# Fuzzing results

**Erik Poll**

**Digital Security group**

**Radboud University Nijmegen**

# Fuzzing results

| group | application | | format(s) |
|---|---|---|---|
| 3 | stb_image | old version | PNG, JPEG |
| 4 | FreeImage | old version | JPEG, GIF, WEB, IFF/LBM, TGA |
| 5 | OpenTTD | latest? | save files?? |
| 6 | flacon | latest | flac |
| 7 | PrusaSlicer | latest | config file |
| 8 | Mplayer | latest? | mp4, mcv |
| 9 | Ristretto, gdkpixbuf | latest | TIFF |
| 10 | picojpg | latest? | JPEG |
| 11 | libjxl | | JPEG XL |
| 13 | wavpack | old version | WAV, Wave64, CAF, DSDIFF, DSF,… |
| 14 | Radar2e | latest | binaries |
| 15 | p7zip | old version | 7z, zip,tar, gz,bz2 |
| 16 | VisIt | latest | Silo, PNG, ASCII, STL |
| 17 | FFmpeg | v2.4 | GIF, PNG |
| 18 | MP3 decompression | latest? | mp3 |
| 19 | echoprint codegen | latest, but 7 yrs old | mp3 |
| 20 | Audiowaveform | ? | mp3, wav, flac, ogg vobis, opus |
| 21 | gifdec | latest | GIF |
| 22 | cmus | | mp3, wav,… |
| 25 | PDFio | latest | PDF |

# Fuzzing results

| | application | afl(++) findings | dumb fuzzers? |
|---|---|---|---|
| 3 | **stb_image** | plenty of problems with & without ASan | Radamsa found one unique crash, an AssertionFailure |
| 4 | **FreeImage** | plenty of problems with & without ASan | ***Radamsa used as front-end*** |
| 5 | **OpenTTD** | plenty of problems, ***more hangs without ASan*** | zuff also found many crashes |
| 6 | **flacon** | problems found with & without ASan | zuff just as good, R**adamsa better!** |
| 7 | **PrusaSlicer** | problems found ***without ASan, not with ASan!*** | **Radamsa just as good?** |
| 8 | **Mplayer** | hangs found, with and without ASan, no crashes | Radamsa also found hangs |
| 9 | **Ristretto, gdkpixbuf** | hangs & crashes ***without ASan, not with ASan*** | Radamsa found nothing |
| 10 | **picojpg** | hangs & crashes, with & without ASan | zuff & Radamsa found problems, only with ASan |
| 11 | **libjxl** | plenty of crashes, with ASan | zuff & Radamsa found nothing? |
| 13 | **wavpack** | crashes and hangs, with & without Asan | zuff & Radamsa also found problems |
| 14 | **Radar2e** | afl found crashes, with & without ASan | Radamsa found crashes too, with and without ASan |
| 15 | **p7zip** | known CVE found, with & without ASan | |
| 16 | **VisIt** | afl++ found a hang, afl did not | 2 hangs with Radamsa |
| 17 | **FFmpeg** | crash found but it was error message | |
| 18 | **MP3 decompression** | crashes & hangs, with & without ASan | zzuf found no crashes, Radamsa did |
| 19 | **echoprint codegen** | | |
| 20 | **Audiowaveform** | lots of crashes, with & without ASan | zzuf found no crashes, Radamsa did |
| 21 | **gifdec** | plenty of problems, with & without ASan | zuff & Radamsa also found many problems |
| 22 | **cmus** | | |
| 25 | **PDFio** | new bugs found, ***faster without ASan!*** | Radamsa found similar (same?) problems too, but slowe |

# Fuzzing results

| | application | misc |
|---|---|---|
| 3 | stb_image | mutated file not displayable in Overleaf |
| 4 | FreeImage | known CVE found, but also another one? 33 different file formats supported! |
| 5 | OpenTTD | zuff finished in a few minutes, afl took hours?? |
| 6 | flacon | |
| 7 | PrusaSlicer | most problems found not security vulns, but one SEGF |
| 8 | Mplayer | |
| 9 | Ristretto, gdkpixbuf | used tiff dictionary; problematic mutations with huge sizes |
| 10 | picojpg | |
| 11 | libjxl | bugs found were not security-critical |
| 13 | wavpack | some flaws were known CVEs |
| 14 | Radar2e | (parts of) UBSan used too |
| 15 | p7zip | fuzzed the printer too? |
| 16 | VisIt | |
| 17 | FFmpeg | |
| 18 | MP3 decompression | |
| 19 | echoprint codegen | 1 path only… This program just calculates a hash? |
| 20 | Audiowaveform | |
| 21 | gifdec | some fixes |
| 22 | cmus | had to disable CRC chec |
| 25 | PDFio | Crashwalk used to analyse bugs |

# Recurring issues

- afl unique != unique
- afl zzzzz….
- Azure :-( ?

