What’s the Fuzz

ATS Summit

Kit Chan (kichan@apache.org)

What is fuzzing?

- Automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program
- Program can fail with crash, memory leak, or failing assertions

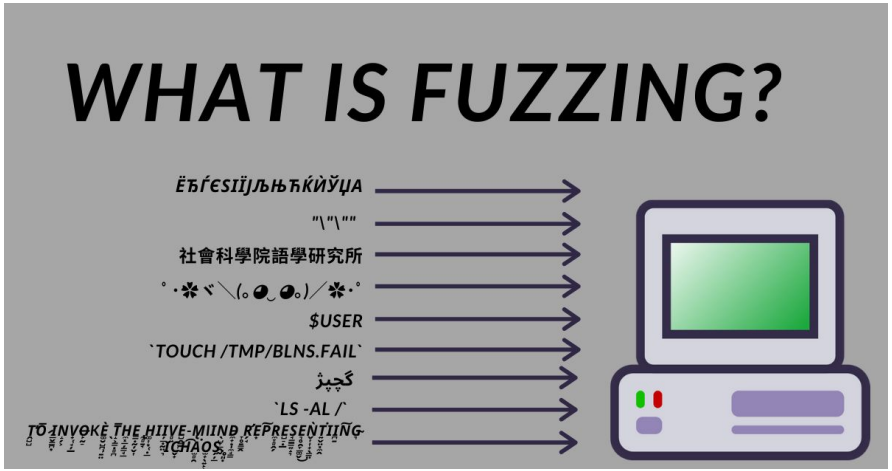

Figure 1. Image from "What is Fuzzing" by Keelan Parr, 2021, https://www.freecodecamp.org/news/whats-fuzzing-fuzz-testing-explained/

