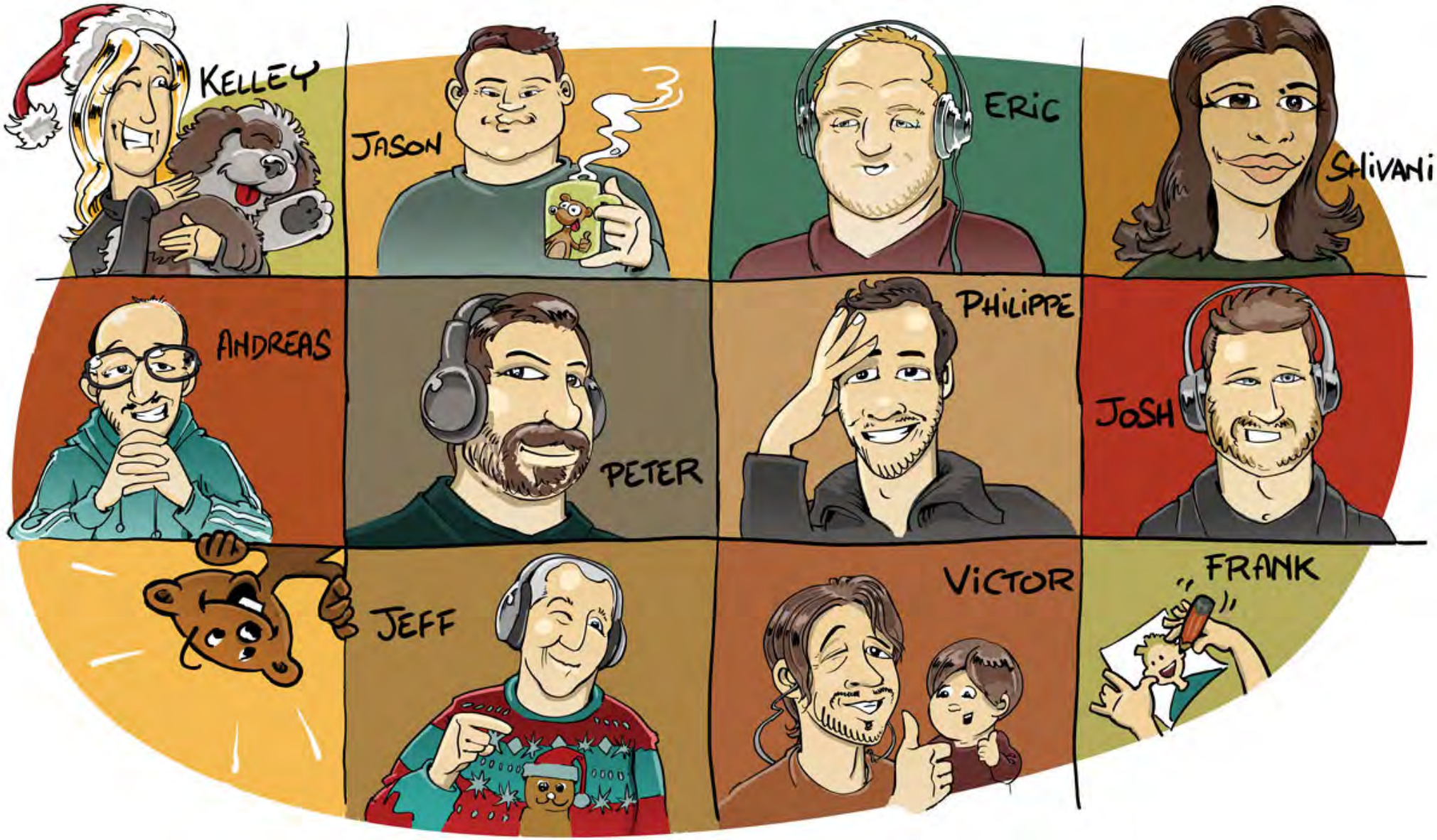
# Keeping on fuzzing and fixing Suricata

Philippe Antoine


SURICON BOSTON 2021

OISF

# Who am I ?




SURICATA

# Threat model

# What do I call fuzzing ?

```
while(CPUavailable()) {
     input = GetInputFromCorpus(corpus);
     mutate(&input);
     process(input, &feedback); // checks for bugs
     updateCorpus(&corpus, input, feedback);
}
```

