

Project 3

Security In Software Applications

Giovanni Pica
1816394

1 Introduction

This project is based on **Fuzz Testing** and the tool used is **American Fuzzy Lop** that is a security-oriented fuzzer that employs a novel type of compile-time instrumentation and genetic algorithms to automatically discover clean, interesting test cases that trigger new internal states in the targeted binary. The software that is tested is **ImageMagick** that is an image manipulation software, in particular in this project I decide to test the 6.7.0-10 version. I decide also to use **ASAN** which is a memory error detector for C/C++.

2 Setup

At the beginning I decide to develop this project on my **Macbook Air M1**, but I notice that AFL doesn't support this CPU and when I tried to compile the makefile it gives to me many errors, so I decide to use my old **Lenovo** with a VM based on **Linux Mint**. The steps to correctly install AFL were:

- get AFL from github with
`git clone https://github.com/google/AFL.git`
- make the MakeFile of AFL and install it

Now the AFL is installed and afl-clang or afl-gcc (afl-clang++ and afl-g++ for C++ programs) can be used for the fuzz testing of any kind of software. To perform AFL to ImageMagick the steps were:

- get ImageMagick from <https://sourceforge.net/projects/imagemagick/files/old-sources/>
- change the version of the ImageMagick with
- set the `CC=afl-gcc`, `CXX=afl-g++`, `CFLAGS="-fsanitize=address-g"` and also for `CXXFLAGS` for ASAN and execute
`./configure --prefix=/path/to/ImageMagick`

- make and install the file created by the instruction above and setting AFL_USE_ASAN=1
- create the afl_in and afl_out directory for input files and output
- finally perform fuzz testing with -d for quick and dirty mode (skips deterministic steps) and -m none for ASAN
`afl-fuzz -d -m none -i afl_in -o afl_out -- ./bin/convert`
`@@ /dev/null`

3 Results

Number of files used is **1652** and I take the downloadable archive from <https://lcamtuf.coredump.cx/afl/demo/> and I decide to use **PNG** format. Results are summarized in the figure 1:

