build") {
         agent {
             // dockerfile true
             dockerfile {
               filename 'Dockerfile.dev01'
             }
         }
         steps {
.....
         }
     }
```

Далее - происходит загрузка кода imgify из github:

```
          checkout([
              $class: 'GitSCM',
              branches: [[name: 'master']],
              extensions: [[$class: 'CloneOption', shallow: false, depth:
0, reference: '']],
              userRemoteConfigs: [[credentialsId:  'gh-ci', url:
"https://github.com/jweyrich/imgify.git"]],
          ])
```

При попытке собрать релизную версию командой make -j8 возникают ошибки компиляции, т.к. определения констант предроцессора PROGRAM, VERSION и т.д. вводятся после начала их использования, в связи с этим сделал в Jenkinsfile патч - перенес #include "common_options.h" - после включения версии и определения константы PROGRAM в модулях bin2png и png2bin

```
          sh """
              echo "patch defines"
              temp_file_name="\$(mktemp /tmp/foo.XXXXXXXXX)" && \
                 cat ./png2bin.c | \
                 awk -v replacement="" 'NR==30{\$0=replacement}{print}' |
                 awk -v replacement='#include \"common_options.h\"'
'NR==34{\$0=replacement}{print}' > \$temp_file_name && \
                 mv -f \$temp_file_name ./png2bin.c

              echo "patch defines"
              temp_file_name="\$(mktemp /tmp/foo.XXXXXXXXX)" && \
                 cat ./bin2png.c | \
                 awk -v replacement="" 'NR==30{\$0=replacement}{print}' |
                 awk -v replacement='#include \"common_options.h\"'
'NR==34{\$0=replacement}{print}' > \$temp_file_name && \
                 mv -f \$temp_file_name ./bin2png.c
            """
```

После патча - сборка релизной версии imgify проходит успешно

Сборка отладочной версии происходит аналогично командой make -j8 CFLAGS="-g -Wall"

2. Порядок сборки с датчиками ошибок

Сборка инструментированной версии происходит аналогично сборке отладочной версии, при этом добавляются опции компилятора:

```
              make -j8 CFLAGS="-g -DFORTIFY_SOURCE=2 -Wall -
fsanitize=address -fsanitize=pointer-compare -fsanitize=pointer-subtract -
fsanitize=leak \
                          -fsanitize-address-use-after-scope -
fsanitize=unreachable -fsanitize=undefined -fcf-protection=full \
                          -fstack-check -fstack-protector-all --coverage"
```

3. Особенности стенда

Добавил в пайплайн сборки вызов собранных утилит bin2png и png2bin с тестовым файлом png. При запуске:

```
./bin2png -i ./out.bin -o ./out.png -p 253
```

сработал датчик "double free"

```
+ ./bin2png -i ./out.bin -o ./out.png
=================================================================
==43==ERROR: AddressSanitizer: attempting double-free on 0x61e000000080 in
thread T0:
    #0 0x5629b5710112 in free (/var/lib/jenkins/workspace/tcpdump_imgify-
build_dev01@2/bin2png+0xa4112) (BuildId:
1bfae3096d841a443ffb13120401d5bca2379679)
    #1 0x5629b574f836 in png_save
/var/lib/jenkins/workspace/tcpdump_imgify-build_dev01@2/imgify.c:257:2
    #2 0x5629b574c674 in do_work /var/lib/jenkins/workspace/tcpdump_imgify-
build_dev01@2/bin2png.c:43:12
    #3 0x5629b574b51e in main /var/lib/jenkins/workspace/tcpdump_imgify-
build_dev01@2/./common_main.h:36:2
    #4 0x7fc6d1471d8f  (/lib/x86_64-linux-gnu/libc.so.6+0x29d8f) (BuildId:
229b7dc509053fe4df5e29e8629911f0c3bc66dd)
    #5 0x7fc6d1471e3f in __libc_start_main (/lib/x86_64-linux-
gnu/libc.so.6+0x29e3f) (BuildId: 229b7dc509053fe4df5e29e8629911f0c3bc66dd)
    #6 0x5629b568d574 in _start (/var/lib/jenkins/workspace/tcpdump_imgify-
build_dev01@2/bin2png+0x21574) (BuildId:
1bfae3096d841a443ffb13120401d5bca2379679)

