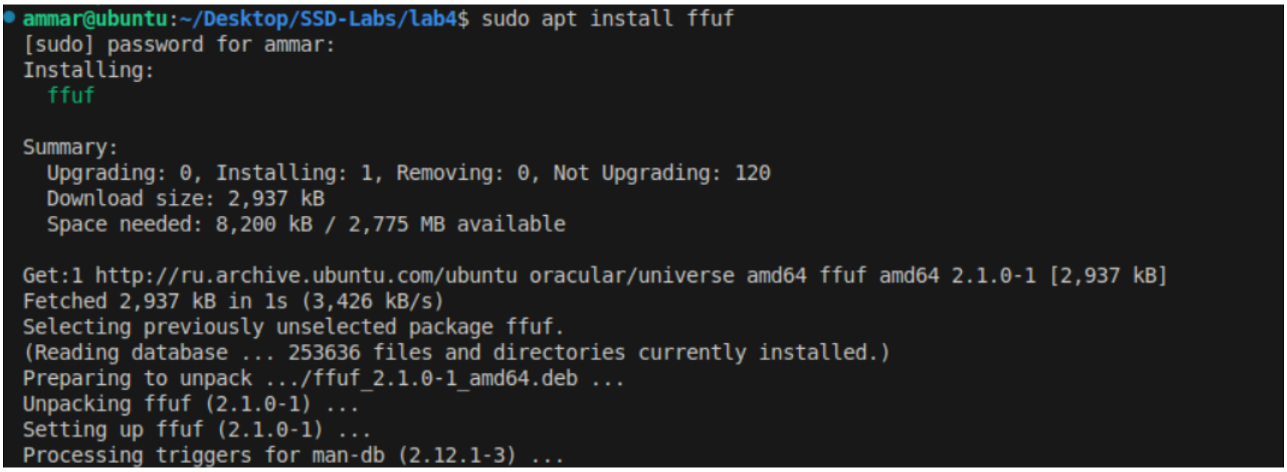
Ammar Meslmani - CBS-01

a.meslmani@innopolis.university

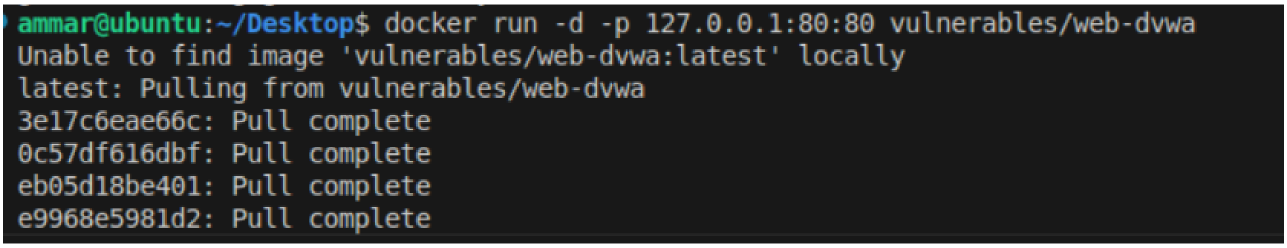
the repo link to check the files used in this assignment: full report