# group 3

Figure 7: id:000036,time:0,orig:id:004043,sync:jpeg9,src:005204.jpeg

Interestingly enough we had to create a screenshot of the image, since Overleaf (the program in which we write our report) displayed a white image which might indicate their image library also doesn't play well with this kind of input.

# Group 5 also fuzzed my printer / PDF viewer

It was quite difficult to get interesting results using Radamsa. This was mostly due to the fact that p7zip doesn'
0 if an error is handled by the program. That means that all kinds of errors, whether handled by the program or not, return with the same sta-

tive to attempt it. However, as so often with Microsoft, setting this up was very unintuitive. It almost seemed as if Microsoft'
"accidentally" sell you a subscription immediately instead of allowing access to the trial. By the time we finally managed to claim our free

raise positives overshadows the true positives found. As both also slow down the fuzzer, we would not recommend using them.

Lastly, we used CMPLOG and COMPCOV, which are used to improve the fuzzer' s ability to explore boolean structures. However, this provided no substantial benefits for our fuzzing target. Not only did it find less flaws compared to the fuzzer with no additional instrumentation, it also slowed down the generation of mu-

# group 14 - Radare

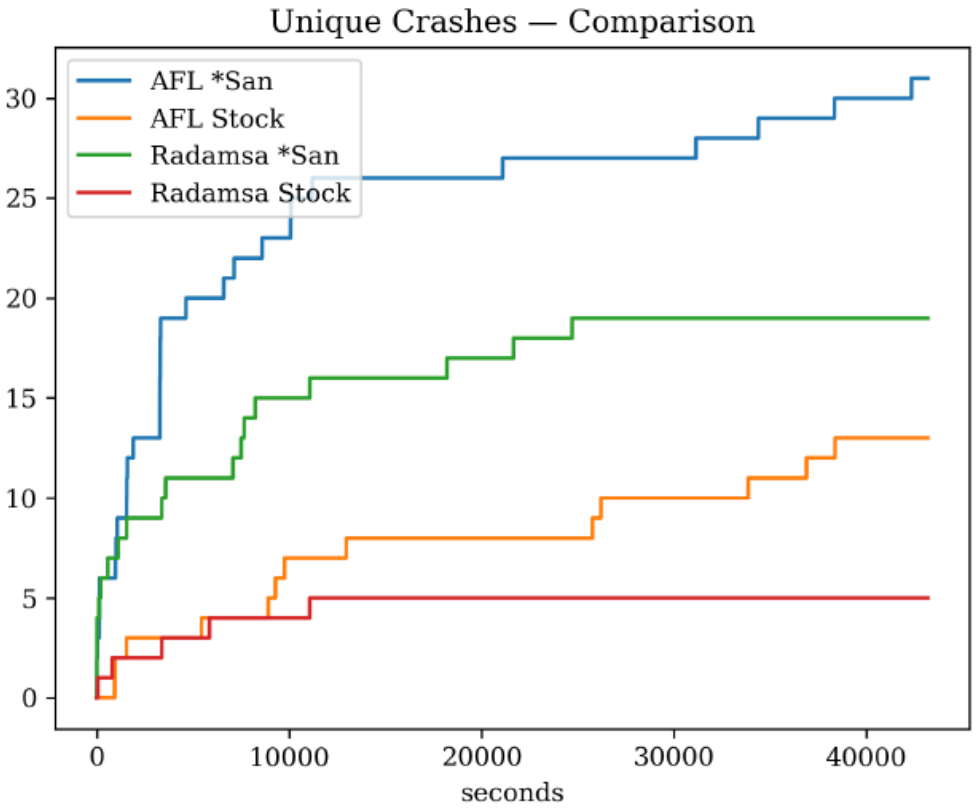


Figure 1: Comparison of unique crashes found over time

# group 14 - Radare

| Bug class | Sub class | Sink | AFL | AFL* | Radamsa | Radamsa* | Reference |
|---|---|---|---|---|---|---|---|
| Segmentation Fault | null deref | util.c:76 | X | X | X | X | Fixed in [9] |
| | null deref | elf.c:3818 | X | X | X | X | Fixed in [9] |
| | null deref | pe.c:1067 | X | X | X | X | Reported in [10] |
| | null deref | pe.c:1075 | X | X | | | Reported in [10] |
| | null deref | x509.c:26 | | X | | | Reported in [11] |
| | null deref | x509.c:286 | X | | | | Reported in [11] |
| Heap buffer overflow | read 1 byte | bfile.c:195 | | X | | X | Cannot repro |
| | read 1 byte | bfile.c:199 | | X | | X | Cannot repro |
| | read 1 byte | ne.c:375 | | X | | X | (not triaged) |
| | read 2 bytes | ne.c:83 | | X | | X | (not triaged) |
| | read 2 bytes | ne.c:393 | | X | | | (not triaged) |
| | read 2 bytes | ne.c:394 | | X | | | (not triaged) |
| | read 2 bytes | ne.c:396 | | X | | | (not triaged) |
| | read 2 bytes | ne.c:483 | | X | | X | (not triaged) |