0x61e000000080 is located 0 bytes inside of 2464-byte region
[0x61e000000080,0x61e000000a20)
freed by thread T0 here:
    #0 0x5629b5710112 in free (/var/lib/jenkins/workspace/tcpdump_imgify-
build_dev01@2/bin2png+0xa4112) (BuildId:
1bfae3096d841a443ffb13120401d5bca2379679)
    #1 0x5629b574f801 in png_save
/var/lib/jenkins/workspace/tcpdump_imgify-build_dev01@2/imgify.c:253:4
    #2 0x5629b574c674 in do_work /var/lib/jenkins/workspace/tcpdump_imgify-
build_dev01@2/bin2png.c:43:12
    #3 0x5629b574b51e in main /var/lib/jenkins/workspace/tcpdump_imgify-
build_dev01@2/./common_main.h:36:2
    #4 0x7fc6d1471d8f  (/lib/x86_64-linux-gnu/libc.so.6+0x29d8f) (BuildId:
229b7dc509053fe4df5e29e8629911f0c3bc66dd)

previously allocated by thread T0 here:
    #0 0x5629b57103be in __interceptor_malloc
(/var/lib/jenkins/workspace/tcpdump_imgify-build_dev01@2/bin2png+0xa43be)
(BuildId: 1bfae3096d841a443ffb13120401d5bca2379679)
    #1 0x5629b574f44b in png_save
```

```
/var/lib/jenkins/workspace/tcpdump_imgify-build_dev01@2/imgify.c:231:28
    #2 0x5629b574c674 in do_work /var/lib/jenkins/workspace/tcpdump_imgify-
build_dev01@2/bin2png.c:43:12
    #3 0x5629b574b51e in main /var/lib/jenkins/workspace/tcpdump_imgify-
build_dev01@2/./common_main.h:36:2
    #4 0x7fc6d1471d8f  (/lib/x86_64-linux-gnu/libc.so.6+0x29d8f) (BuildId:
229b7dc509053fe4df5e29e8629911f0c3bc66dd)

SUMMARY: AddressSanitizer: double-free
(/var/lib/jenkins/workspace/tcpdump_imgify-build_dev01@2/bin2png+0xa4112)
(BuildId: 1bfae3096d841a443ffb13120401d5bca2379679) in free
==43==ABORTING
```

пропатчил код:

```
            echo "fix double free in imgify.c 253"
            temp_file_name="\$(mktemp /tmp/foo.XXXXXXXXX)" && \
               cat ./imgify.c | \
               awk -v replacement="" 'NR==253{\$0=replacement}{print}'  >
\$temp_file_name && \
               mv -f \$temp_file_name ./imgify.c
```

С патчем - код собирается успешно и тест отрабатывает без срабатывания датчиков.

4. Установка Jenkins, создание заданий на сборку

- создал ВМ ubuntu 22.04
- установил Jenkins

```
sudo apt-get install jenkins
```

- установил jdk

```
sudo apt install default-jre
```

- установил docker-ce, добавил текущему пользователю и дженкинсу права на докер

```
echo "deb [arch=$(dpkg --print-architecture) signed-
by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo
tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt install docker-ce=5:24.0.6-1~ubuntu.22.04~jammy
```

