



Friendly fuzzing in Python

□ Pavel Lonkin

Hi, there!



Pavel “Pasha” Lonkin



<https://www.linkedin.com/in/plonkin>



@pavel_lonkin



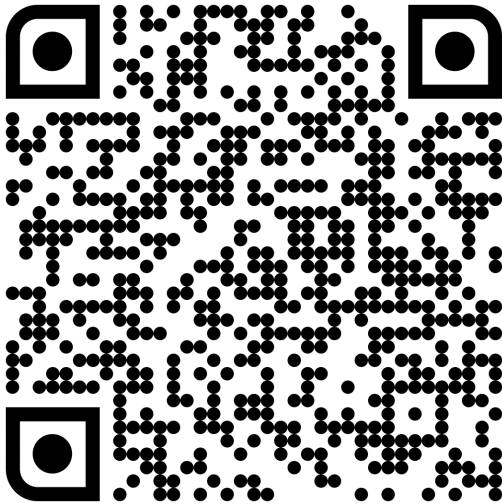
@d3rky

What is it about?

- ✓ What the fuzzing tests are?
- ✓ Why do I need to use fuzzing tests?
- ✓ Learn in practice how to cook the fuzzing tests using Atheris
- ✓ Homework and where to go to learn more?

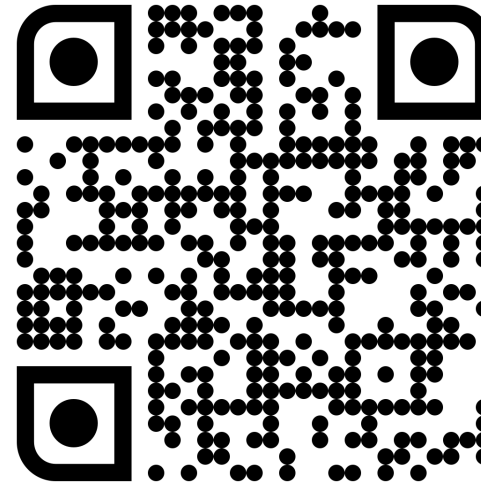
Step by step instruction

d3rky/pyday2022-bcn/blob/main/docs/instruction.md



Instruction

d3rky/pyday2022-bcn



Repo

What is fuzzing?

Fuzzing or Fuzz testing is a software testing technique of putting invalid or random data into the software system to discover coding errors and security loopholes

What is fuzzing?

 Data is generated in an automated or semiautomated way

What is fuzzing?

Data is generated in an automated or semiautomated way



Generated data is invalid and abnormal

What is fuzzing?

Data is generated in an automated or semiautomated way

Generated data is invalid and abnormal



It tries to “hack” your code

What is fuzzing?

Data is generated in an automated or semiautomated way

Generated data is invalid and abnormal

It tries to “hack” your code

 It catches system crashes, built-in code failures, unhandled exception


What is fuzzing?

