Continuously fuzzing and improving Suricata

Philippe Antoine
Who am I?
What is fuzzing for?

Find bugs!

Efficiently

Make Suricata more robust against untrusted inputs
What do I call fuzzing?

Demo
while(CPUavailable()) {
    input = GetInputFromCorpus(corpus);
    mutate(&input);
    process(input, &feedback); // checks for bugs
    updateCorpus(&corpus, input, feedback);
}
What kinds of bug?

- Address Sanitizer ASAN (include leaks)
- Memory Sanitizer (MSAN)
- Undefined Sanitizer (UBSAN)
- Timeouts, out-of-memory
- Assertions
What about Suricata?
Where is my fuzzer?

Outside of Suricata
• Fuzz drivers
• Corpuses
• Oss-fuzz

In Suricata
• Fuzz targets
• Asserts

In src/tests/fuzz
Targets

- fuzz_applayerparserparse
- fuzz_applayerprotodetectgetproto
- fuzz_confyamllloadstring
- fuzz_decodepcapfile
- fuzz_mimedecparseline
- fuzz_siginit
- fuzz_sigpcap
Oss-fuzz

- Continuously run by Google
- Various fuzz drivers, architectures, sanitizers...
- Input: project configuration
- Output: stats, bugs
- ClFuzz
### Suricata in oss-fuzz

**Input**
- Project.yaml
- Dockerfile
- Build.sh

**Output**
- Stats
- Bugs
Suricata 6.0.1

- 37 redmine tickets
- 10 from oss-fuzz
  - One incomplete fix from 6.0
- 15 oss-fuzz issues
  - Two issues with variants
  - 2 (temporary) build failures
  - One unreproducible false positive
How can you contribute?
While coding and testing

- More inputs for corpuses
- More asserts
- Fix bugs (why was it not found by fuzzing ?)
- Run targets on corpus as part of CI

- More targets
- Better targets (performance, coverage)
While fuzzing Suricata

- More targets
- Better targets
  - Performance
  - Coverage
  - Structure-aware (aka not fuzz libpcap)
  - Differential fuzzing (evasions through TCP split)
- More/better fuzz drivers
- Run fuzzers and report bugs
- Improve oss-fuzz Suricata project
Thank you

Questions?