OSS-fuzz

Continuous Fuzzing for Open Source Software

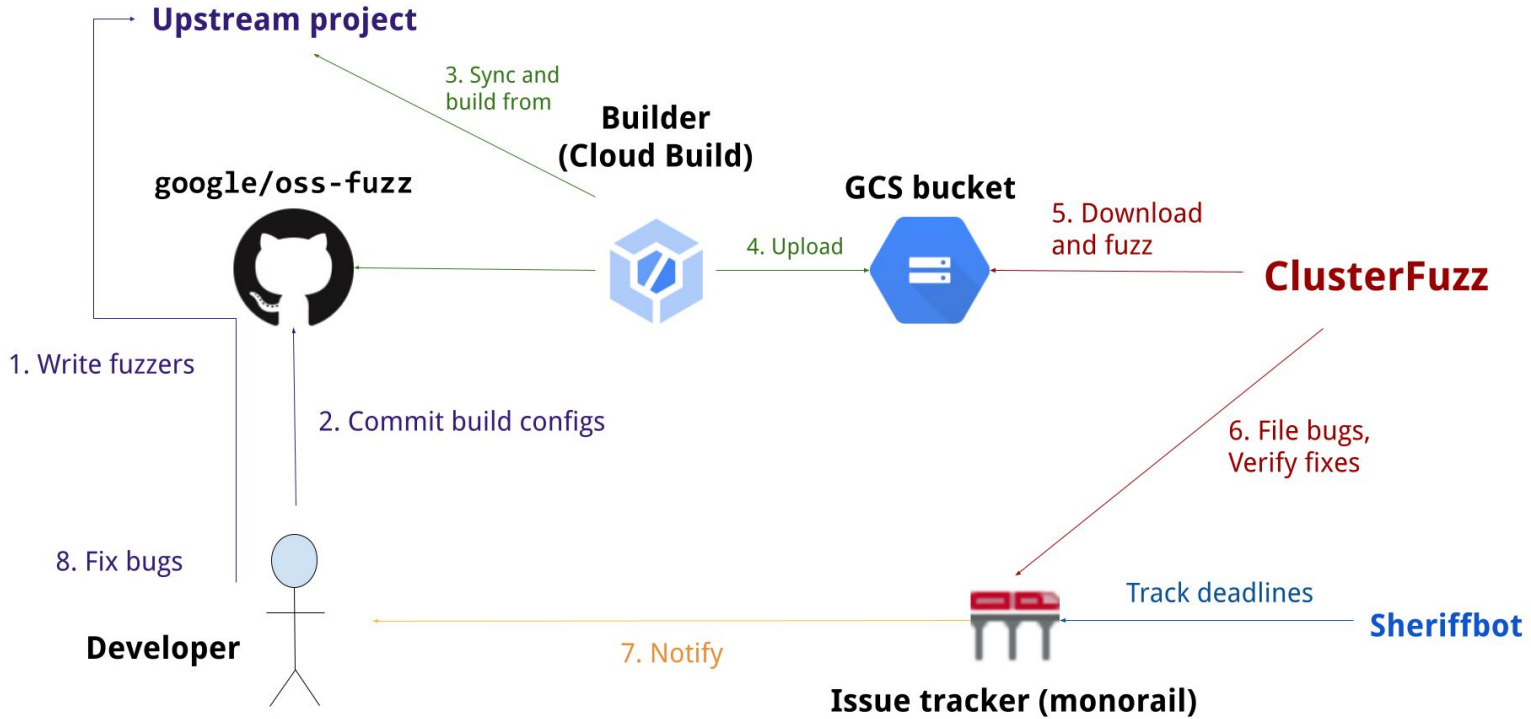

Figure 2: Image from "Architecture", https://google.github.io/oss-fuzz/architecture/

Fuzzer Anatomy - compilation

- Fuzzer program - https://github.com/google/oss-fuzz/blob/master/projects/trafficserver/fuzzer/FuzzEsi.cc
- Fuzzing Engines - E.g. libfuzzer, AFL, honggfuzz
  - For generating invalid, unexpected and random data
  - Coverage-guided
- Compilation recommends to include some sanitizer
  - Asan - address sanitizer (finding memory leak)
  - Msan - memory sanitizer (finding memory not initialized error)
  - Ubsan - undefined behavior sanitizer (finding the program to behave weirdly)

Fuzzer Anatomy - to increase efficiency

- Seed corpus - Initial set of inputs
- Dictionary - list of tokens / magic words
- Minimizer - Automatically find and group test cases into minimized input that can cause same behavior

Fuzzer Anatomy - Operation Details

- Parallel fuzzing
- Distributed fuzzing
- Continuous fuzzing

Setting up a project

https://github.com/google/oss-fuzz/tree/master/projects/trafficserver

- project.yaml - info on the open source project
- build.sh - script to build fuzzer
- Dockerfile - Docker image to run the build script
- fuzzer/ - Fuzzer program source code

Requires Google CLA - https://github.com/google/oss-fuzz/blob/master/CONTRIBUTING.md

Getting a report & Reproducing a problem

Example report - https://oss-fuzz.com/testcase-detail/4953201724227584

How to reproduce - https://google.github.io/oss-fuzz/advanced-topics/reproducing/

Web UI

- https://oss-fuzz.com
  - Coverage report
  - Crash statistics

Rewards and Disclosure

- Rewards from google on setting the project and fixing bugs
  - https://bughunters.google.com/about/rules/5097259337383936/oss-fuzz-reward-program-rules
  - For software with high Criticality score - to determine if it is important for internet infrastructure
    - https://github.com/ossf/criticality_score
    - ATS - 0.68283
- Bug Disclosure from OSS-fuzz
  - https://google.github.io/oss-fuzz/getting-started/bug-disclosure-guidelines/
  - 90 days
  - Weekend exemption
  - Grace period

Extra Info

- Support open source projects written in C/C++, Rust, Go, Java
- Fuzz Introspector - help you to write the fuzz suite
- ClusterFuzz can be used to do local fuzzing or for closed source projects
- Differential Fuzzing
  - Fuzzing inputs against series of similar applications (e.g. compilers, JVM, WAF, etc)
  - E.g. For proxy - ATS, envoy, haproxy, nginx, httpd
  - Observe the differences
  - Good for finding security issues
  - E.g Ffuf / T-Reqs

Questions for Discussions

Should we put more tests on OSS-fuzz?

Which unit tests should we convert to fuzz tests? E.g.

- plugins/esi/lib/EsiParser.h (done!)
- plugins/esi/lib/EsiGunzip.h
- plugins/esi/lib/EsiGzip.h
- iocore/net/ProxyProtocol.h
- proxy/hdrs/HTTP.h
- proxy/hdrs/URL.h
- proxy/hdrs/MIME.h
- proxy/hdrs/XPACK.h

Questions for Discussions

Which unit tests can we convert to fuzz tests? E.g. continue

- proxy/http/HttpTransact.h
- proxy/http/HttpBodyFactory.h
- proxy/http2/HTTP2.h
- proxy/http2/Http2Frame.h
- proxy/http3/Http3Frame.h
- proxy/http3/QPACK.h
- proxy/logging/LogUtils.h
- proxy/src/records/I_RecHttp.h

Any other code we want to write unit tests and put on OSS-fuzz?

References

- https://github.com/google/oss-fuzz/
- https://google.github.io/oss-fuzz/
- https://github.com/google/clusterfuzz