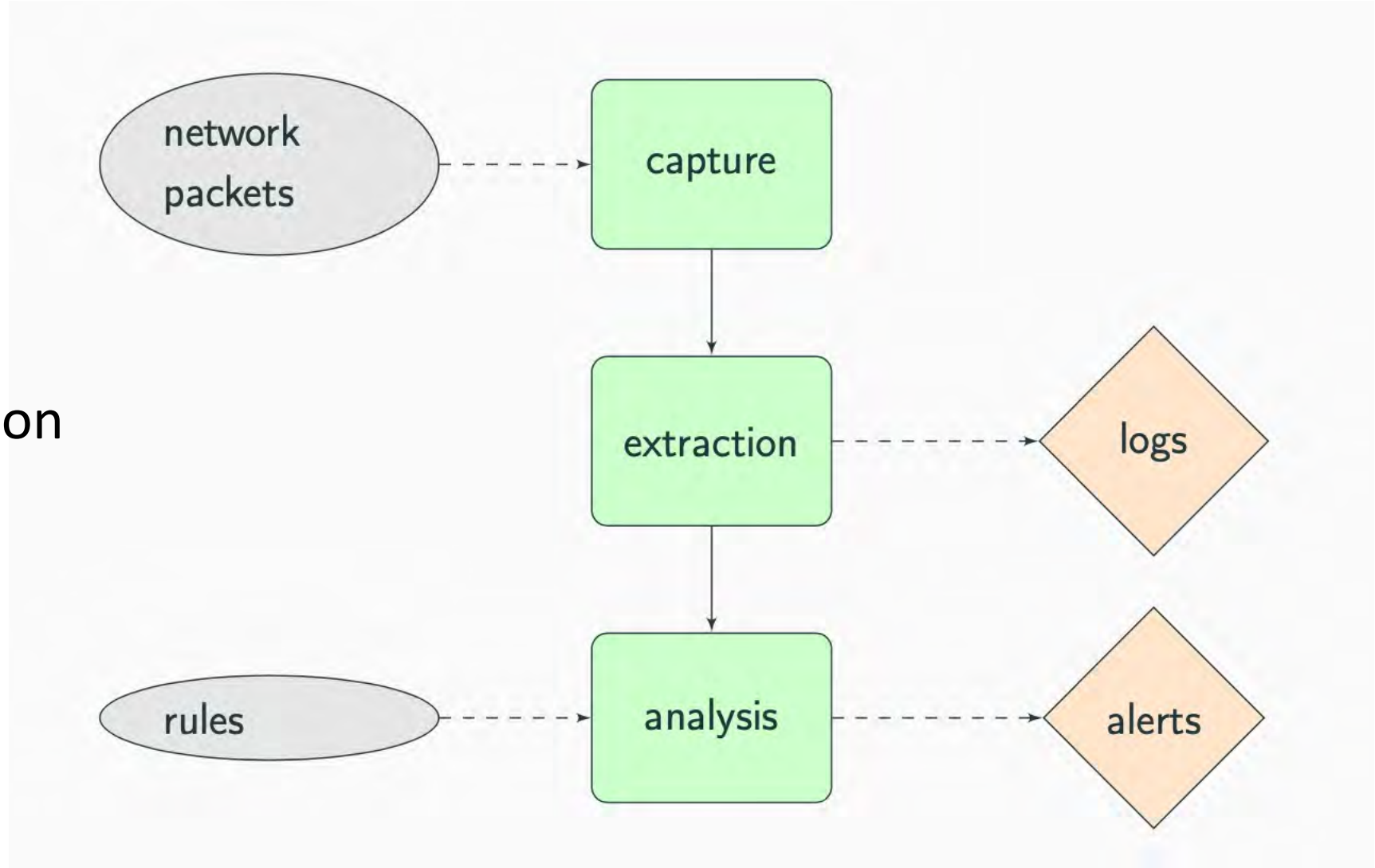
SURICATA

# What do I call fuzzing ?

Demo

# Suricata inputs

Configuration



# Why fuzzing Suricata ?

Find bugs to fix them !

Efficiently

So that Suricata provides a positive cybersecurity value

# Cybersecurity value

- Remote Code Execution : negative value
- Complete Denial Of Service : zero value
  - Rust panics
  - NULL dereferences
- Partial Denial Of Service : positive, but reduced value
  - Memory leaks
  - Quadratic complexity
  - Logic bugs/evasions/failed assertions

SURICATA

# Assessing bugs

- Bug class
- Default configuration ? Specific rules ?
- One or both network endpoints controlled by an attacker ?
- Which versions are affected ?

SURICATA

# Results since Suricata 6.0.0

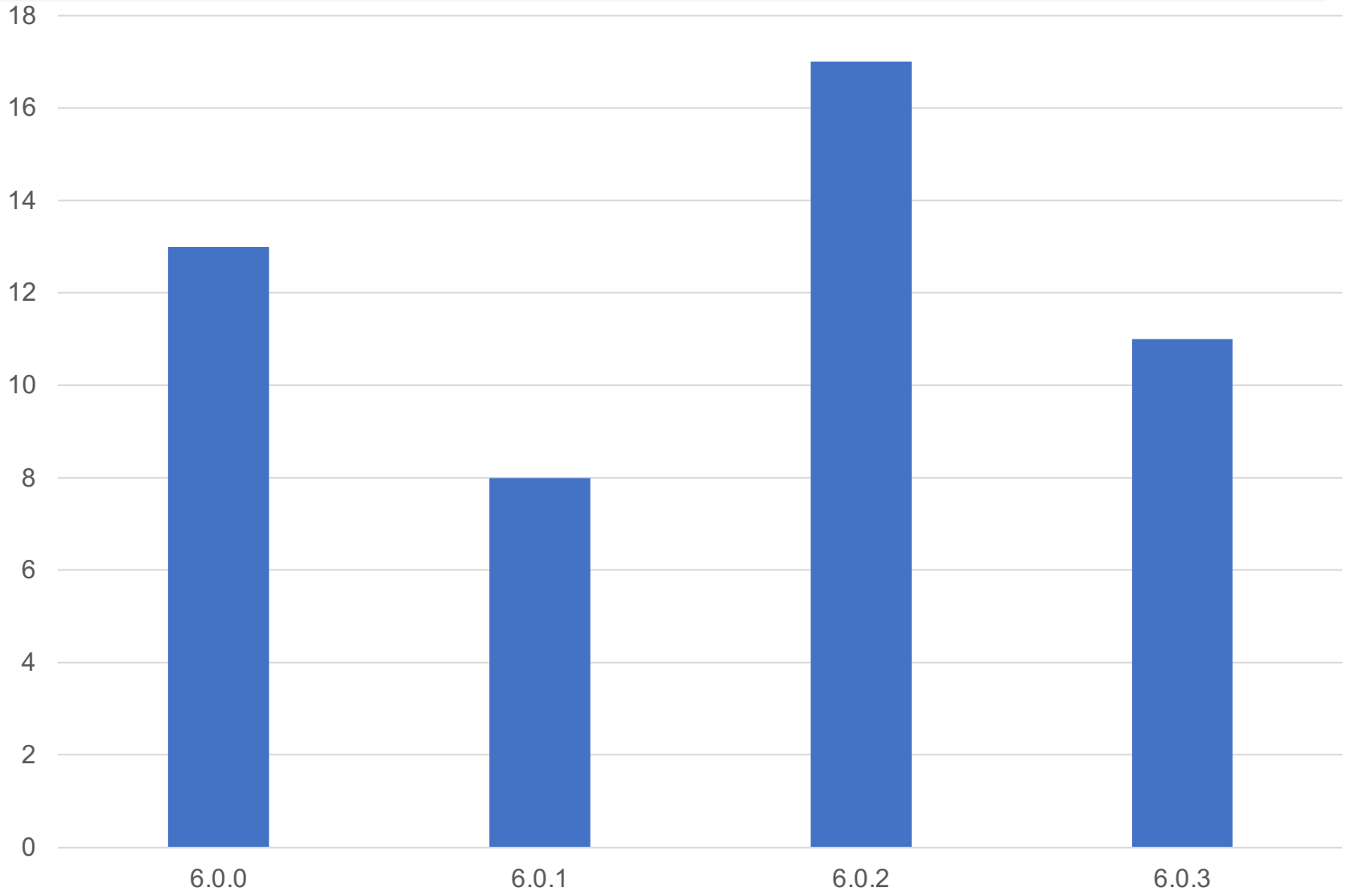
# Methodology

- Sources:
  - https://bugs.chromium.org/p/oss-fuzz/issues/list?q=label%3AProj-suricata
  - https://redmine.openinfosecfoundation.org/
  - Only bugs that reached git master (no CIFuzz stats)
- Manual review
  - Remove duplicates
  - Remove false positives (bug in targets)
- Complete data
  - https://catenacyber.fr/suricata-fuzz.csv
  - https://catenacyber.fr/suricata-bugs.csv

