Programming Project 2: Fuzz Testing

-Sridevi Divya Krishna Devisetty(4436572)

Goal:

To implement mutation-based fuzzer or fuzz tester. Fuzz testing is one way of discovering security vulnerabilities in any code that processes potentially malicious input.

A mutation-based fuzzer takes a valid input, in our case cross.jpg file and mutates the input by making random changes and generates new test cases.

Our goal is to track down few of the bugs introduced in jpg2bmp executable by mutating the input image and feeding it to jpg2bmp and saving the images that triggered specific bugs.

Program Approach:

Analysis: Given a 1kb input image cross.jpg. We need to mutate the image's byte info such that we could trigger the manually introduced bugs in jpg2bmp exe file, when we use mutated image as its input. Mutation can be done by changing byte values to random values using rand() function in c.

Design and Implementation:

Programming Language: C

Libraries: stdio.h , stdlib.h, string.h, unistd.h, sys/wait.h

In this project, the input image is read to input buffer and byte value mutations are carried out in 2000,4000,10000 iterations to conclude the final results.

- Bugs 4, 7 and 8 and rarely 5 and 2 were easy to generate compared to 1 and 3.when ran for 10000 iterations by varying input_buff content at rand_index with rand_value.

- rand_index and rand_value are generated using rand()%800, which generates value between 1 to 800.

- But it takes 10000 iterations for bug-1 to appear, so I tracked down rand_val at which *bug-1 triggered, it was at index 48*, so I added an if statement accordingly

and changed iterations to 2000 also tested this with 4000 iterations which increased bug-3 count but bug-1 count stays at 1.

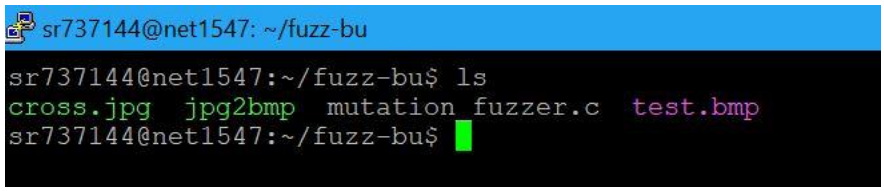
- The main reason to have various iterations is to check the least number of mutations at which all 7 bugs appear. The lesser the number of iteration lesser the memory usage and execution time.

Implementation Steps:

- Created two files one to read in the image info and other to write out the mutated files.
- In each iteration, We mutate the input image using rand() function.
- Once the input image is mutated it is then fed to jpg2bmp and the command is executed to run it to generate bmp files.
- If the system responds with a bug for the executed command then read the bug information and if it is a valid bug between 1-8 then increment corresponding bug counter and also write back the image test#bug_num to system.
- Thus the last mutated image which triggers the bug overwrites previously generated bug of same category, hence only one test-#bug_num file exists when the program finishes.
- We also keep track of number of segmentation faults occurred using crash-counter variable, this includes creation of images with name crashed-#crash_counter, these images also include bugs which are not part of bugs 1-8.
- Finally, we print the total number of times each bug that's triggered and also total_num of times segmentation fault has occurred (response_code 128+11,128+6 indicates segmentation fault).

Execution_Steps:

1) We place the fuzzer code, mutation_fuzzer.c and jpg2bmp exe file and cross.jpg in a directory and change permissions to jpg2bmp using
   $chmod u+x jpg2bmp



2) We compile our program
   gcc –o mut_fuz mutation_fuzzer.c

```
sr737144@net1547: ~/fuzz-bu
sr737144@net1547:~/fuzz-bu$ ls
cross.jpg  jpg2bmp  mutation_fuzzer.c  test.bmp
sr737144@net1547:~/fuzz-bu$ gcc -o mut_fuz mutation_fuzzer.c
```

3) We execute the program and redirect output logs to output.txt:
   ./mut_fuz
   (OR)
   ./mut_fuz >ouput.txt (to redirect logs to text file)

```
sr737144@net1547: ~/fuzz-bu
sr737144@net1547:~/fuzz-bu$ ls
cross.jpg  jpg2bmp  mutation_fuzzer.c  test.bmp
sr737144@net1547:~/fuzz-bu$ gcc -o mut_fuz mutation_fuzzer.c
sr737144@net1547:~/fuzz-bu$ ls
cross.jpg  jpg2bmp  mutation_fuzzer.c  mut_fuz  test.bmp
sr737144@net1547:~/fuzz-bu$ ./mut_fuz >output.txt
```