Data is generated in an automated or semiautomated way

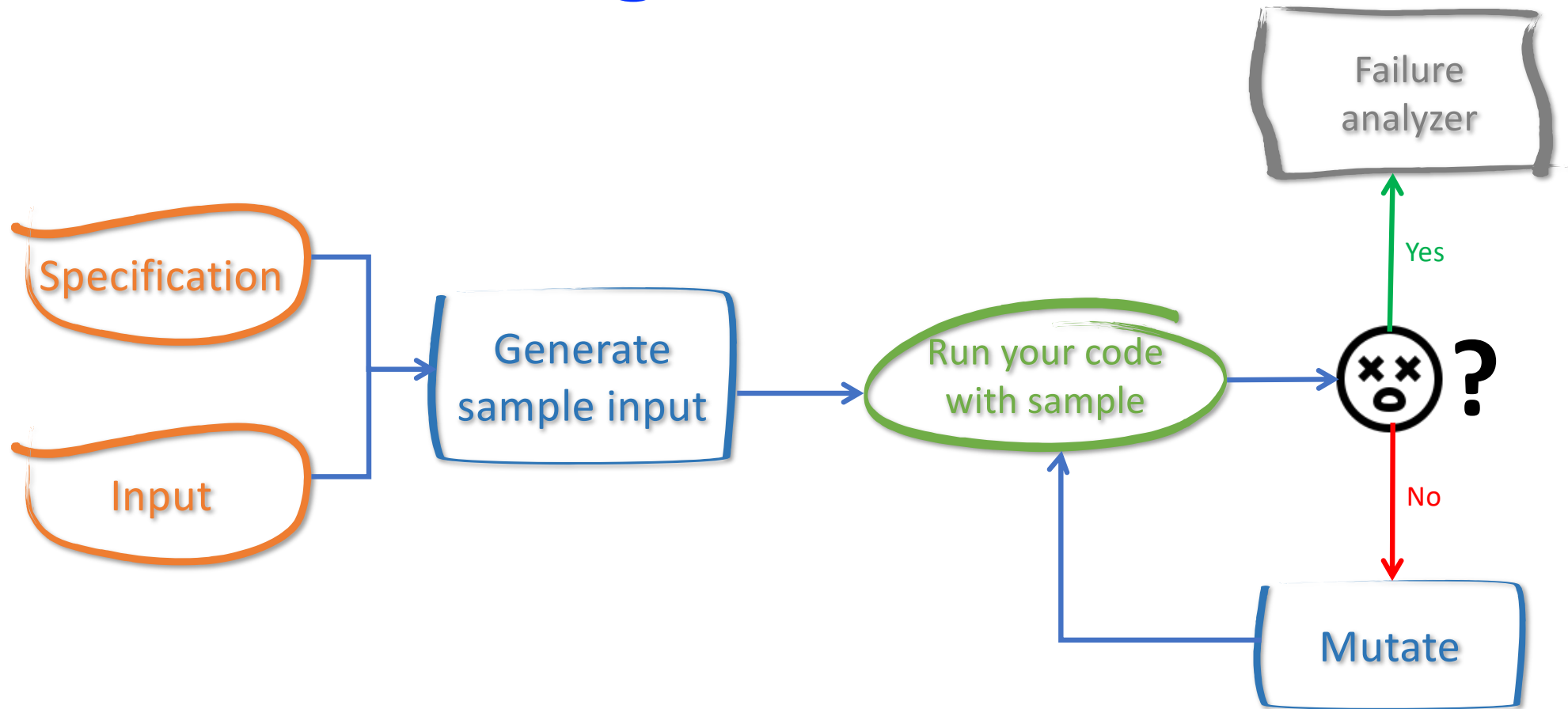
Generated data is invalid and abnormal

It tries to “hack” your code

It catches system crashes, built-in code failures, unhandled exception

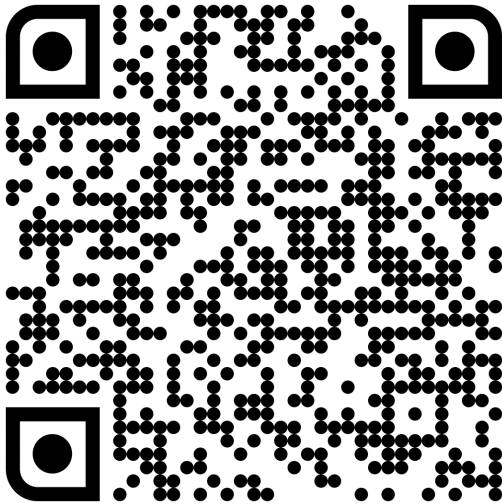
 Doesn't replace usual tests (unit, functional, integrations, etc)

What is fuzzing?



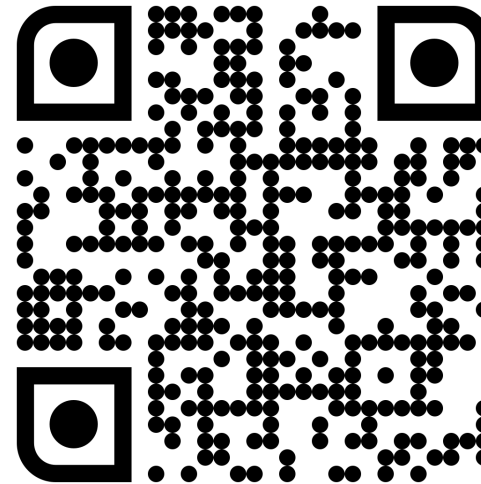
Step by step instruction

d3rky/pyday2022-bcn/blob/main/docs/instruction.md



Instruction

d3rky/pyday2022-bcn



Repo

What is Atheris?



Graybox Coverage Guided mutation-based fuzzer

Graybox Coverage Guided mutation-based fuzzer

- 📍 Atheris tracks the execution flows in the fuzzed program

Graybox Coverage Guided mutation-based fuzzer

Atheris tracks the execution flows in the fuzzed program

- 📍 Atheris uses code coverage to track execution flows

Graybox Coverage Guided mutation-based fuzzer

Atheris tracks the execution flows in the fuzzed program

Atheris uses code coverage to track execution flows

 Atheris generates new samples using mutations

Graybox Coverage Guided mutation-based fuzzer

Atheris tracks the execution flows in the fuzzed program

Atheris uses code coverage to track execution flows

- 📍 Atheris generates new samples using mutations
 - ✓ bit flips

Graybox Coverage Guided mutation-based fuzzer

Atheris tracks the execution flows in the fuzzed program

Atheris uses code coverage to track execution flows

 Atheris generates new samples using mutations

- ✓ bit flips
- ✓ byte copies

Graybox Coverage Guided mutation-based fuzzer

Atheris tracks the execution flows in the fuzzed program

Atheris uses code coverage to track execution flows

 Atheris generates new samples using mutations

- ✓ bit flips
- ✓ byte copies
- ✓ byte removals, etc

Atheris's features



 Based on the LLVM [libfuzzer](#) library

Atheris's features



Based on the LLVM [libfuzzer](#) library

 You can write [fuzzing tests](#) on Python

Atheris's features



Based on the LLVM [libfuzzer](#) library

You can write [fuzzing tests](#) on Python

 You can write your own [mutators](#) on Python

Atheris's features



Based on the LLVM [libfuzzer](#) library

You can write [fuzzing tests](#) on Python

You can write your own [mutators](#) on Python

 Atheris can track and analyze [regex](#) and [C-dependencies](#)

Let's fuzz!