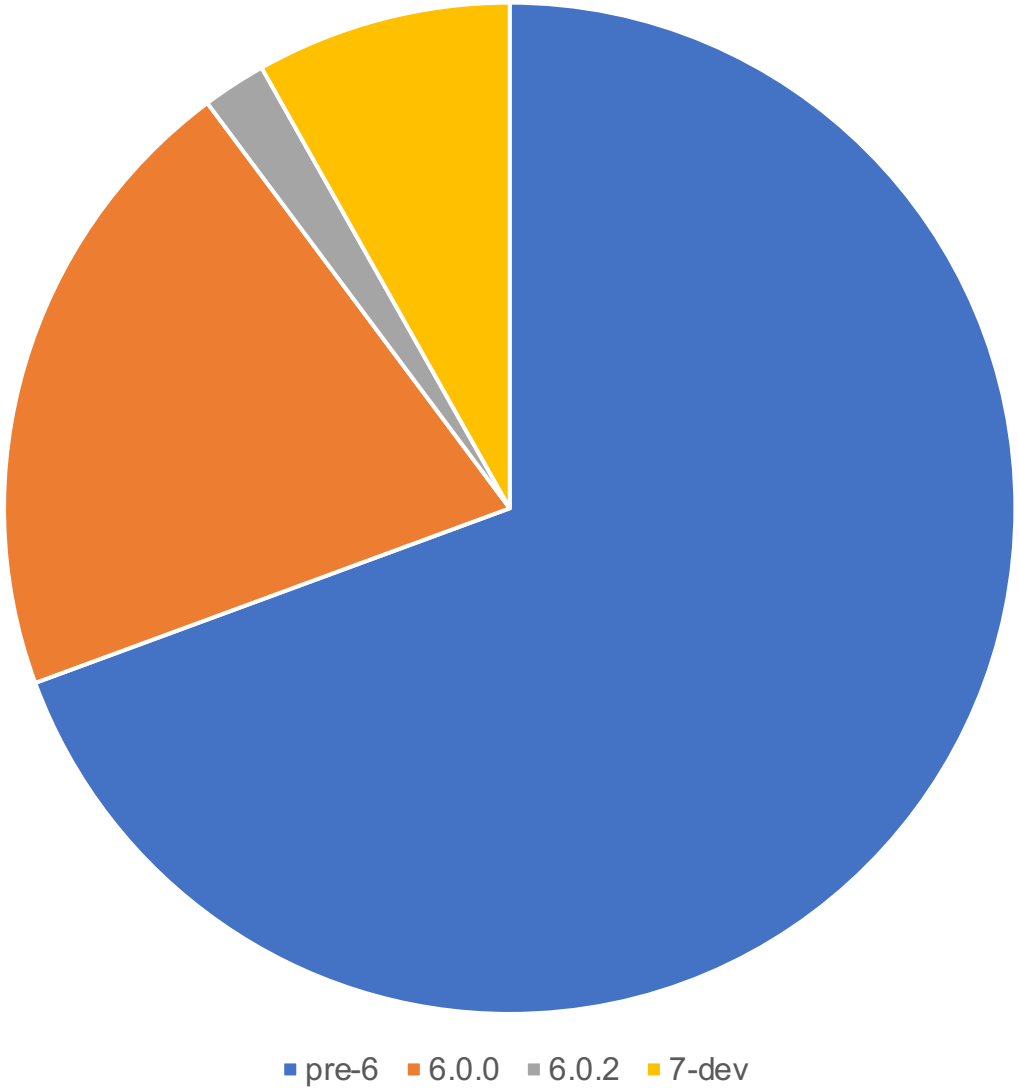
# Overall results

- 49 unique bugs in about one year
- 41 reported only by oss-fuzz, 8 reported also by other means
- Every bug found fixed in next version except :
  - 4 bugs fixed before getting into any release
  - 1 bug found in 6.0.0 fixed in 6.0.4 (evasion)
  - 1 bug found in 6.0.0 left until 7 (PCRE2)
  - 1 bug introduced and found in 6.0.2 to be fixed in 6.0.4 (RCE)

