Fuzzing Experiences

Erik Poll

Digital Security

Radboud University Nijmegen

Radboud Universiteit Nijmegen

TRU/e Master in Cyber Security

Dumb mutational fuzzers: Radamsa & zzuf

- Radamsa typically better than zzuf
- Tweaking parameters of zzuf for optimal results can be tricky
- Bottleneck: malformed inputs may be rejected straight away, and not get very deep into the code.
  - E.g. due to incorrect initial bytes or broken CRC check (e.g. in PNG)
  - NB many inputs does not always mean thorough testing, with good coverage
  - Programs rejecting such malformed files (without or ideally with some error message) is not a bug, and certainly not a security bug

  Solutions:

  - Not mutation of initial n bytes of a file
  - Remove correctness checks from code
- Note that you should start these fuzzers with a legal input

AFL

- Overall much better than dumb fuzzers
  - When comparing raw numbers of problems found by AFL and say zzuf beware that latter will contain many duplicates
- Typically Asan needed to get interesting warnings,
  but in a few cases just AFL on its own could produce e.g. seg-faults
- Bottleneck with malformed inputs being rejected straightaway may still exist
  - AFL will figure out that some initial header should left unchanged, but it will not figure out that file should have correct CRC checksum
  - ALF does have dictionary option to guide generation of mutations by means of a grammar

Input sizes

- Large input files are *not* the best approach
  - execution will be slower
  - random mutation unlikely to hit interesting places; test mutations may simply try ‘more of the same’

Some interesting results

- Group 15
  - ImageMagick with JNG & RLE formats
  - difference 6.7.7 and 7.0.9 versions
- Group 16
  - Radamsa on ed with txt format
- Group 25
  - Polybar with INI file format
  - memory flaws (illegal writes & reading uninitialized memory) found with ASan & MSan