IDATT2503 - Exercise 05 - Fuzzing

The assignment:

> Perform fuzzing with address sanitizer on the C function you created in exercise 4-2.
>
> Fix bugs you find through fuzzing, or introduce bugs that are discovered through fuzzing

Write the fuzzy code.

Build with:

```
mkdir build
cd build
CC=clang cmake ..
make
```

Run with:

```
./tests/escape_fuzzer_test -max_total_time=60
```

Output of the original code:

```
INFO: Running with entropic power schedule (0xFF, 100).
INFO: Seed: 464668237
INFO: Loaded 1 modules   (3 inline 8-bit counters): 3 [0x588d6972fea8, 0x588d6972feab),
INFO: Loaded 1 PC tables (3 PCs): 3 [0x588d6972feb0,0x588d6972fee0),
INFO: -max_len is not provided; libFuzzer will not generate inputs larger than 4096 bytes
INFO: A corpus is not provided, starting from an empty corpus
#2      INITED cov: 2 ft: 2 corp: 1/1b exec/s: 0 rss: 32Mb
#4194304        pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1398101 rss: 523Mb
#8388608        pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1398101 rss: 525Mb
#16777216       pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1290555 rss: 525Mb
#33554432       pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1290555 rss: 526Mb
#67108864       pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1266204 rss: 527Mb
#76587767       DONE   cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1255537 rss: 527Mb
Done 76587767 runs in 61 second(s)
```

Apparently no issues here. Tried introducing bugs, or complexity, by adding more special characters to the escaping, like `"` (`&quot;`) and `\` (`&#39;`), but this was really just more of the same.

It struck me that there might be a possibility for a loop of never-ending escaping. If we have `&` which is replaced (escaped) by `&amp;`, the replacement string itself contains the character being espaced. It would go like this:

1. First pass: `&` → `&amp;`
2. Second pass: `&` (from `&`) → `&amp;`
3. Third pass: `&` (from `&amp;`) → `&amp;amp;`
4. Und so weiter.

Which is what the Internet calls "Double Escaping". In the code, there are no safeguards against this, e.g. checking if a sequence have already been escaped, by deciding that if there is a string sequence `&amp` it's already escaped and will be skipped.

Added a `corpus` directory, containing three seed.txt files, to be able to pull specific test strings to trigger this. From project root directory:

```
echo "&amp;" > corpus/seed1.txt
echo "&lt;&gt;" > corpus/seed2.txt
echo "&lt;&amp;&gt;" > corpus/seed3.txt
```

Build and make the code once more.

```
cd build
CC=clang cmake ..
make
```

Then run the fuzzer with the corpus directory (from `/build`):

`./tests/escape_fuzzer_test ../corpus echo "&amp;" > ../corpus/seed1.txt`

First run was uneventfull:

```
INFO: Running with entropic power schedule (0xFF, 100).
INFO: Seed: 320584246
INFO: Loaded 1 modules   (3 inline 8-bit counters): 3 [0x65129f148ea8, 0x65129f148eab),
INFO: Loaded 1 PC tables (3 PCs): 3 [0x65129f148eb0,0x65129f148ee0),
INFO:        3 files found in ../corpus
INFO: -max_len is not provided; libFuzzer will not generate inputs larger than 4096 bytes
INFO: seed corpus: files: 3 min: 6b max: 14b total: 29b rss: 31Mb
#4      INITED cov: 2 ft: 2 corp: 1/6b exec/s: 0 rss: 32Mb
#5      REDUCE cov: 2 ft: 2 corp: 1/5b lim: 6 exec/s: 0 rss: 32Mb L: 5/5 MS: 1 EraseBytes-
#8      REDUCE cov: 2 ft: 2 corp: 1/4b lim: 6 exec/s: 0 rss: 32Mb L: 4/4 MS: 3
ChangeBinInt-ChangeBit-EraseBytes-
#12     REDUCE cov: 2 ft: 2 corp: 1/3b lim: 6 exec/s: 0 rss: 32Mb L: 3/3 MS: 4
ChangeBinInt-CrossOver-ChangeByte-EraseBytes-
#13     REDUCE cov: 2 ft: 2 corp: 1/2b lim: 6 exec/s: 0 rss: 32Mb L: 2/2 MS: 1 EraseBytes-
#15     REDUCE cov: 2 ft: 2 corp: 1/1b lim: 6 exec/s: 0 rss: 32Mb L: 1/1 MS: 2
ShuffleBytes-EraseBytes-
#4194304        pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1398101 rss: 570Mb
#8388608        pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1398101 rss: 571Mb
#16777216       pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1398101 rss: 572Mb
#33554432       pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1342177 rss: 573Mb
```

```
#67108864       pulse  cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1315860 rss: 574Mb
#79108525       DONE   cov: 2 ft: 2 corp: 1/1b lim: 4096 exec/s: 1296861 rss: 574Mb
Done 79108525 runs in 61 second(s)
```