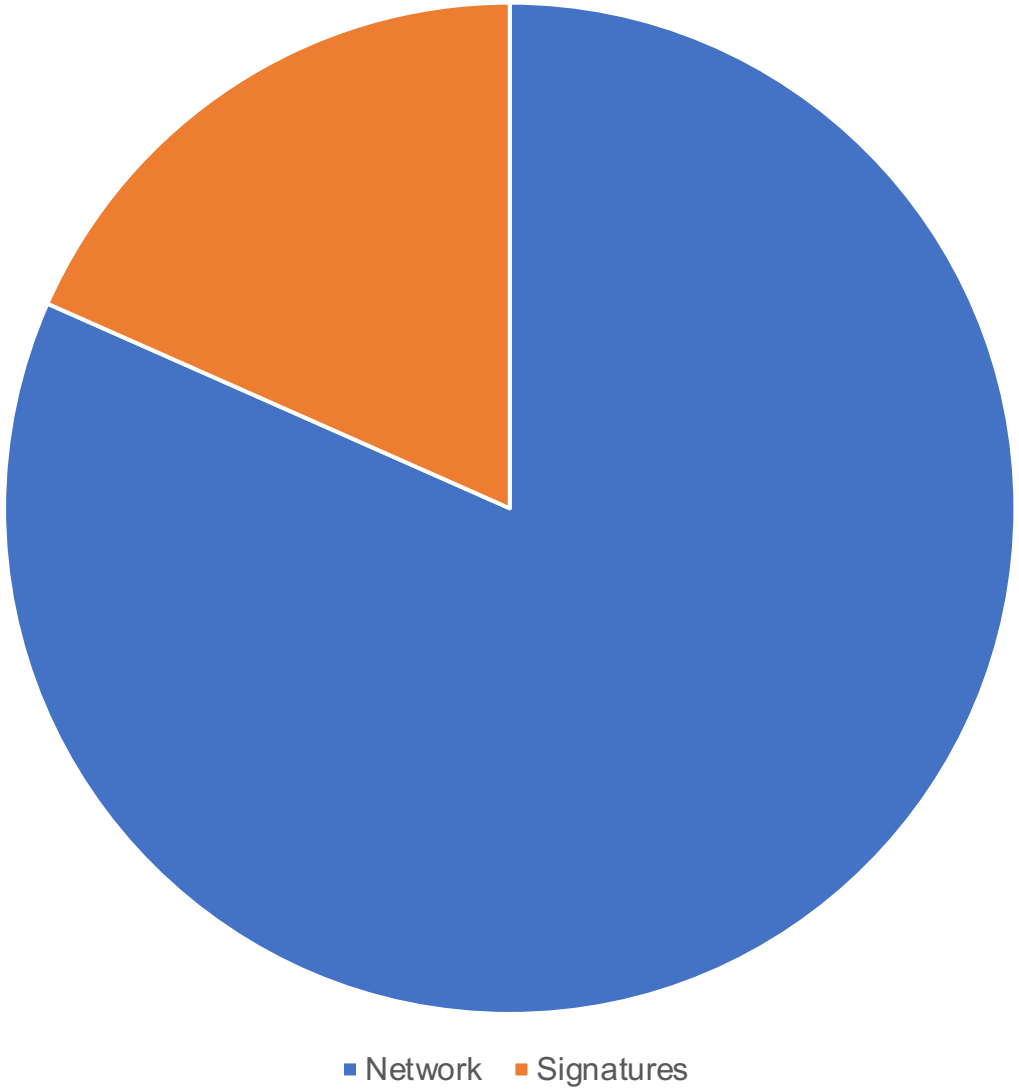
# Bugs found in version



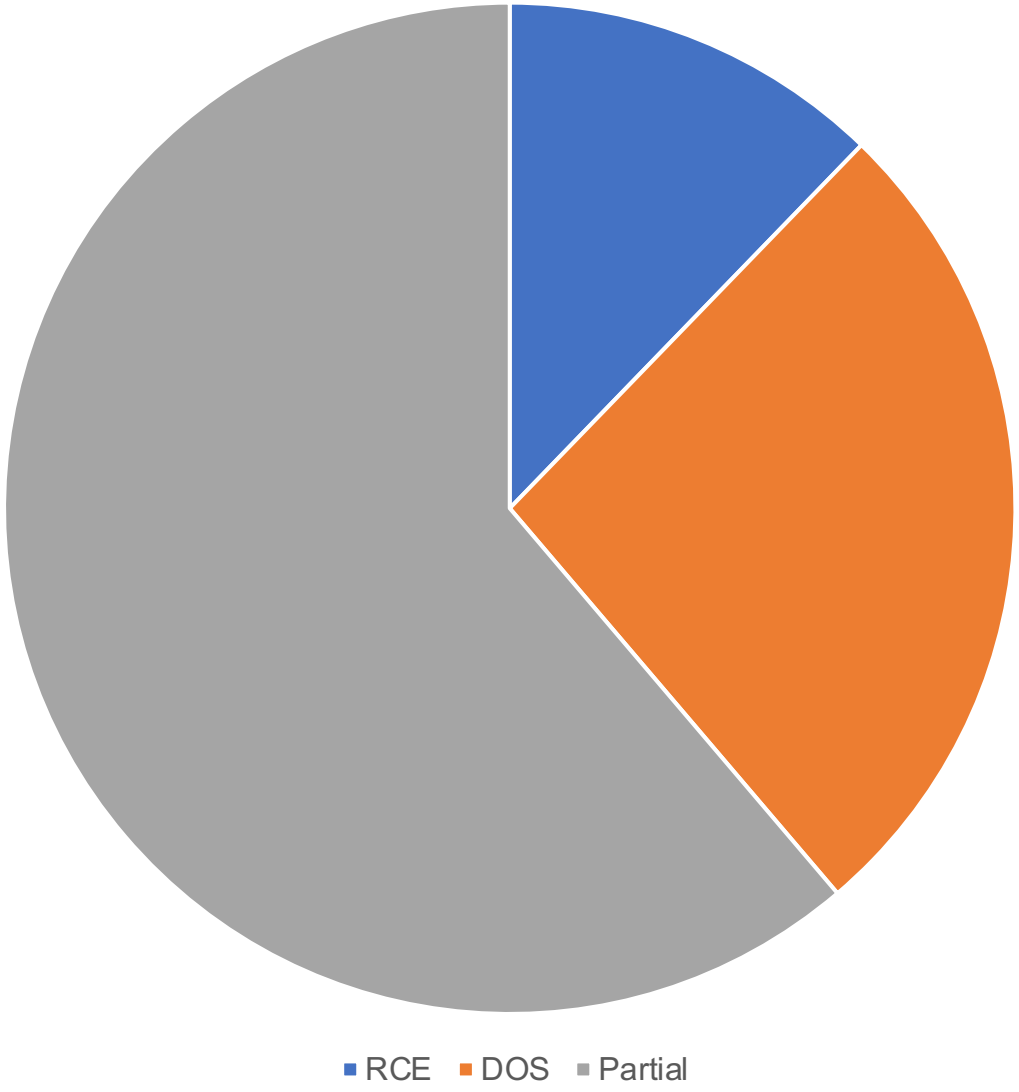
# Bugs appearance



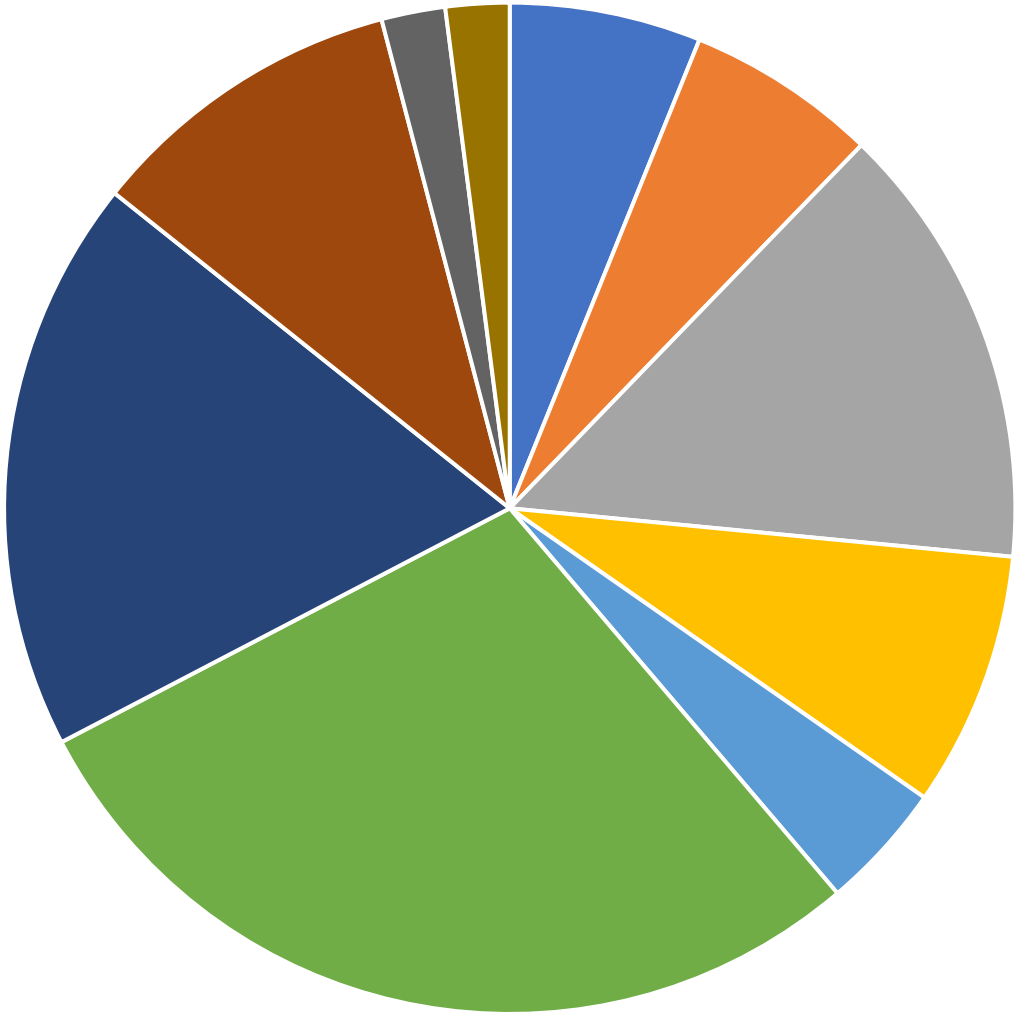
# Bugs found by source



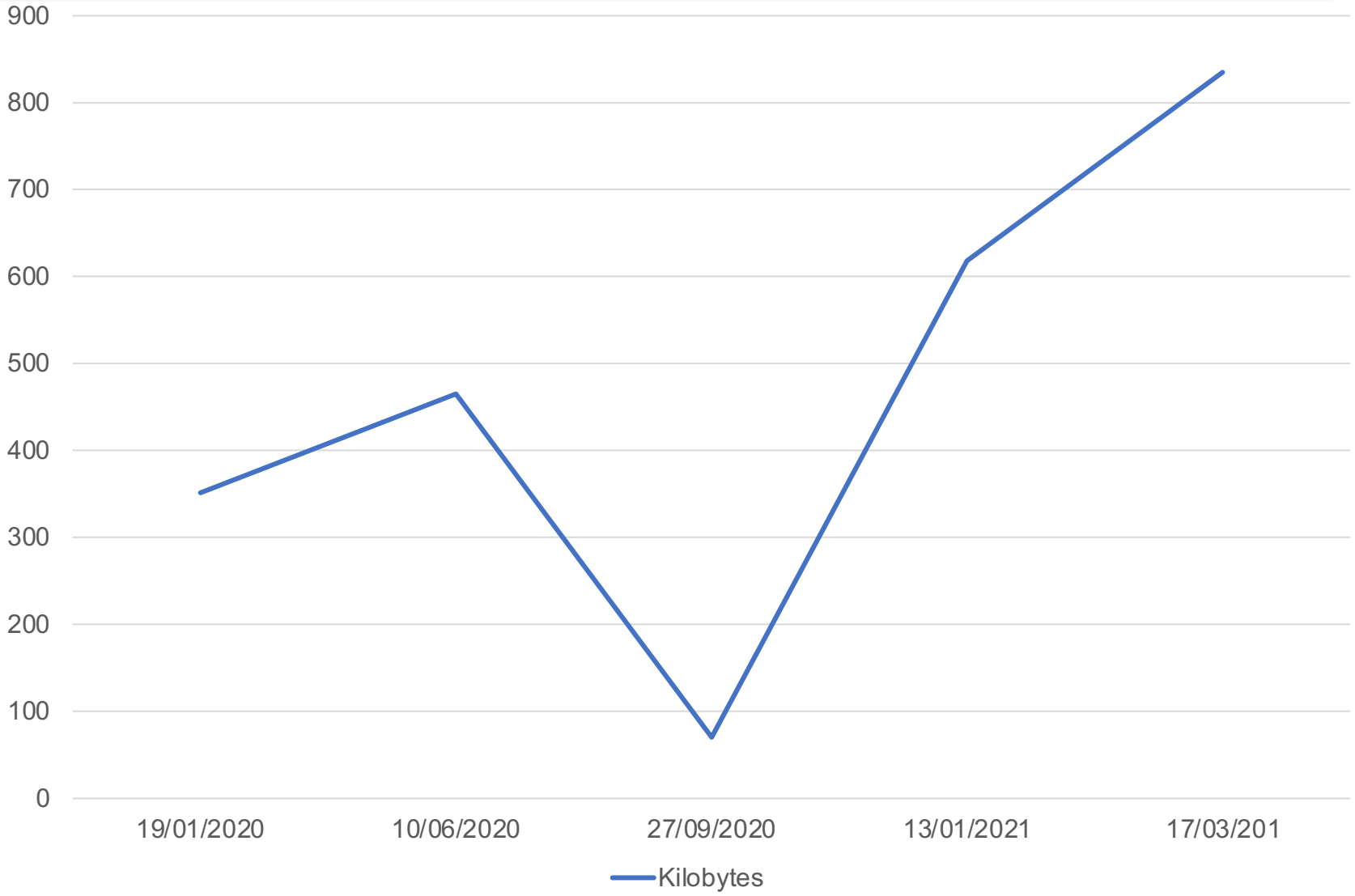
# Bugs found by impact



# Bugs found by impact



# Quadratic complexity in libhtp

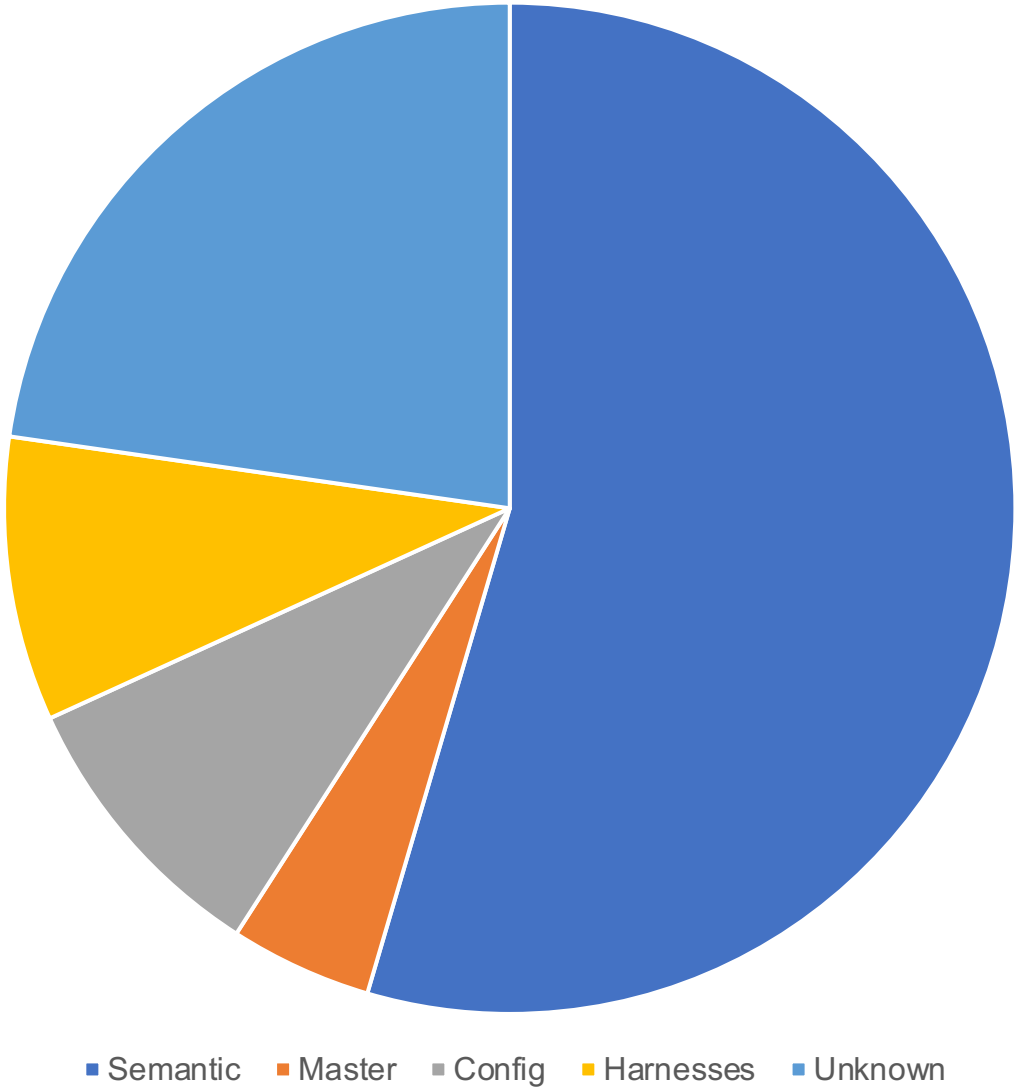


# Bugs missed by fuzzing

- 34 redmine tickets for 6.0.3
  - 1 duplicate, 1 irrelevant
  - 9 uniquely found by oss-fuzz
  - 1 found by oss-fuzz already reported otherwise
  - 22 missed

https://redmine.openinfosecfoundation.org/versions/167

# Reasons for fuzzers misses



# Improvements

# Coverage

- Your fuzzer cannot find bugs in code it does not cover
- Mix of Rust and C
- RUSTFLAGS -Cpasses=sancov

https://github.com/google/oss-fuzz/pull/4697
https://github.com/google/oss-fuzz/pull/5352
https://github.com/google/oss-fuzz/pull/6517

# Coverage

## Coverage Report

View results by: Directories | Files

| PATH | LINE COVERAGE | FUNCTION COVERAGE | REGION COVERAGE |
|---|---|---|---|
| libhtp/ | 49.93% (4241/8494) | 56.25% (234/416) | 40.88% (4007/9802) |
| rust/ | 21.95% (4382/19967) | 37.84% (400/1057) | 12.48% (1985/15911) |
| src/ | 36.81% (46197/125487) | 40.44% (2178/5386) | 33.77% (40456/119789) |
| TOTALS | 35.61% (54820/153948) | 41.00% (2812/6859) | 31.92% (46448/145502) |

