

Tutorial: Fuzz Testing via AFL++

Foundations of Cybersecurity

Giacomo Benedetti

Institute for Applied Mathematics and Information Technologies

National Research Council of Italy

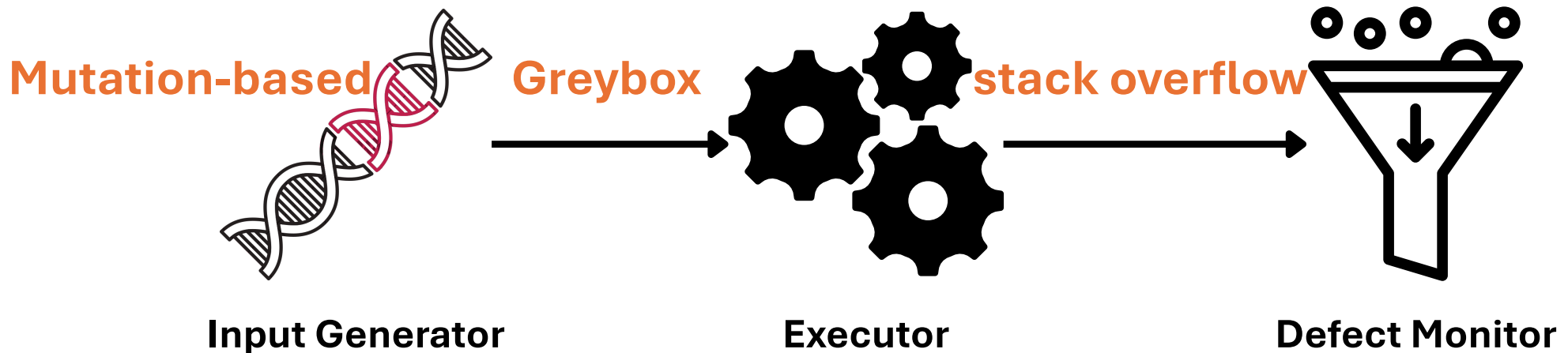
giacomo.benedetti@ge.imati.cnr.it



University of Pavia – Department of Electrical, Computer and Biomedical Engineering

Tutorial: Fuzzing xpdf

- Step 1: getting the AFL++ fuzzer and some sample inputs
- Step 1.b: install afl++ ()
- Step 2: instrumenting the target program
- Step 3: running afl++ against the instrumented program



Tutorial: Fuzzing xpdf

- Step 1: getting the program...

```
wget https://dl.xpdfreader.com/old/xpdf-3.02.tar.gz  
tar -xvzf xpdf-3.02.tar.gz
```

...and get some sample inputs

```
cd $HOME/fuzzing_xpdf mkdir pdf_examples && cd pdf_examples wget  
https://github.com/mozilla/pdf.js-sample-files/raw/master/helloworld.pdf  
wget http://www.africau.edu/images/default/sample.pdf wget  
https://www.melbpc.org.au/wp-content/uploads/2017/10/small-example-pdf-  
file.pdf
```

Tutorial: Fuzzing xpdf

- Step 1.b: install afl++ ()

Install dependencies

```
sudo apt-get update
sudo apt-get install -y build-essential python3-dev automake git flex bison
libglib2.0-dev libpixmap-1-dev python3-setuptools
sudo apt-get install -y lld-11 llvm-11 llvm-11-dev clang-11 || sudo apt-get
install -y lld llvm llvm-dev clang
sudo apt-get install -y gcc-$(gcc --version|head -n1|sed 's/.* //'|sed
's/\..*//')-plugin-dev libstdc++-$(gcc --version|head -n1|sed 's/.* //'|sed
's/\..*//')-dev
```

Tutorial: Fuzzing xpdf

- Step 1.b: install afl++ ()

Install afl

```
cd $HOME
git clone https://github.com/AFLplusplus/AFLplusplus && cd AFLplusplus
export LLVM_CONFIG="llvm-config-11"
make distrib
sudo make install
```

Check that the fuzzer is installed

```
afl-fuzz
```

Tutorial: Fuzzing xpdf

- Step 2: instrumenting the program

We use afl-clang-fast: one of the compiler provided from this fuzzer to instrument a binary

```
rm -r $HOME/fuzzing_xpdf/install
cd $HOME/fuzzing_xpdf/xpdf-3.02/

export LLVM_CONFIG="llvm-config-11"
CC=$HOME/AFLplusplus/afl-clang-fast CXX=$HOME/AFLplusplus/afl-clang-fast++
./configure --prefix="$HOME/fuzzing_xpdf/install/"
make
make install
```

Tutorial: Fuzzing xpdf

- Step 3: running afl++ against the instrumented program

afl-fuzz

```
-i $HOME/fuzzing_xpdf/pdf_examples/  
-o $HOME/fuzzing_xpdf/out/  
-s 123  
-- $HOME/fuzzing_xpdf/install/bin/pdftotext @@ $HOME/fuzzing_xpdf/output
```

directory containing the input cases (the sample PDF files)

Tutorial: Fuzzing xpdf

- Step 3: running afl++ against the instrumented program

afl-fuzz

-i \$HOME/fuzzing_xpdf/pdf_examples/

-o \$HOME/fuzzing_xpdf/out/

-s 123

-- \$HOME/fuzzing_xpdf/install/bin/pdftotext @@ \$HOME/fuzzing_xpdf/output

directory where the mutate files will be
stored

Tutorial: Fuzzing xpdf

- Step 3: running afl++ against the instrumented program

```
afl-fuzz
```

```
-i $HOME/fuzzing_xpdf/pdf_examples/
```

```
-o $HOME/fuzzing_xpdf/out/
```

```
-s 123
```

```
-- $HOME/fuzzing_xpdf/install/bin/pdftotext @@ $HOME/fuzzing_xpdf/output
```

The static random seed to use. The files
will be mutated accordingly to this seed.

Tutorial: Fuzzing xpdf

- Step 3: running afl++ against the instrumented program

```
afl-fuzz
```

```
-i $HOME/fuzzing_xpdf/pdf_examples/
```

```
-o $HOME/fuzzing_xpdf/out/
```

```
-s 123
```

```
-- $HOME/fuzzing_xpdf/install/bin/pdftotext @@ $HOME/fuzzing_xpdf/output
```


The instrumented binary that is our
program under test

Tutorial: Fuzzing xpdf

- Step 3: running afl++ against the instrumented program

afl-fuzz

```
-i $HOME/fuzzing_xpdf/pdf_examples/  
-o $HOME/fuzzing_xpdf/out/  
-s 123  
-- $HOME/fuzzing_xpdf/install/bin/pdftotext @@ $HOME/fuzzing_xpdf/output
```



placeholder that will be substituted with
the name of the mutated input