Fuzz Testing

Michael Van Sickle

@vansimke

Introduction

What is Fuzzing?

How do Fuzz Tests Work?

Creating and Running Fuzz Tests

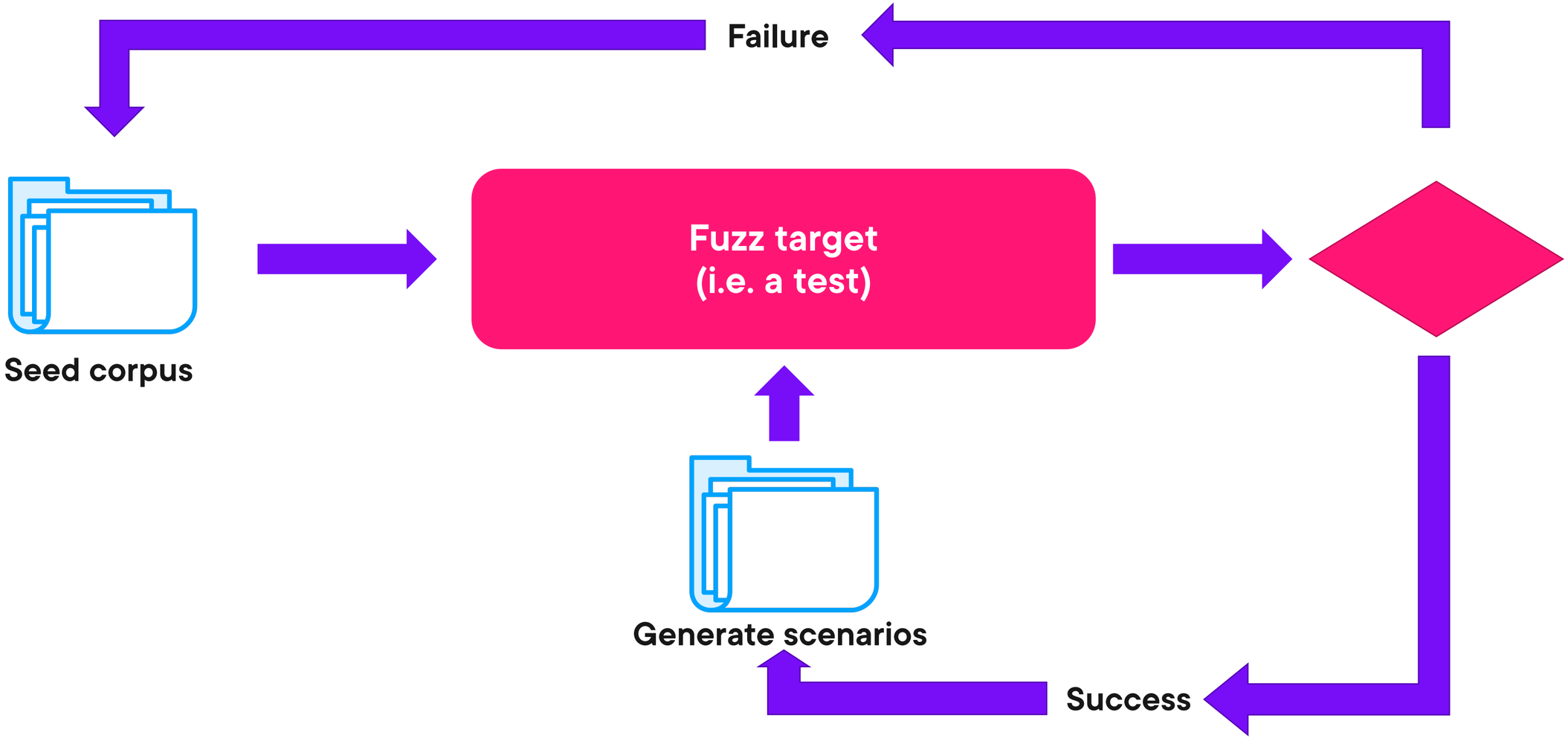
Configuring Fuzz Test Runs

Fuzzing

In programming and software development, fuzzing or fuzz testing is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program.

https://en.wikipedia.org/wiki/Fuzzing

How do Fuzz Tests Work?



Writing a fuzz test

Running fuzz tests

- components
  - f.Fuzz – req'd Fuzz target
  - f.Add – optional add to seed corpus
- as a unit test
- as a fuzz test
  - seeks scenarios that expand coverage ("interesting")

Working with failed scenarios

Configuring fuzz tests

- -fuzztime
- -fuzzminimizetime

```
func FuzzFoo(f *testing.F) {

    f.Add(...args)

    f.Fuzz(func(t *testing.T, ...args) {
        // arrange
        // act
        // assert
    })
}
```

- Prefix test with “Fuzz”
- Add arguments in order they should be passed to fuzz test
  - string, []byte
  - int, int8, int16, int32, int64
  - uint, uint8, uint16, uint32, uint64
  - float32, float64
  - bool
- One and only one f.Fuzz per test
- Tests run in parallel – don’t test shared memory!
- Arguments controlled by fuzzing engine
- Assertions typically made against arguments

```
go test -fuzz regexp

go test -fuzz regexp -fuzztime 30s
go test -fuzz regexp -fuzzminimizetime 30s
```

- Run fuzz tests matching regular expression
- Failed tests stored in ./testdata/fuzz/{FuzzTestName}
- Set the max test time (default: infinite)
- Set the max failure optimization time (default: 60s)