2021 March 9th : first coverage report including Rust

SURICATA

# Coverage

## Coverage Report

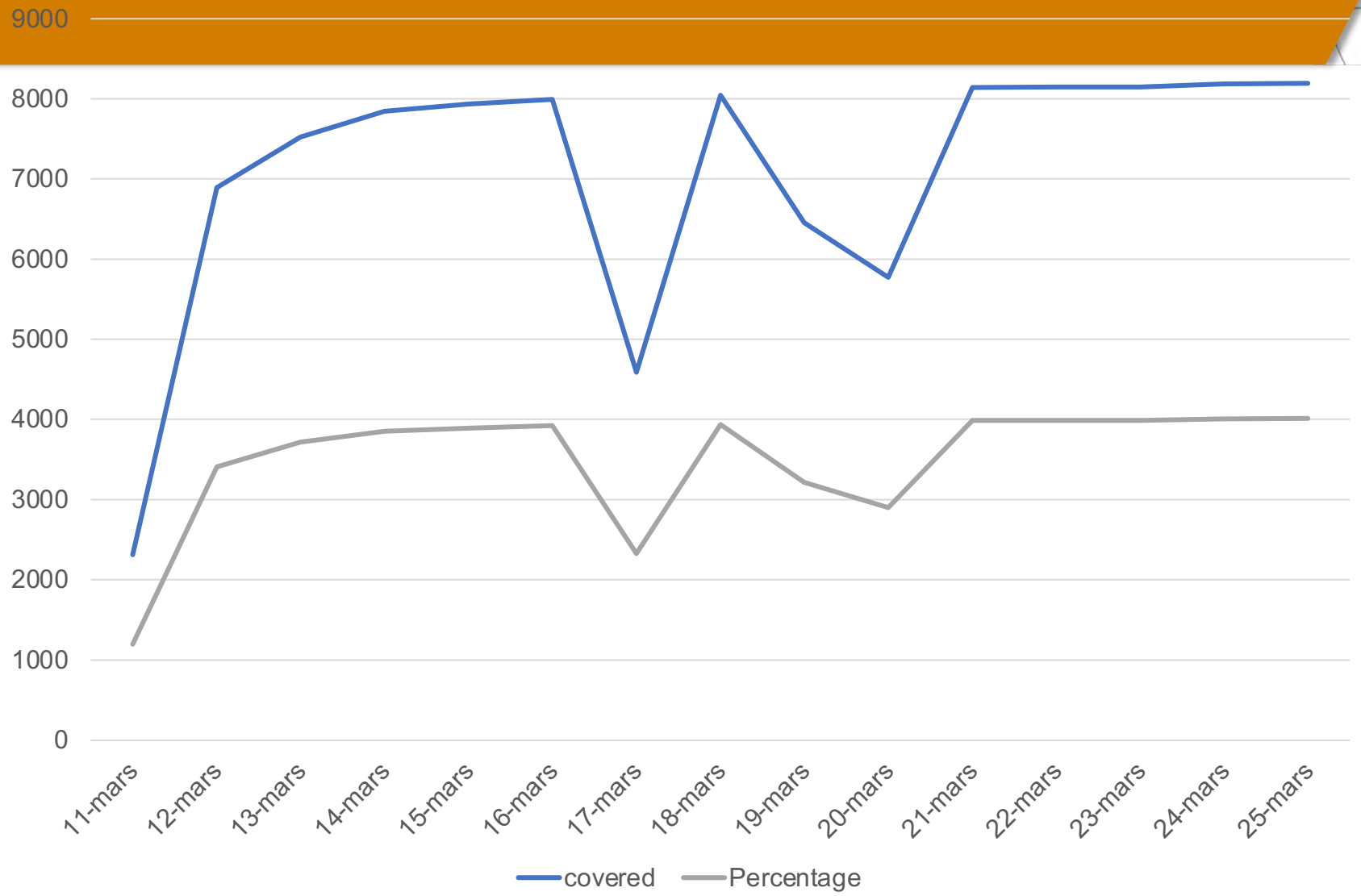
View results by: Directories | Files

| PATH | LINE COVERAGE | FUNCTION COVERAGE | REGION COVERAGE |
|---|---|---|---|
| libhtp/ | 50.22% (4144/8252) | 56.39% (234/415) | 41.37% (4036/9755) |
| rust/ | 71.35% (15461/21668) | 70.68% (1150/1627) | 59.77% (8939/14956) |
| src/ | 52.85% (65739/124377) | 58.87% (3149/5349) | 54.85% (63150/115140) |
| TOTALS | 55.31% (85344/154297) | 61.33% (4533/7391) | 54.43% (76125/139851) |

2021 March 31st

SURICATA

# Results



# More of the same fuzz target

- fuzz_applayerparserparse
- Choose app-layer protocol based on data[0]
- Or force one specific with env variable FUZZ_APPLAYER

https://github.com/google/oss-fuzz/pull/5596

SURICATA

# Results

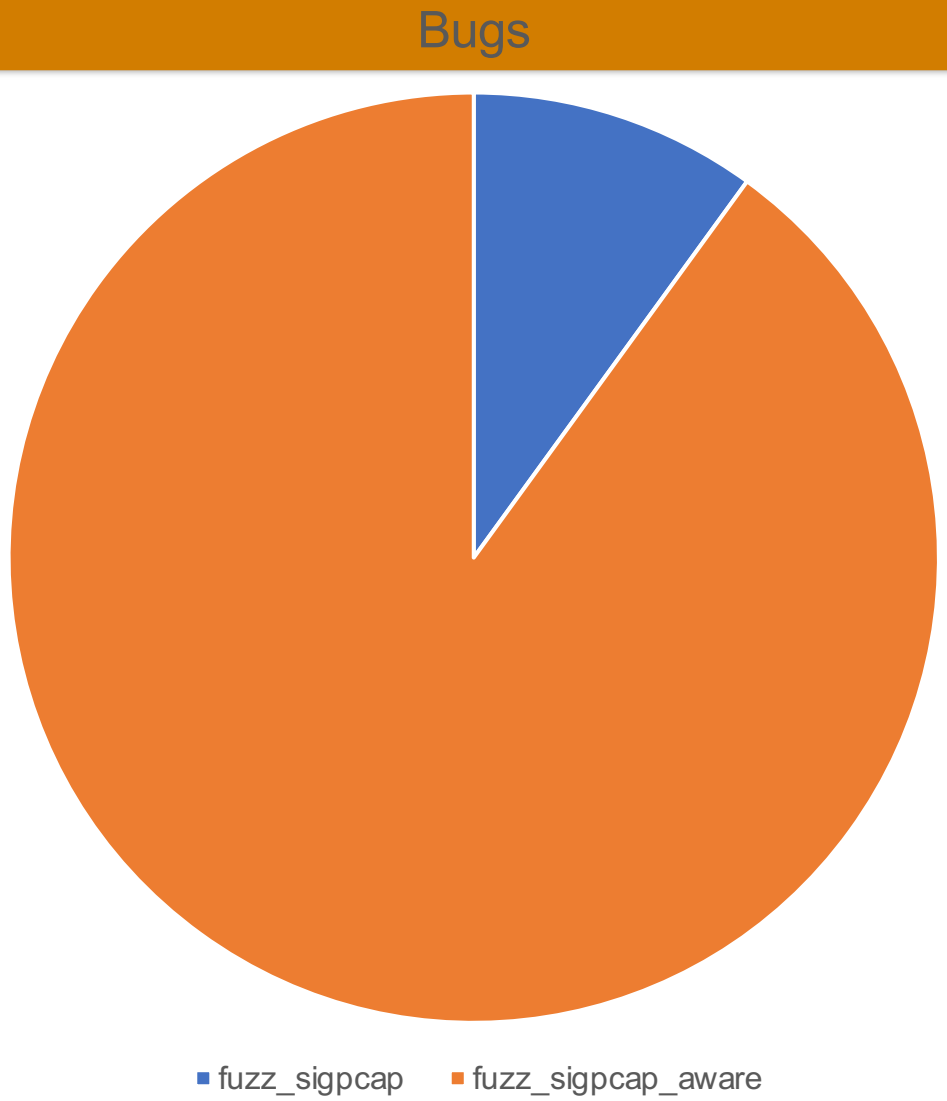
- 1 fuzz_applayerparserparse_dcerpc
- 1 fuzz_applayerparserparse_dnp3
- 1 fuzz_applayerparserparse_ftp
- 2 fuzz_applayerparserparse_http2
- 1 fuzz_applayerparserparse_ike
- 1 fuzz_applayerparserparse_mqtt
- 1 fuzz_applayerparserparse_smtp
- 0 fuzz_applayerparserparse