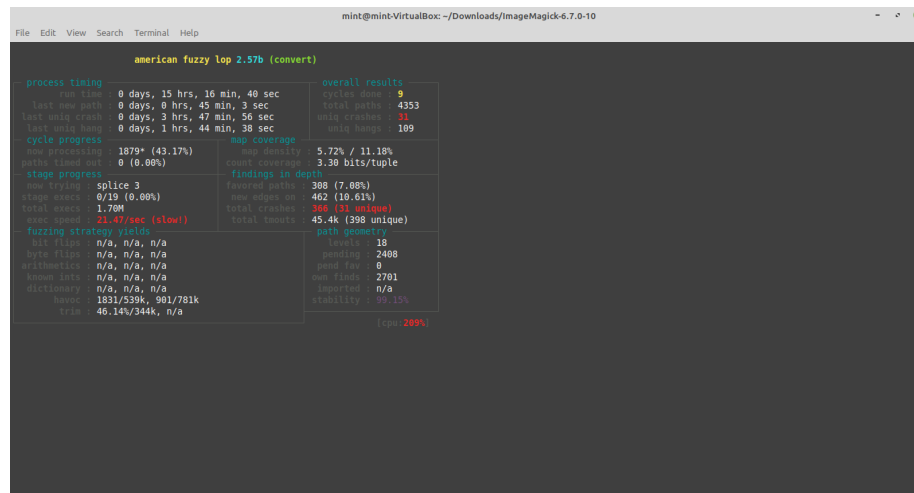


Figure 1: Results of AFL

3.1 Flaws Found are known CVEs?

For flaws research I ran "convert" function of ImageMagick of the files that were in the folder "crashes" (that are 31) of "afl_out" folder and I discovered that all of those crashes are caused by Heap-Buffer-Overflow and in this site there are the CVEs based on this flaw <https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=heap+buffer+overflow>. In the figure 2 I show how the ASAN works to find those flaws. I also handled the "hangs" and I discovered that some of those timeouts were caused by memory leaks that the LeakSanitizer found or unexpected end-of-file or invalid pixels or pixel cache allocation failed and so on. In figure 3 I give an example of how LeakSanitizer works.

```

mint@mint-VirtualBox: ~/Downloads/ImageMagick-6.7.0-10/afl_out/hangs
File Edit View Search Terminal Help
id:000000,sig:06,src:002042,op:havoc,rep:16 id:000019,sig:06,src:002102,op:havoc,rep:0 id:000039,sig:00,src:004103,op:havoc,rep:2
id:000009,sig:06,src:002083,op:havoc,rep:16 id:000020,sig:06,src:002102,op:havoc,rep:2 README.txt
id:000010,sig:06,src:002042,op:havoc,rep:16 id:000021,sig:06,src:002181+001638,op:splice,rep:2
mint@mint-VirtualBox:~/Downloads/ImageMagick-6.7.0-10/afl_out/crashes$ /home/mint/Downloads/ImageMagick-6.7.0-10/bin/convert id:000026,sig:06,src:002317+002044,op:splice,rep:0 /dev/null
=====
==2070904==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x61f000000c07 at pc 0x77fe0b13d700 bp 0x77ffac48ef30 sp 0x77ffac48ef20
READ of size 1 at 0x61f000000c07 thread 10
#0 0x77fe0b13d700 in ReadPCXImage /home/mint/Downloads/ImageMagick-6.7.0-10/coders/pcx.c:582:23
#1 0x77fe0b78cdd9 in ReadImage /home/mint/Downloads/ImageMagick-6.7.0-10/magick/constitute.c:523:13
#2 0x77fe0b78dbcb in ReadImages /home/mint/Downloads/ImageMagick-6.7.0-10/magick/constitute.c:884:10
#3 0x77fe07238f23 in ConvertImageCommand /home/mint/Downloads/ImageMagick-6.7.0-10/wand/convert.c:593:18
#4 0x77fe073686f7 in MagickCommandGenesis /home/mint/Downloads/ImageMagick-6.7.0-10/wand/mogrify.c:169:16
#5 0x4c6298 in main /home/mint/Downloads/ImageMagick-6.7.0-10/utilities/convert.c:80:10
#6 0x77fe0b30802 in libc_start_main /build/glibc-2.31/csu/../csu/libc-start.c:308:16
#7 0x41e39d in start (/home/mint/Downloads/ImageMagick-6.7.0-10/bin/convert+0x41e39d)

0x61f000000c07 is located 0 bytes to the right of 3047-byte region [0x61f000000080,0x61f000000c67)
allocated by thread 10 here:
#0 0x496add in malloc (/home/mint/Downloads/ImageMagick-6.7.0-10/bin/convert+0x496add)
#1 0x77fe0b1368c1 in ReadPCXImage /home/mint/Downloads/ImageMagick-6.7.0-10/coders/pcx.c:389:34
#2 0x77fe0b78cdd9 in ReadImage /home/mint/Downloads/ImageMagick-6.7.0-10/magick/constitute.c:523:13
#3 0x77fe0b78dbcb in ReadImages /home/mint/Downloads/ImageMagick-6.7.0-10/magick/constitute.c:884:10
#4 0x77fe07238f23 in ConvertImageCommand /home/mint/Downloads/ImageMagick-6.7.0-10/wand/convert.c:593:18
#5 0x77fe073686f7 in MagickCommandGenesis /home/mint/Downloads/ImageMagick-6.7.0-10/wand/mogrify.c:169:16
#6 0x4c6298 in main /home/mint/Downloads/ImageMagick-6.7.0-10/utilities/convert.c:80:10

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/mint/Downloads/ImageMagick-6.7.0-10/coders/pcx.c:582:23 in ReadPCXImage
Shadow bytes around the buggy address:
 0xc3e7fff8130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0xc3e7fff8140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0xc3e7fff8150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0xc3e7fff8160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0xc3e7fff8170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0xc3e7fff8180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Figure 2: ASAN heap buffer overflow

```

mint@mint-VirtualBox: ~/Downloads/ImageMagick-6.7.0-10/afl_out/hangs
File Edit View Search Terminal Help
mint@mint-VirtualBox:~/Downloads/ImageMagick-6.7.0-10/afl_out/hangs$ /home/mint/Downloads/ImageMagick-6.7.0-10/bin/convert id:000103,src:004160,op:havoc,rep:32 /dev/null
convert: unexpected end-of-file 'id:000103,src:004160,op:havoc,rep:32': @ error/sgi.c/ReadSGIImage/675.
mint@mint-VirtualBox:~/Downloads/ImageMagick-6.7.0-10/afl_out/hangs$ /home/mint/Downloads/ImageMagick-6.7.0-10/bin/convert id:000088,src:003373+001573,op:splice,rep:32 /dev/null
convert: unexpected end-of-file 'id:000088,src:003373+001573,op:splice,rep:32': @ error/pnm.c/ReadPNMImage/1253.

=====
==2071003==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 4096 byte(s) in 1 object(s) allocated from:
#0 0x496add in malloc (/home/mint/Downloads/ImageMagick-6.7.0-10/bin/convert+0x496add)
#1 0x77f49c6b0245b in AcquireString /home/mint/Downloads/ImageMagick-6.7.0-10/magick/string.c:124:24
#2 0x77f49c6b0245b in PNMInteger /home/mint/Downloads/ImageMagick-6.7.0-10/coders/pnm.c:181:19

SUMMARY: AddressSanitizer: 4096 byte(s) leaked in 1 allocation(s).
mint@mint-VirtualBox:~/Downloads/ImageMagick-6.7.0-10/afl_out/hangs$ /home/mint/Downloads/ImageMagick-6.7.0-10/bin/convert id:000088,src:003373+001573,op:splice,rep:32 /dev/null
convert: unexpected end-of-file 'id:000088,src:003373+001573,op:splice,rep:32': @ error/pnm.c/ReadPNMImage/1253.

=====
==2071006==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 4096 byte(s) in 1 object(s) allocated from:
#0 0x496add in malloc (/home/mint/Downloads/ImageMagick-6.7.0-10/bin/convert+0x496add)
#1 0x77f4c0b06c45b in AcquireString /home/mint/Downloads/ImageMagick-6.7.0-10/magick/string.c:124:24
#2 0x77f4c0b06c45b in PNMInteger /home/mint/Downloads/ImageMagick-6.7.0-10/coders/pnm.c:181:19

SUMMARY: AddressSanitizer: 4096 byte(s) leaked in 1 allocation(s).
mint@mint-VirtualBox:~/Downloads/ImageMagick-6.7.0-10/afl_out/hangs$ /home/mint/Downloads/ImageMagick-6.7.0-10/bin/convert id:000046,src:002208+001309,op:splice,rep:128 /dev/null

```

Figure 3: Hangs memory leaks