4) If the crash-file creation lines are uncommented we need to do following step to remove them.
   rm crashed*.jpg

```
sr737144@net1547: ~/fuzz-bu
Segmentation fault (core dumped)
crashed image : ./crashed-164.jpg
Bug #4 triggered.
Segmentation fault (core dumped)
crashed image : ./crashed-165.jpg
Bug #2 triggered.
Segmentation fault (core dumped)
crashed image : ./crashed-166.jpg
Bug #8 triggered.
Segmentation fault (core dumped)
crashed image : ./crashed-167.jpg
Bug #8 triggered.
Segmentation fault (core dumped)
crashed image : ./crashed-168.jpg
Bug #8 triggered.
Segmentation fault (core dumped)
crashed image : ./crashed-169.jpg
sr737144@net1547:~/fuzz-bu$ ls
crashed-100.jpg  crashed-11.jpg   crashed-139.jpg  crashed-158.jpg  crashed-23.jpg  crashed-42.jpg  crashed-61.jpg  crashed-80.jpg  crashed-9.jpg
crashed-101.jpg  crashed-120.jpg  crashed-13.jpg   crashed-159.jpg  crashed-24.jpg  crashed-43.jpg  crashed-62.jpg  crashed-81.jpg  cross.jpg
crashed-102.jpg  crashed-121.jpg  crashed-140.jpg  crashed-15.jpg   crashed-25.jpg  crashed-44.jpg  crashed-63.jpg  crashed-82.jpg  jpg2bmp
crashed-103.jpg  crashed-122.jpg  crashed-141.jpg  crashed-160.jpg  crashed-26.jpg  crashed-45.jpg  crashed-64.jpg  crashed-83.jpg  mutation_fuzzer.c
crashed-104.jpg  crashed-123.jpg  crashed-142.jpg  crashed-161.jpg  crashed-27.jpg  crashed-46.jpg  crashed-65.jpg  crashed-84.jpg  mut_fuz
crashed-105.jpg  crashed-124.jpg  crashed-143.jpg  crashed-162.jpg  crashed-28.jpg  crashed-47.jpg  crashed-66.jpg  crashed-85.jpg  output.txt
crashed-106.jpg  crashed-125.jpg  crashed-144.jpg  crashed-163.jpg  crashed-29.jpg  crashed-48.jpg  crashed-67.jpg  crashed-86.jpg  test-1.jpg
crashed-107.jpg  crashed-126.jpg  crashed-145.jpg  crashed-164.jpg  crashed-2.jpg   crashed-49.jpg  crashed-68.jpg  crashed-87.jpg  test-2.jpg
crashed-108.jpg  crashed-127.jpg  crashed-146.jpg  crashed-165.jpg  crashed-30.jpg  crashed-4.jpg   crashed-69.jpg  crashed-88.jpg  test-3.jpg
crashed-109.jpg  crashed-128.jpg  crashed-147.jpg  crashed-166.jpg  crashed-31.jpg  crashed-50.jpg  crashed-6.jpg   crashed-89.jpg  test-4.jpg
crashed-10.jpg   crashed-129.jpg  crashed-148.jpg  crashed-167.jpg  crashed-32.jpg  crashed-51.jpg  crashed-70.jpg  crashed-8.jpg   test-5.jpg
crashed-110.jpg  crashed-12.jpg   crashed-149.jpg  crashed-168.jpg  crashed-33.jpg  crashed-52.jpg  crashed-71.jpg  crashed-90.jpg  test-7.jpg
crashed-111.jpg  crashed-130.jpg  crashed-14.jpg   crashed-169.jpg  crashed-34.jpg  crashed-53.jpg  crashed-72.jpg  crashed-91.jpg  test-8.jpg
crashed-112.jpg  crashed-131.jpg  crashed-150.jpg  crashed-16.jpg   crashed-35.jpg  crashed-54.jpg  crashed-73.jpg  crashed-92.jpg  test.bmp
crashed-113.jpg  crashed-132.jpg  crashed-151.jpg  crashed-17.jpg   crashed-36.jpg  crashed-55.jpg  crashed-74.jpg  crashed-93.jpg  test.jpg
crashed-114.jpg  crashed-133.jpg  crashed-152.jpg  crashed-18.jpg   crashed-37.jpg  crashed-56.jpg  crashed-75.jpg  crashed-94.jpg
crashed-115.jpg  crashed-134.jpg  crashed-153.jpg  crashed-19.jpg   crashed-38.jpg  crashed-57.jpg  crashed-76.jpg  crashed-95.jpg
crashed-116.jpg  crashed-135.jpg  crashed-154.jpg  crashed-1.jpg    crashed-39.jpg  crashed-58.jpg  crashed-77.jpg  crashed-96.jpg
crashed-117.jpg  crashed-136.jpg  crashed-155.jpg  crashed-20.jpg   crashed-3.jpg   crashed-59.jpg  crashed-78.jpg  crashed-97.jpg
crashed-118.jpg  crashed-137.jpg  crashed-156.jpg  crashed-21.jpg   crashed-40.jpg  crashed-5.jpg   crashed-79.jpg  crashed-98.jpg
crashed-119.jpg  crashed-138.jpg  crashed-157.jpg  crashed-22.jpg   crashed-41.jpg  crashed-60.jpg  crashed-7.jpg   crashed-99.jpg
sr737144@net1547:~/fuzz-bu$ rm crashed*.jpg
sr737144@net1547:~/fuzz-bu$ ls
cross.jpg  mutation_fuzzer.c  output.txt  test-2.jpg  test-4.jpg  test-7.jpg  test.bmp
jpg2bmp    mut_fuz            test-1.jpg  test-3.jpg  test-5.jpg  test-8.jpg  test.jpg
sr737144@net1547:~/fuzz-bu$
```