# Structure-aware fuzzing

- fuzz_sigpcap
- We want to fuzz suricata, not libpcap
- https://github.com/catenacyber/fuzzpcap
- TCP single stream

https://github.com/google/oss-fuzz/pull/5528

# Results

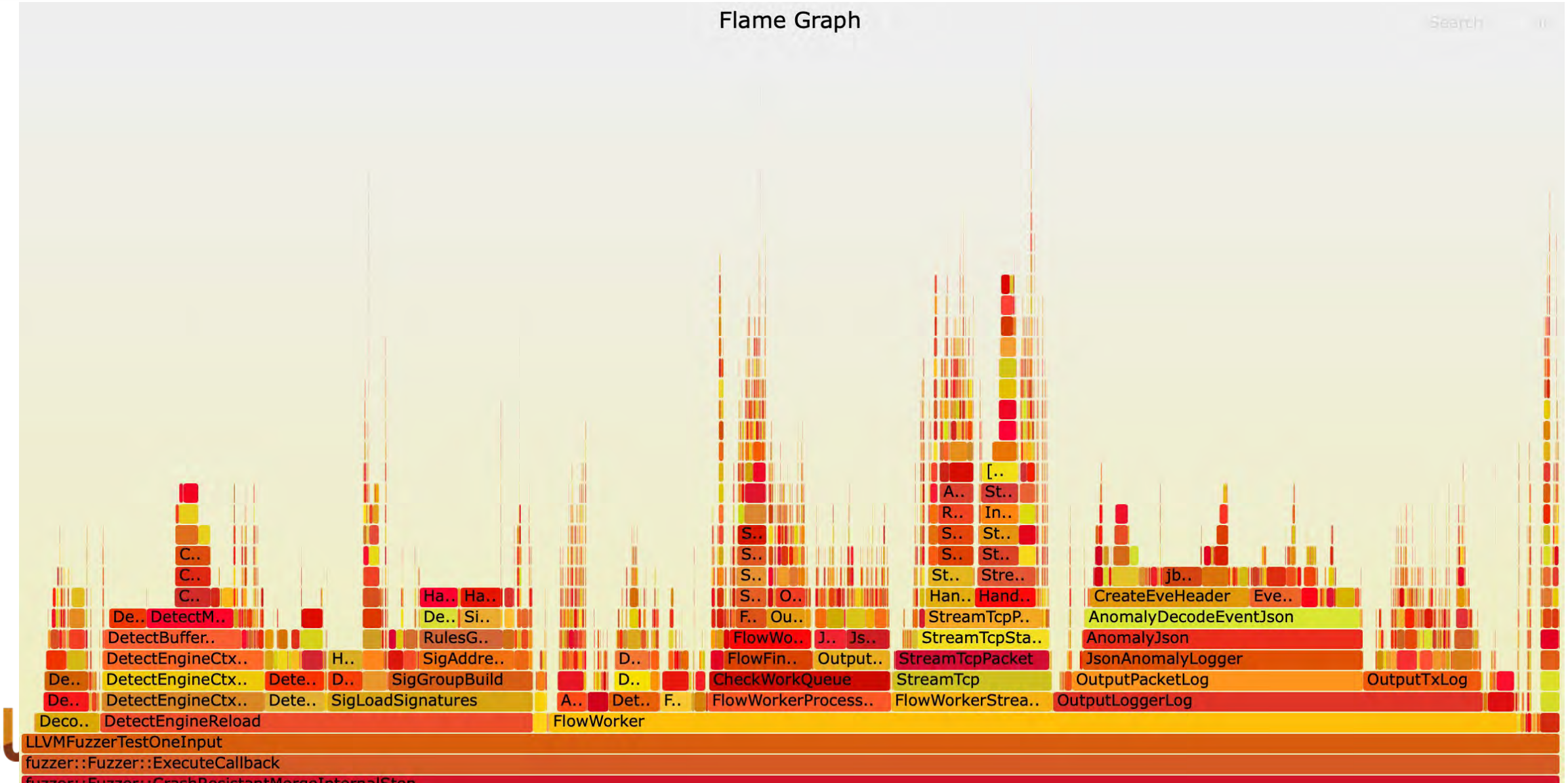


# Performance report

- POC : run on corpus
- Generate flamegraph

https://github.com/google/oss-fuzz/pull/5995

# Flamegraph


Flame Graph
Search
LLVMFuzzerTestOneInput
fuzzer::Fuzzer::ExecuteCallback
FlowWorker
DetectEngineReload
SigLoadSignatures
SigGroupBuild
DetectEngineCtx..
DetectBuffer..
DetectM..
RulesG..
SigAddre..
FlowWorkerProcess..
CheckWorkQueue
FlowFin..
FlowWo..
StreamTcp
StreamTcpPacket
StreamTcpP..
StreamTcpSta..
OutputLoggerLog
OutputPacketLog
JsonAnomalyLogger
AnomalyJson
AnomalyDecodeEventJson
CreateEveHeader
OutputTxLog

# Improvements summary

- Coverage investigation
- Performance investigation
- Split per app-layer protocol
- Structure-awareness
- Extend configuration
- More asserts
- Asking why a bug was not found by fuzzers

SURICATA

# Improvements to be done

- Just done : Default ruleset and fuzz only pcap
- Blockers
- Differential fuzzing for TCP splitting evasion
- More coverage and performance inspection

https://redmine.openinfosecfoundation.org/issues/4125

# Thank you

Questions ?

# Coverage

## Coverage Report

View results by: Directories | Files

| PATH | LINE COVERAGE | FUNCTION COVERAGE | REGION COVERAGE |
|---|---|---|---|
| libhtp/ | 52.27% (4314/8254) | 59.52% (247/415) | 42.89% (4221/9841) |
| rust/ | 76.28% (19140/25092) | 63.52% (1480/2330) | 64.81% (10053/15512) |
| src/ | 55.75% (70084/125708) | 62.26% (3316/5326) | 58.60% (67314/114879) |
| TOTALS | 58.81% (93538/159054) | 62.48% (5043/8071) | 58.18% (81588/140232) |

2021 October 4th

SURICATA