```
sudo usermod -aG docker jenkins
sudo usermod -aG docker ${USER}
```

- создал на гитхабе PAT - для возможности скачивания дженкинсом кода с гитхаба
- прописал PAT в разделе credentials конфигурации дженкинса с идентификатором "gh-ci"
- создал в дженкинсе multibranch pipeline "ipras-fuzz", прописал там как источник веток репозиторий на гитхабе https://github.com/drJabber/ispras-fuzz.git
- ветки проекта https://github.com/drJabber/ispras-fuzz.git соержат Jenkinsfile и Dockerfile для соответствующей версии собираемого кода
- Jenkins вытаскивает из гитхаба все ветки, которые имеются в проекте и исполняет файлы Jenkinsfile, которые находятся в этом проекте
- проект https://github.com/drJabber/ispras-fuzz.git также содержит Dockerfile для сборки образа с afl++ и зависимостями собираемого проекта (imgify)
- в процессе исполнения Jenkinfile дженкинс собирает образ с необходимыми зависимостями и выполняет внутри образа загрузку кода собираемого проекта imgify и собирает его с соответствующими опциями компилятора в зависимости от версии (релизная, отладочная инструментированная)
- итого, чтобы добавить еще одно задание на сборку - необходимо создать в проекте ispras-fuzz по одной ветке для каждой версии нового ПО, изменить Dockerfile, чтобы добавить зависимости нового проекта, изменить Jenkinsfile, чтобы прописать там url нового проекта, добавить необходимые патчи и опции компилятора

PS

Dockerfile - для инструментированной версии:

```
FROM aflplusplus/aflplusplus:stable

RUN cat /etc/os-release && \
    apt update && \
    apt install -y libpng-dev
```

Jenkinsfile - для инструментированной версии

```
pipeline {
  agent any
  stages {
    stage("build") {
        agent {
            // dockerfile true
            dockerfile {
              filename 'Dockerfile.dev01'
            }
        }
        steps {
          checkout([
              $class: 'GitSCM',
```

```
              branches: [[name: 'master']],
              extensions: [[$class: 'CloneOption', shallow: false, depth:
0, reference: '']],
              userRemoteConfigs: [[credentialsId:  'gh-ci', url:
"https://github.com/jweyrich/imgify.git"]],
          ])

          sh """
             echo "patch defines"
             temp_file_name="\$(mktemp /tmp/foo.XXXXXXXXX)" && \
                cat ./png2bin.c | \
                awk -v replacement="" 'NR==30{\$0=replacement}{print}' |
                awk -v replacement='#include \"common_options.h\"'
'NR==34{\$0=replacement}{print}' > \$temp_file_name && \
                mv -f \$temp_file_name ./png2bin.c

             echo "patch defines"
             temp_file_name="\$(mktemp /tmp/foo.XXXXXXXXX)" && \
                cat ./bin2png.c | \
                awk -v replacement="" 'NR==30{\$0=replacement}{print}' |
                awk -v replacement='#include \"common_options.h\"'
'NR==34{\$0=replacement}{print}' > \$temp_file_name && \
                mv -f \$temp_file_name ./bin2png.c

             echo "fix double free in imgify.c 253"
             temp_file_name="\$(mktemp /tmp/foo.XXXXXXXXX)" && \
                cat ./imgify.c | \
                awk -v replacement="" 'NR==253{\$0=replacement}{print}'  >
\$temp_file_name && \
                mv -f \$temp_file_name ./imgify.c


             make -j8 CFLAGS="-g -DFORTIFY_SOURCE=2 -Wall -
fsanitize=address -fsanitize=pointer-compare -fsanitize=pointer-subtract -
fsanitize=leak \
                          -fsanitize-address-use-after-scope -
fsanitize=unreachable -fsanitize=undefined -fcf-protection=full \
                          -fstack-check -fstack-protector-all --coverage"

             ./png2bin -i ./screenshot.png -o ./out.bin -p 254

             ./bin2png -i ./out.bin -o ./out.png -p 253

          """

          archiveArtifacts artifacts: '**/bin2png, **/png2bin'
        }
    }
  }
}
```