**Fuzzing looks
familiar, doesn't it?**

Homework:

Fuzzing vs Property-based tests

What about the REST API?

Let's fuzz the REST API!

**What other mutations you can use
for JSON?**

Other mutations



- 📍 Generate the field with more length than expected

Other mutations



Generate the field with more length than expected

 Add, drop the field

Other mutations



Generate the field with more length than expected

Add, drop the field

 Move the branch of JSON to another position

Other mutations



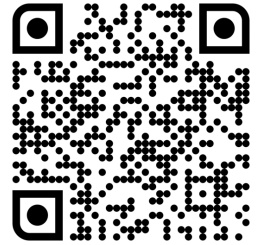
Generate the field with more length than expected

Add, drop the field

Move the branch of JSON to another position

 Replace the type of the property

Restler Fuzzer



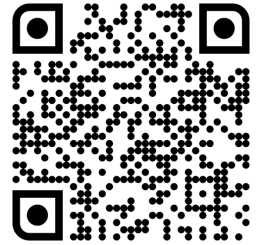
microsoft/restler-fuzzer

 Is created specifically for fuzzing the API

Consumes the OpenAPI specification to generate samples

Supports chaining of the HTTP requests

Restler Fuzzer



microsoft/restler-fuzzer

Is created specifically for fuzzing the API

 Consumes the OpenAPI specification to generate samples

Supports chaining of the HTTP requests

Restler Fuzzer



microsoft/restler-fuzzer

Is created specifically for fuzzing the API

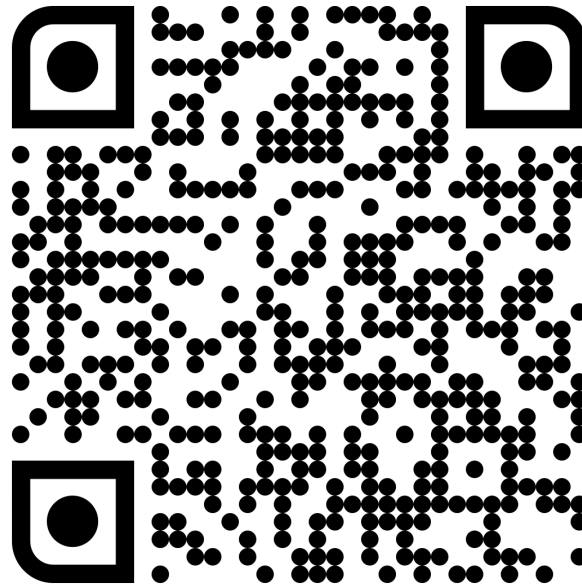
Consumes the OpenAPI specification to generate samples

 Supports chaining of the HTTP requests

Restler Fuzzer: homework



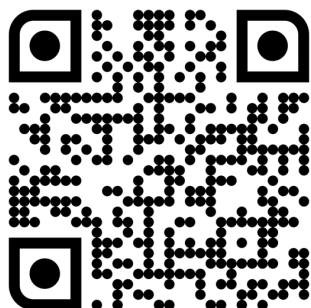
[microsoft/restler-fuzzer](#)



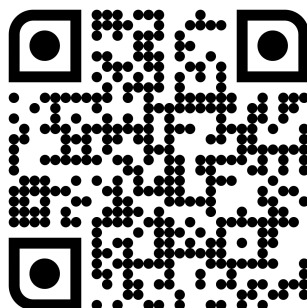
[wkoszolko/restler-fuzzer-getting-started](#)

Results

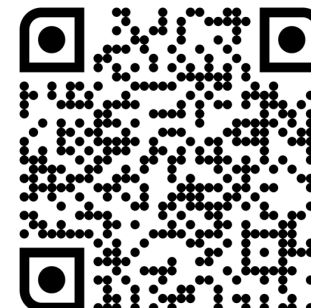
- 📍 Fuzzing is not so complex
- 📍 Fuzzing can find “hidden” bugs
- 📍 Fuzzing generates millions of samples automatically
- 📍 But for efficiency it requires configuration



google/atheris



d3rky/pyday2022-bcn



microsoft/restler-fuzzer



<https://www.linkedin.com/in/plonkin>



@pavel_lonkin



@d3rky