| | read 8 bytes | pkcs7.c:651 | X | X | | X | (not triaged) |
| | read 8 bytes | x509.c:190 | | X | | | (not triaged) |
| | read 40 bytes | pe.c:4241 | | | | X | Reported in [12] |
| Heap buffer overflow | write 1 bytes | marshal.c:226 | X | X | | | Reported in [13] |
| | write 183 bytes | memmove | | X | | | Reported in [14] |
| | write 228 bytes | memmove | X | | | | Reported in [14] |
| | write 5282 bytes | memmove | X | | | | Reported in [14] |
| | write 54 bytes | memmove | X | | | | Reported in [14] |
| | write 61 bytes | memmove | X | | | | Reported in [14] |
| Use after free | read 4 bytes | marshal.c:788 | | X | | X | Reported in [15] |
| Signed integer overflow | 'int' | marshal.c:199 | X | X | X | X | Solved by [15] |
| | 'int' | pe.c:4244 | | X | | | (not triaged) |
| | 'long long int' | bin_elf.inc:634 | | X | | | (not triaged) |
| | 'long long int' | bin_elf.inc:636 | | X | | | (not triaged) |
| Invalid bitwise shift | exponent too large | ne.c:297 | | X | | X | (not triaged) |
| | exponent too large | ne.c:560 | | X | | X | (not triaged) |
| | negative left shift | bin_mz.c:51 | | X | | X | (not triaged) |
| | negative left shift | pe.c:3397 | | X | | X | (not triaged) |
| | shift exceeds int type | pe.c:3397 | | X | | X | (not triaged) |
| Alignment problems | member access with misaligned address | coff.c:41 | | X | | X | (not triaged) |
| | load of misaligned address | ne.c:396 | | X | | X | (not triaged) |
| | load of misaligned address | bin_pe.c:433 | | X | | | (not triaged) |

Table 2: Flaws found by each fuzzer. The * indicates the presence of sanitizers

# group 16 : Fuzzing PQ candidates?

done in general and we would like to continue working in this area and try and fuzz the NIST post quantum candidates [2] (by the time we came up with this idea it was too late, but we will perform that as external task).

# group 21 – GIF decoder

| Mutation Rate | Crashes and Hangs Found | Inputs Processed |
|---|---|---|
| 0.0001 | 305 | 10150 |
| 0.001 | 543 | 2451 |
| 0.01 | 539 | 2492 |
| 0.1 | 37 | 29679 |

Table 4: Testing different rates for zzuf

| | AFL | zuff | Radamsa |
|---|---|---|---|
| Segmentation faults | 619 | 12,902 | 2,209 |
| Invalid pointers | 27 | 5,813 | 208 |
| Double free or corruption | 16 | 105 | 9 |
| Free invalid size | 8 | 1,218 | 5 |
| Corrupted size | 6 | 9,799 | 128 |

Table 7: The different kinds of crashes we found with the different tools