Added more seed files :

```
echo "&amp;" > corpus/seed1.txt
echo "&lt;" > corpus/seed2.txt
echo "&lt;&gt;" > corpus/seed3.txt
echo "&amp;&lt;&gt;" > corpus/seed4.txt
echo "&lt;&amp;" > corpus/seed5.txt
echo "&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;" > corpus/seed6.txt
```

And increased input size:

`./tests/escape_fuzzer_test ../corpus -max_len=128 -max_total_time=60`

Still nothing interesting:

```
./tests/escape_fuzzer_test ../corpus -max_len=128 -max_total_time=60

INFO: Running with entropic power schedule (0xFF, 100).
INFO: Seed: 390590001
INFO: Loaded 1 modules   (3 inline 8-bit counters): 3 [0x5c49364d2ea8, 0x5c49364d2eab),
INFO: Loaded 1 PC tables (3 PCs): 3 [0x5c49364d2eb0,0x5c49364d2ee0),
INFO:        6 files found in ../corpus
INFO: seed corpus: files: 6 min: 5b max: 56b total: 100b rss: 31Mb
#7      INITED cov: 2 ft: 2 corp: 1/5b exec/s: 0 rss: 32Mb
#8      REDUCE cov: 2 ft: 2 corp: 1/1b lim: 5 exec/s: 0 rss: 32Mb L: 1/1 MS: 1 CrossOver-
#4194304        pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1398101 rss: 533Mb
#8388608        pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1398101 rss: 534Mb
#16777216       pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1290555 rss: 534Mb
#33554432       pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1290555 rss: 535Mb
#67108864       pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1266204 rss: 535Mb
#76124739       DONE   cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1247946 rss: 535Mb
Done 76124739 runs in 61 second(s)
```

Apparantly the fuzzer keeps reducing the corpus input to 1 byte, which likely means that there's no input of significance being tested.

Tried with even more seed files:

```
echo "&amp;&lt;&gt;" > corpus/seed7.txt
echo "&&&lt;&&&amp;" > corpus/seed8.txt
```

And more runtime:

`./tests/escape_fuzzer_test ../corpus -max_len=128 -max_total_time=300`

5 minutes runtime should find something, but the input was still reduced to a single byte.

```
INFO: seed corpus: files: 9 min: 1b max: 56b total: 129b rss: 31Mb
#10     INITED cov: 2 ft: 2 corp: 1/1b exec/s: 0 rss: 32Mb
```

No loop here either:

```
./tests/escape_fuzzer_test ../corpus -max_len=128 -max_total_time=300

INFO: Running with entropic power schedule (0xFF, 100).
INFO: Seed: 3100794381
INFO: Loaded 1 modules   (3 inline 8-bit counters): 3 [0x5869bdee1ea8, 0x5869bdee1eab),
INFO: Loaded 1 PC tables (3 PCs): 3 [0x5869bdee1eb0,0x5869bdee1ee0),
INFO:        9 files found in ../corpus
INFO: seed corpus: files: 9 min: 1b max: 56b total: 129b rss: 31Mb
#10     INITED cov: 2 ft: 2 corp: 1/1b exec/s: 0 rss: 32Mb
#4194304        pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1398101 rss: 532Mb
#8388608        pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1398101 rss: 533Mb
#16777216       pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1290555 rss: 535Mb
#33554432       pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1290555 rss: 535Mb
#67108864       pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1266204 rss: 535Mb
#134217728      pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1254371 rss: 536Mb
#268435456      pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1237029 rss: 536Mb
#368591945      DONE   cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1224557 rss: 536Mb
Done 368591945 runs in 301 second(s)
```

Ran once more without the custom corpus, for 2 minutes:

```
./tests/escape_fuzzer_test -max_len=128 -max_total_time=120
INFO: Running with entropic power schedule (0xFF, 100).
INFO: Seed: 1506619403
INFO: Loaded 1 modules   (3 inline 8-bit counters): 3 [0x569563f00ea8, 0x569563f00eab),
INFO: Loaded 1 PC tables (3 PCs): 3 [0x569563f00eb0,0x569563f00ee0),
INFO: A corpus is not provided, starting from an empty corpus
#2      INITED cov: 2 ft: 2 corp: 1/1b exec/s: 0 rss: 32Mb
#4194304        pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1398101 rss: 532Mb
#8388608        pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1398101 rss: 533Mb
#16777216       pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1290555 rss: 533Mb
#33554432       pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1290555 rss: 534Mb
#67108864       pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1290555 rss: 535Mb
#134217728      pulse  cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1278264 rss: 536Mb
#153810807      DONE   cov: 2 ft: 2 corp: 1/1b lim: 128 exec/s: 1271163 rss: 536Mb
Done 153810807 runs in 121 second(s)
```

No ampersandloop for you!

NO AMPERSANDLOOP
FOR YOU!
imgflip.com