DocFuzz

Documentation-Based Fuzzing Stub Generator

Corey Capooci

MCS DE Student

Area of Investigation

What is Fuzzing?

- Fuzzing is a software testing technique that quickly and automatically explores the input space of a program without knowing its internals. (J. Jung, H. Hu, D. Solodukhin, D. Pagan, K. H. Lee, and T. Kim, *FuzziFication: Anti-Fuzzing Techniques*.)

Problem

Two Parts of Fuzzing

Fuzzing Engines

- Program that analyzes the library through calculated inputs.
- Examples:
  - libFuzzer
  - american fuzzy lop (AFL)
- Techniques in fuzzing engines received lots of focus

Fuzzing Stubs

- Inputted into the fuzzing engines.
- Provides the interface into the functions that will be tested.
- Requires expertise in library under test, and time to develop an effective stub.

Solution: DocFuzz

DocFuzz

- Utilizes natural language processing to create fuzzing stubs.
- Desired outcomes:
  - Output desirable library information using library documentation.
  - Fast stub generation times
  - Reliable stub generation
  - Code coverage results comparable to current state of the art fuzzing stub generators.

spacy.io

Methodology

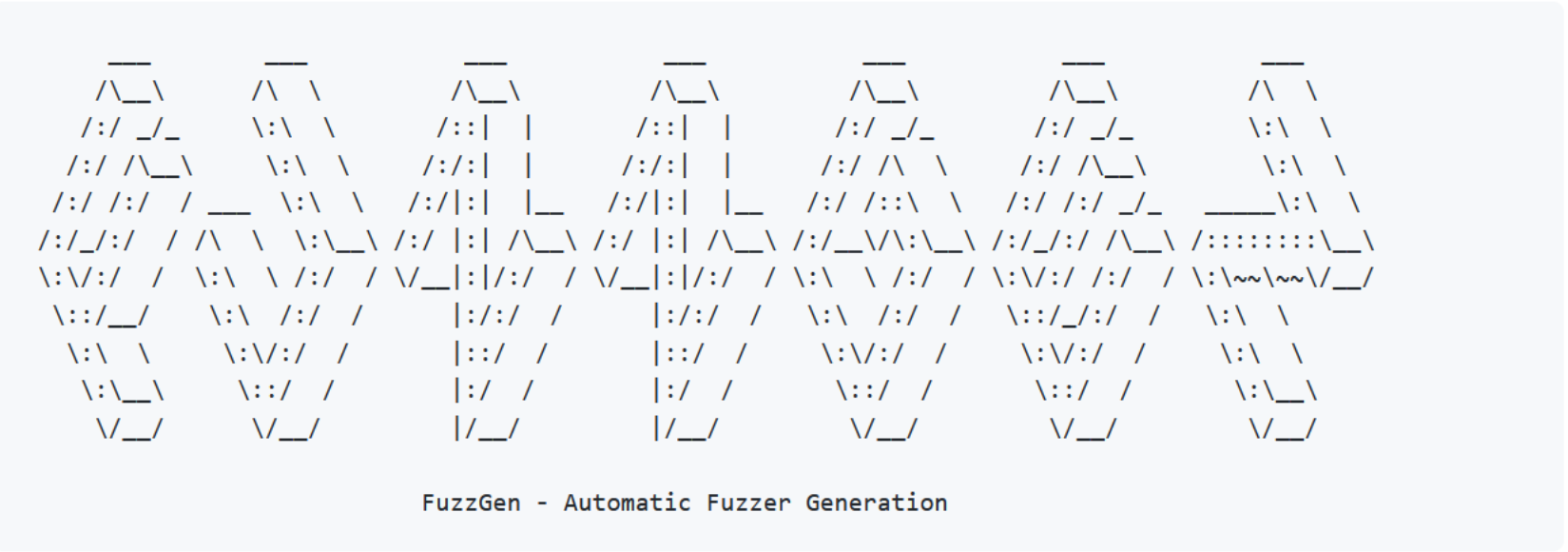

Image From https://github.com/HexHive/FuzzGen

Tests

- Speed evaluation
  - Comparison of stub generation times.
- Code coverage
  - Comparison of code coverage results across multiples libraries.
- Bug coverage
  - Comparison of bug coverage results.
- Time-lapse of code coverage
  - Graph of performance of NLP model.

Preliminary Results

Effective Fuzzing Stub Generation is Difficult

- FuzzGen requires:
  - The standard implementation of FuzzGen uses Android Open Source Project (AOSP)
    - According to Android.com, building AOSP requires:
      - At least 400GB of disk space for checking out code, and for building it.
      - At least 16 GB of RAM
  - Only certain branches of AOSP work with current FuzzGen.
  - FuzzGen requires the compiler to product assembler and object files of the source code
    - Not provided by default in AOSP.

Takeaways

Fuzzing Stub Generation has much room for improvement!

- Fuzzing is an testing effective tool that is gaining traction.
  - But… fuzzing stub generation is difficult and time consuming.
- There is still improvement to be made in the area of generating fuzzing stubs.
  - Make them easier to generate
  - Continue to improve code coverage
- More investigation into the use of NLP to accomplish this task.