SECTION 1 - Designing the mutation based Fuzzer

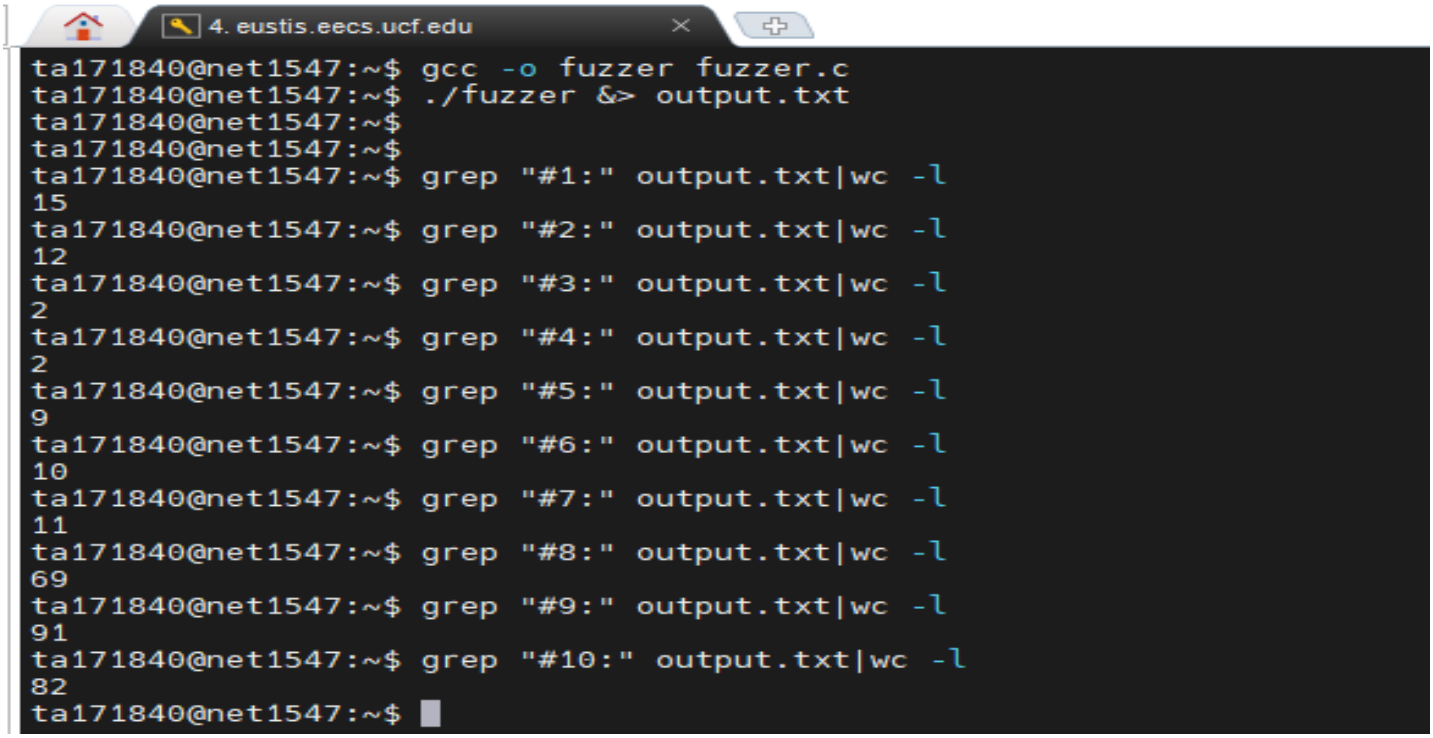
We designed a C program to implement this mutation based fuzzing test on the given jpeg file (cross.jpg) using mutation technique discussed below. Our primary task was to detect the bugs/crashes in the image conversion program given to us: "jpeg2bmp", by generating the mutated versions of input jpeg files and one by one checking for each of the 10 bugs during this conversion.

To run this source code please use these commands on Eustis machine:
gcc -o fuzzer fuzzer.c
./fuzzer

The steps and mutation rules that are being used in designing this fuzzer code are:

1. **Reading the file:**
    - Firstly, we read the content of our original jpeg file ("cross.jpg") into a character array named buffer.
2. **Performing mutation in the file:**
    - After reading is done, then we initiate a loop for 1000 iterations and perform the mutations on the buffer based on these 3 conditions:
        - For the very first 100 iterations, using rand() method we randomly select a position in the buffer and replace the byte at that position with a random value.
        - And for the iterations between 100 to 500, we randomly select a position range (5 bytes in our case) in the buffer and replace each byte in this range with a random value obtained using rand() function.
        - Lastly, for iterations after 500 till 1000, we randomly select a position in the buffer and replace the byte at that position with a random value again using rand() function. We repeat this process 6 times using a for loop.
3. **Writing to the File:**
    - After we performed each mutation, each time the mutated buffer was written to a new jpeg file (cross_muted.jpg).
4. **Performing Conversion:**
    - Then we execute the "jpeg2bmp" on the mutated jpeg file using the system command. The output (return code) of this command is checked for crashes with exit code 134 or 139 for Segmentation fault(11) and Abort(6) respectively.
5. **Finally detecting the Bugs:**
    - Whenever a crash is detected by exit codes 134 or 139, we extract the bug number from the output of jpeg conversion performed for that file.
    - We write the mutated buffer to a new jpeg file named "test-n.jpg" , where n is the Bug which got triggered at that time.
    - Finally, we print the information about the detected return code 134/139 and the name of the generated test-n image file with this crash.

