

CSE 543: Information Assurance and Security

Name: Mohammed Ahmed Ragab

university ID: 122*****

university: Arizona state university (ASU)

project name: Fuzz them all

course: CSE 543 Information Assurance and Security

Description:

I developed a fuzzer software that generates random string with different length based on two inputs provided in the command line argument, which are seed value and number of iteration

how does my fuzzer program works ?

My fuzzer program takes two inputs from the command line , the first one is a seed value which is used to be given to any random value generator so that each time the random value generator generates the same output, the second input is the number of iterations which represents the length of the generates random string.

in python programming language, if you set a seed value in the “random” module(by calling `random.seed(seed_value)`) before calling any random function in “random” module such as “ `random.choice(letters)`”, the random function will generate the same value each time you run the python script, this is called “deterministic behavior”, I tested this by myself and commented the test code in my python fuzzer source code so that anyone can test it.

How is the random string generated?

in python , there is “string” module which contains a constant named “*printable*”, you can use it by typing “*string.printable*” which contains all printable ASCII characters including upper and lower English letters, digits(0 - 9), and special characters(\$#@..etc) and space characters.

In addition, there is a random function” *random.choice(letters)*” which chooses a random character from a string(i.e “*string.printable*” constant) , this function is called in number of iterations(giving in the command line argument) so that a random string is built from adding joining(adding) characters to an empty string.

Finally, the string is printed to the standard output (i.e the terminal). In Linux you can pipeline the output of the fuzzer to one of the prog_x where x is the program number you want to crash.

What is the command did I use for crashing the programs?

```
python ./fuzzer.py [seed value] [num of iterations] | ./all_test_programs/prog_x
```

where x is the program number you want to crash

for example:

```
python ./fuzzer.py 2 100 | ./all_test_programs/prog_0
```

```
python ./fuzzer.py 2 100 | ./all_test_programs/prog_1
```

```
python ./fuzzer.py 2 300 | ./all_test_programs/prog_2
```

```
python ./fuzzer.py 2 300 | ./all_test_programs/prog_3
```

```
python ./fuzzer.py 2 300 | ./all_test_programs/prog_4
```

```
python ./fuzzer.py 2 300 | ./all_test_programs/prog_5
```

```
python ./fuzzer.py 2 300 | ./all_test_programs/prog_6
```

```
python ./fuzzer.py 2 300 | ./all_test_programs/prog_7
```

```
python ./fuzzer.py 3 300 | ./all_test_programs/prog_9
```

if “Segmentation fault (core dumped)” is printed on the terminal, then the program has been crashed by the fuzzer.

Environment:

- operating system : Linux Ubuntu 20.04 LTS run on virtual machine using VMware workstation 16 player on windows 10
- Python 3.8.10
- text editor: VS code

Screenshots:

The image shows a Visual Studio Code window with a terminal open. The title bar at the top reads "fuzzer.py - fuzzer - Visual Studio Code". The terminal interface has tabs for "PROBLEMS", "DEBUG CONSOLE", "OUTPUT", and "TERMINAL", with "TERMINAL" being the active tab. The terminal shows a series of commands and outputs from a shell prompt "oha ed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer\$". The commands are: 1. `python ./fuzzer.py 2 100 | ./all_test_programs/prog_0` followed by "Segmentation fault (core dumped)". 2. `python ./fuzzer.py 2 100 | ./all_test_programs/prog_1` followed by "Segmentation fault (core dumped)". 3. `python ./fuzzer.py 2 300 | ./all_test_programs/prog_2` followed by "Segmentation fault (core dumped)". 4. `python ./fuzzer.py 2 300 | ./all_test_programs/prog_3` followed by "Segmentation fault (core dumped)". 5. `python ./fuzzer.py 2 300 | ./all_test_programs/prog_4` followed by "Segmentation fault (core dumped)". 6. `python ./fuzzer.py 2 300 | ./all_test_programs/prog_5` followed by "Segmentation fault (core dumped)". 7. `python ./fuzzer.py 2 300 | ./all_test_programs/prog_6` followed by "Segmentation fault (core dumped)". 8. `python ./fuzzer.py 2 300 | ./all_test_programs/prog_7` followed by "Segmentation fault (core dumped)". 9. `python ./fuzzer.py 2 300 | ./all_test_programs/prog_9` followed by "Segmentation fault (core dumped)". 10. After a carriage return (^C), the command `python ./fuzzer.py 3 300 | ./all_test_programs/prog_9` is entered, followed by "Segmentation fault (core dumped)". The prompt "oha ed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer\$" is shown at the bottom. The status bar at the very bottom indicates "Ln 17, Col 13", "Spaces: 4", "UTF-8", "CRLF", "Python", "3.8.10 64-bit", "kite: ready", and a search icon.

This screenshot shows the Visual Studio Code editor with the file `fuZZer.py` open. The Explorer sidebar on the left shows a project structure with folders `all_test_programs` and `test_programs`, and a file `fuZZer.py`. The code in `fuZZer.py` contains comments indicating that programs `prog_0` through `prog_9` were crashed with seed 2 and various iteration counts (100 or 300). Program 8 is noted as being uncrashable, always returning the output "Reading too much!".

The terminal at the bottom shows a series of commands and their outputs:

```
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 100 | ./all_test_programs/prog_0
Segmentation fault (core dumped)
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 100 | ./all_test_programs/prog_1
Segmentation fault (core dumped)
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 300 | ./all_test_programs/prog_2
Segmentation fault (core dumped)
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 300 | ./all_test_programs/prog_3
Segmentation fault (core dumped)
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 300 | ./all_test_programs/prog_4
Segmentation fault (core dumped)
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 300 | ./all_test_programs/prog_5
```

This screenshot continues the terminal output from the previous one, showing further test runs of the programs with different seeds and iteration counts:

```
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 300 | ./all_test_programs/prog_5
Segmentation fault (core dumped)
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 300 | ./all_test_programs/prog_6
Segmentation fault (core dumped)
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 300 | ./all_test_programs/prog_7
Segmentation fault (core dumped)
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 2 300 | ./all_test_programs/prog_9
^C
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$ python ./fuZZer.py 3 300 | ./all_test_programs/prog_9
Segmentation fault (core dumped)
mohammed@ubuntu:~/Desktop/CSE 543/fuzzThemAll/fuzzer$
```