Fuzzing

Code can be found here: https://github.com/c2003-tamu/413

Environment Setup

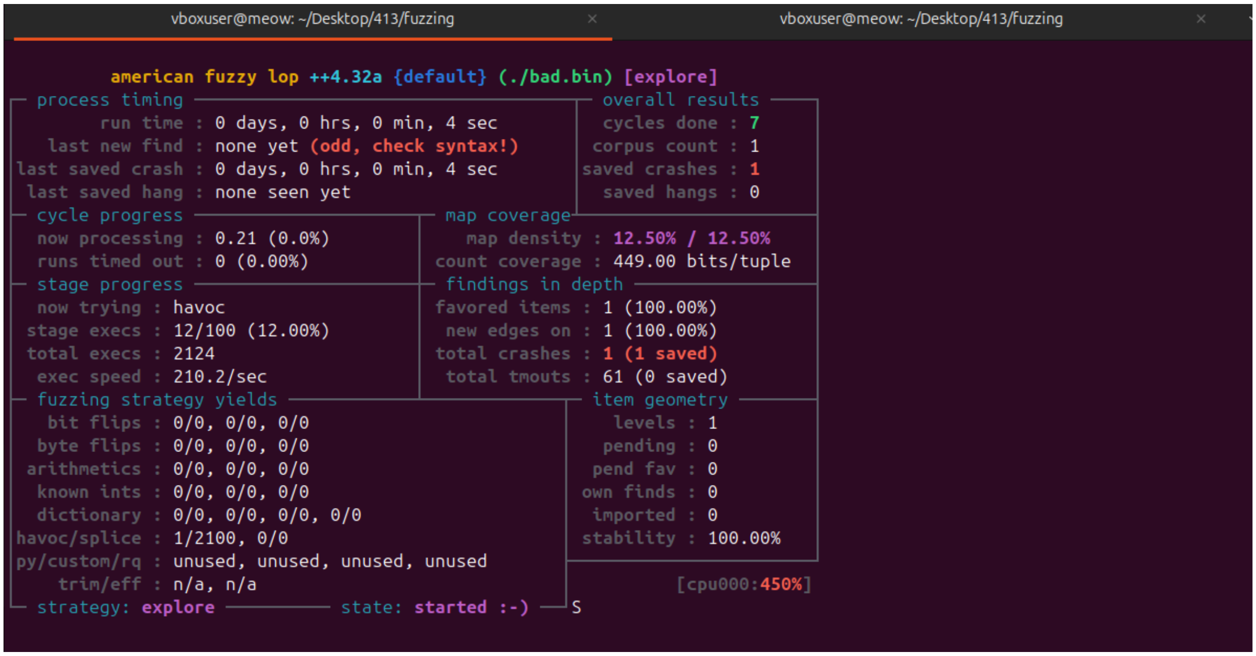
- Clone git repository
- Ensure that docker is installed
    - sudo apt install docker.io
- Navigate to 413/fuzzing directory

Running AFL

NOTE: we are using the docker container found here: https://github.com/AFLplusplus/AFLplusplus

NOTE: we are using a program that is vulnerable to a buffer overflow, using the gets() function