Test Results:

Use command :

./jpg2bmp test-#.jpg test-out.bmp (# = 1 to 8)

```
sr737144@net1547: ~/fuzz-bu
sr737144@net1547:~/fuzz-bu$ ./jpg2bmp test-1.jpg test-out.bmp
Bug #1 triggered.
Segmentation fault (core dumped)
sr737144@net1547:~/fuzz-bu$ ./jpg2bmp test-2.jpg test-out.bmp
Bug #2 triggered.
Segmentation fault (core dumped)
sr737144@net1547:~/fuzz-bu$ ./jpg2bmp test-3.jpg test-out.bmp
Bug #3 triggered.
Segmentation fault (core dumped)
sr737144@net1547:~/fuzz-bu$ ./jpg2bmp test-4.jpg test-out.bmp
Bug #4 triggered.
Segmentation fault (core dumped)
sr737144@net1547:~/fuzz-bu$ ./jpg2bmp test-5.jpg test-out.bmp
Bug #5 triggered.
Segmentation fault (core dumped)
sr737144@net1547:~/fuzz-bu$ ./jpg2bmp test-7.jpg test-out.bmp
Bug #7 triggered.
Segmentation fault (core dumped)
sr737144@net1547:~/fuzz-bu$ ./jpg2bmp test-8.jpg test-out.bmp
Bug #8 triggered.
Segmentation fault (core dumped)
sr737144@net1547:~/fuzz-bu$
```

Empirical Results:

Total Number of Bugs that could be triggered: 7

Bug Number	Image File that triggered the bug
Bug #1	test-1.jpg
Bug #2	test-2.jpg
Bug #3	test-3.jpg
Bug #4	test-4.jpg
Bug #5	test-5.jpg
Bug #7	test-7.jpg
Bug #8	test-8.jpg

Bug Number	**No. of bug occurrences(2000 iterations)**	**No. of bug occurrences(4000 iterations)**	**No. of bug occurrences(10000 iterations)**
Bug #1	1	1	1
Bug #2	8	10	29
Bug #3	1	4	8
Bug #4	58	131	319
Bug #5	9	22	58
Bug #7	18	40	89
Bug #8	63	138	348
Total Bug occurrences	**158**	**346**	**852**
Total Segmentation Faults(includes Unqualified bugs)	**169**	**369**	**908**

Attaching images for reference for 2000, 4000 and 10000 iterations respectively :

sr737144@net1547: ~/fuzz-bu

```
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #7 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #2 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)

169
 Total number of times Bug 1 occurred: 1
 Total number of times Bug 2 occurred: 8
 Total number of times Bug 3 occurred: 1
 Total number of times Bug 4 occurred: 58
 Total number of times Bug 5 occurred: 9
 Total number of times Bug 6 occurred: 0
 Total number of times Bug 7 occurred: 18
 Total number of times Bug 8 occurred: 63
sr737144@net1547:~/fuzz-bu$
```

sr737144@net1547: ~/fuzz-bu

```
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #7 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #5 triggered.
Segmentation fault (core dumped)
Bug #5 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #7 triggered.
Segmentation fault (core dumped)
Bug #5 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Floating point exception (core dumped)
Bug #5 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Floating point exception (core dumped)

369
 Total number of times Bug 1 occurred: 1
 Total number of times Bug 2 occurred: 10
 Total number of times Bug 3 occurred: 4
 Total number of times Bug 4 occurred: 131
 Total number of times Bug 5 occurred: 22
 Total number of times Bug 6 occurred: 0
 Total number of times Bug 7 occurred: 40
 Total number of times Bug 8 occurred: 138
sr737144@net1547:~/fuzz-bu$
```

sr737144@net1547: ~/fuzz-bu

```
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Floating point exception (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #3 triggered.
Segmentation fault (core dumped)
Bug #7 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)
Bug #2 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #8 triggered.
Segmentation fault (core dumped)
Bug #4 triggered.
Segmentation fault (core dumped)

908
 Total number of times Bug 1 occurred: 1
 Total number of times Bug 2 occurred: 29
 Total number of times Bug 3 occurred: 8
 Total number of times Bug 4 occurred: 319
 Total number of times Bug 5 occurred: 58
 Total number of times Bug 6 occurred: 0
 Total number of times Bug 7 occurred: 89
 Total number of times Bug 8 occurred: 348
sr737144@net1547:~/fuzz-bu$
```