SECTION 2 - Experimental Results



Screenshot showing usage of grep command to count the number of times each bug triggered by the Fuzzer.

	Bug #1	Bug #2	Bug #3	Bug #4	Bug #5	Bug #6	Bug #7	Bug #8	Bug #9	Bug #10	Overall
# of crashes triggered	15	12	2	2	9	10	11	69	91	82	303

Fig1: Table shows the statistical results of fuzz testing

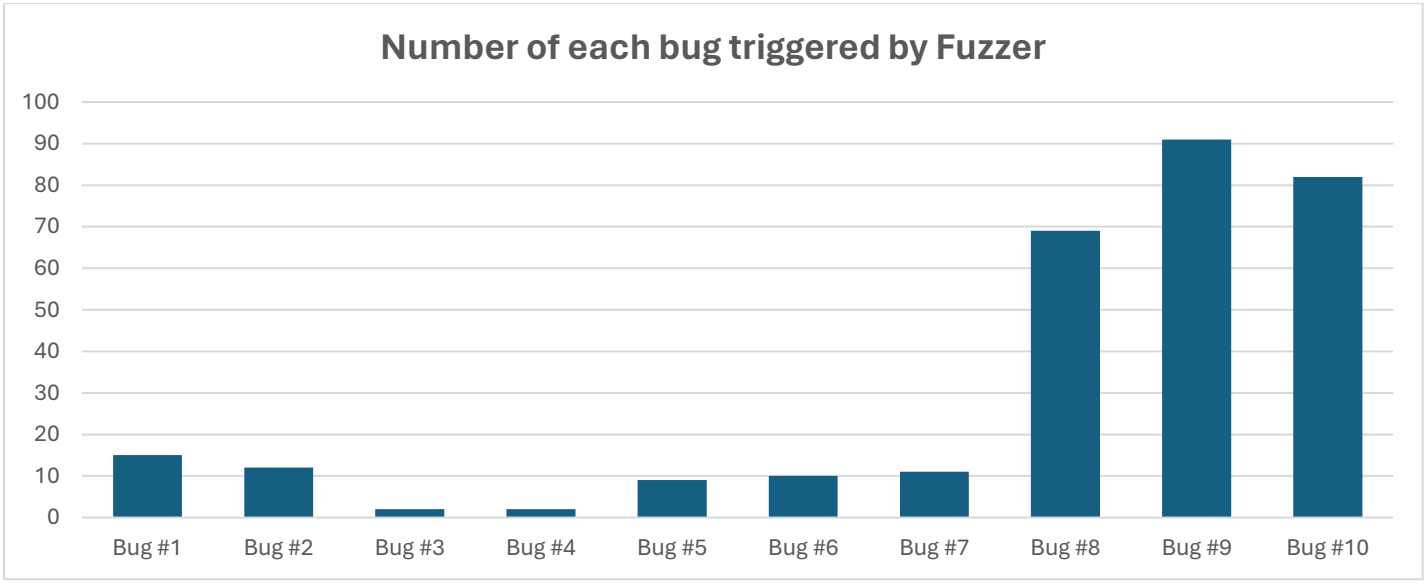


Fig2: Bar chart showing number of times the bugs triggered by Fuzzer

SECTION 3 - Proof of Finding Bugs

Running the jpeg2bmp command for each of the 10 muted image files to demonstrate that it can trigger the 10 bugs found by the Fuzzer.

```
ta171840@net1547:~$
ta171840@net1547:~$ ./jpeg2bmp test-1.jpg temp.bmp
Bug #1: Quantization Table checking - 16 bits
Segmentation fault (core dumped)
ta171840@net1547:~$ ./jpeg2bmp test-2.jpg temp.bmp
Bug #2: More than 2 AC Huffman Tables
Segmentation fault (core dumped)
ta171840@net1547:~$ ./jpeg2bmp test-3.jpg temp.bmp
Bug #3: Only support YCbCr image
Segmentation fault (core dumped)
ta171840@net1547:~$ ./jpeg2bmp test-4.jpg temp.bmp
Bug #4: DC Huffman Table - more than 2
Segmentation fault (core dumped)
ta171840@net1547:~$ ./jpeg2bmp test-5.jpg temp.bmp
Bug #5: too many Huffman tables
Segmentation fault (core dumped)
ta171840@net1547:~$ ./jpeg2bmp test-6.jpg temp.bmp
Bug #6: Not a JPG file
Segmentation fault (core dumped)
ta171840@net1547:~$ ./jpeg2bmp test-7.jpg temp.bmp
Bug #7: Huffman code not found
Segmentation fault (core dumped)
ta171840@net1547:~$ ./jpeg2bmp test-8.jpg temp.bmp
Bug #8: Huffman decoding error
Segmentation fault (core dumped)
ta171840@net1547:~$ ./jpeg2bmp test-9.jpg temp.bmp
Bug #9: More than 1KByte for Huffman Table
Segmentation fault (core dumped)
ta171840@net1547:~$ ./jpeg2bmp test-10.jpg temp.bmp
Bug #10: Bogus JPEG format
Segmentation fault (core dumped)
ta171840@net1547:~$
```