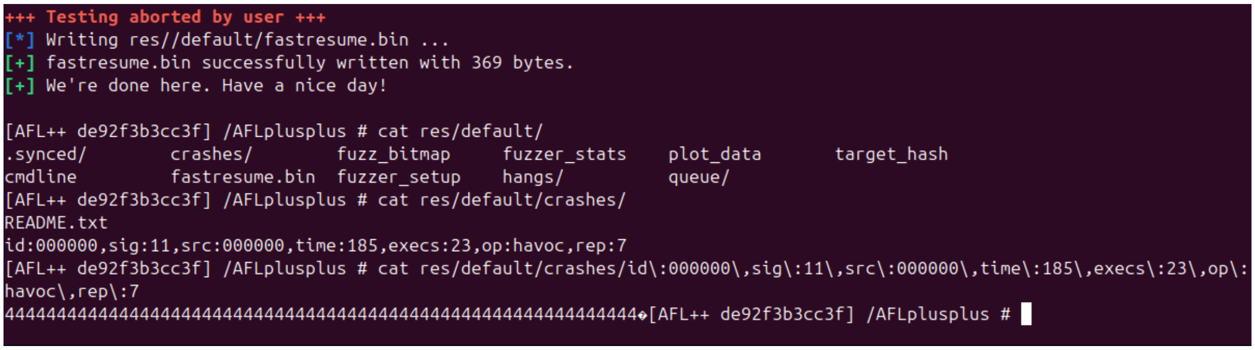
- Run command: docker pull aflplusplus/aflplusplus
- Navigate to 413/fuzzing
- Run command: docker run -ti -v $(pwd):/src aflplusplus/aflplusplus
- Inside docker container:
    - Run command: mkdir seeds
    - Run command: mkdir res
    - Run command: ./afl-cc -std=c99 src/bad.c -o bad.bin
        - NOTE: we had to downgrade std to c99 in order to be able to use the vulnerable gets() function, which is what allows for a buffer overflow.
    - Run command: echo A > seeds/input1.txt
    - Run command: ./afl-fuzz -i seeds/ -o res/ -- ./bad.bin
        - AFL will now be running, testing different inputs for our vulnerable application, as seen below:



```
          american fuzzy lop ++4.32a {default} (./bad.bin) [explore]
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 0 min, 4 sec       │  cycles done : 7     │
│   last new find : none yet (odd, check syntax!)     │ corpus count : 1     │
│last saved crash : 0 days, 0 hrs, 0 min, 4 sec       │saved crashes : 1     │
│ last saved hang : none seen yet                     │  saved hangs : 0     │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 0.21 (0.0%)        │    map density : 12.50% / 12.50%    │
│  runs timed out : 0 (0.00%)          │ count coverage : 449.00 bits/tuple  │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : havoc                  │ favored items : 1 (100.00%)         │
│ stage execs : 12/100 (12.00%)        │  new edges on : 1 (100.00%)         │
│ total execs : 2124                   │ total crashes : 1 (1 saved)         │
│  exec speed : 210.2/sec              │  total tmouts : 61 (0 saved)        │
├─ fuzzing strategy yields ────────────┴─────────────┬─ item geometry ───────┤
│   bit flips : 0/0, 0/0, 0/0                        │    levels : 1         │
│  byte flips : 0/0, 0/0, 0/0                        │   pending : 0         │
│ arithmetics : 0/0, 0/0, 0/0                        │  pend fav : 0         │
│  known ints : 0/0, 0/0, 0/0                        │ own finds : 0         │
│  dictionary : 0/0, 0/0, 0/0, 0/0                   │  imported : 0         │
│havoc/splice : 1/2100, 0/0                          │ stability : 100.00%   │
│py/custom/rq : unused, unused, unused, unused       ├───────────────────────┘
│    trim/eff : n/a, n/a                             │          [cpu000:450%]
└─ strategy: explore ────────── state: started :-) ──┘S
```

AFL Findings

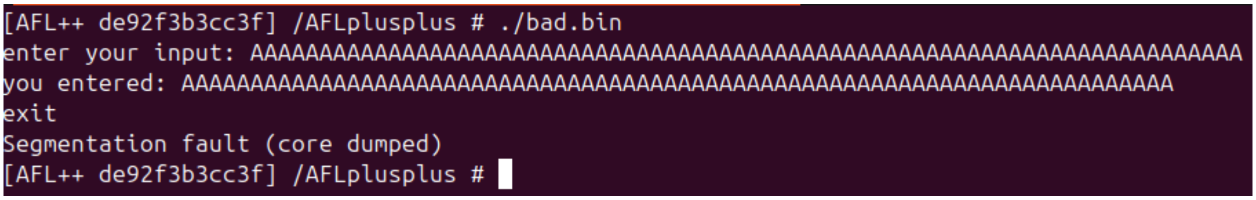
After running AFL, we can see that AFL induced a buffer overflow (and crashed due to seg fault) with the following input:



```
+++ Testing aborted by user +++
[*] Writing res//default/fastresume.bin ...
[+] fastresume.bin successfully written with 369 bytes.
[+] We're done here. Have a nice day!

[AFL++ de92f3b3cc3f] /AFLplusplus # cat res/default/
.synced/         crashes/         fuzz_bitmap      fuzzer_stats     plot_data        target_hash
cmdline          fastresume.bin   fuzzer_setup     hangs/           queue/
[AFL++ de92f3b3cc3f] /AFLplusplus # cat res/default/crashes/
README.txt
id:000000,sig:11,src:000000,time:185,execs:23,op:havoc,rep:7
[AFL++ de92f3b3cc3f] /AFLplusplus # cat res/default/crashes/id\:000000\,sig\:11\,src\:000000\,time\:185\,execs\:23\,op\:
havoc\,rep\:7
44444444444444444444444444444444444444444444444444444444444444�[AFL++ de92f3b3cc3f] /AFLplusplus # 
```

So, now we now that any long input will crash the program, as seen below:



```
[AFL++ de92f3b3cc3f] /AFLplusplus # ./bad.bin
enter your input: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
you entered: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
exit
Segmentation fault (core dumped)
[AFL++ de92f3b3cc3f] /AFLplusplus # 
```