Lab 4

Task 1

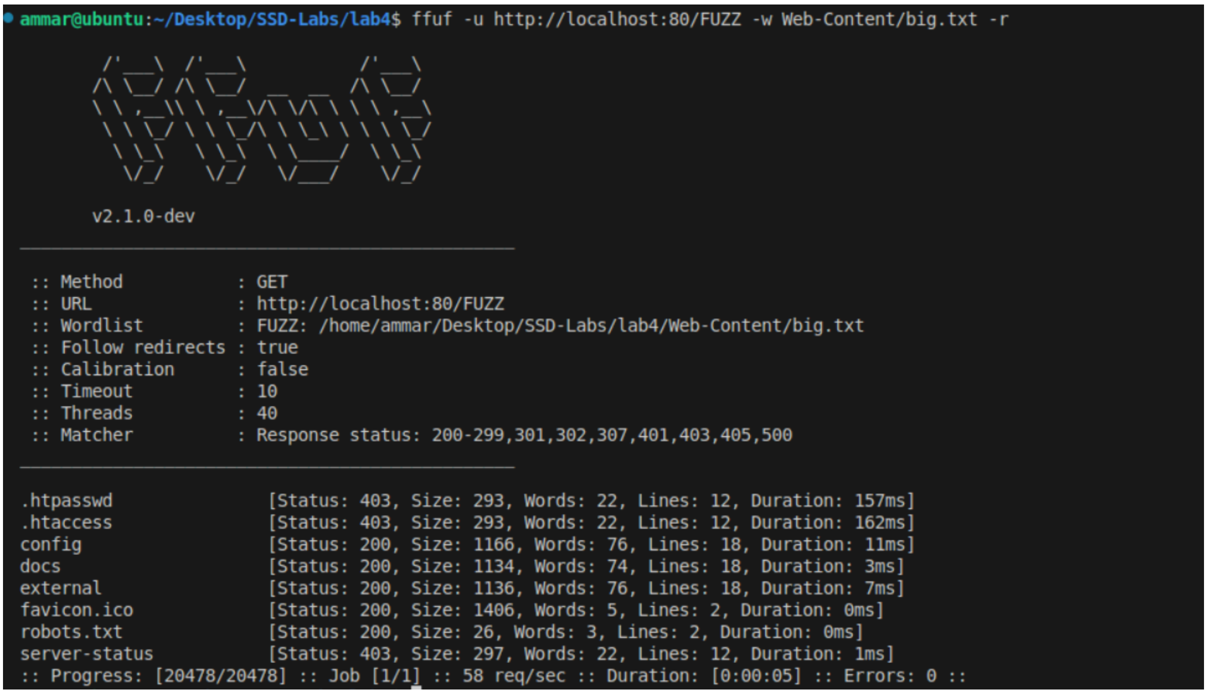
- let's install `ffuf`
- 

```
ammar@ubuntu:~/Desktop/SSD-Labs/lab4$ sudo apt install ffuf
[sudo] password for ammar:
Installing:
  ffuf

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 120
  Download size: 2,937 kB
  Space needed: 8,200 kB / 2,775 MB available

Get:1 http://ru.archive.ubuntu.com/ubuntu oracular/universe amd64 ffuf amd64 2.1.0-1 [2,937 kB]
Fetched 2,937 kB in 1s (3,426 kB/s)
Selecting previously unselected package ffuf.
(Reading database ... 253636 files and directories currently installed.)
Preparing to unpack .../ffuf_2.1.0-1_amd64.deb ...
Unpacking ffuf (2.1.0-1) ...
Setting up ffuf (2.1.0-1) ...
Processing triggers for man-db (2.12.1-3) ...
```

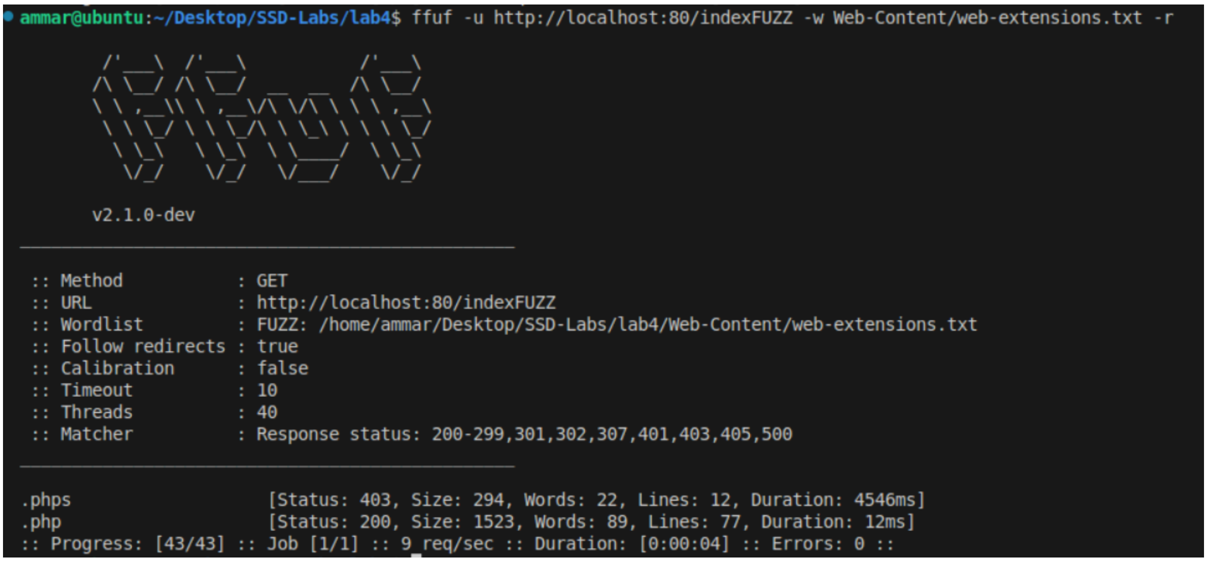
- let's run `DVWA` locally
- 

```
ammar@ubuntu:~/Desktop$ docker run -d -p 127.0.0.1:80:80 vulnerables/web-dvwa
Unable to find image 'vulnerables/web-dvwa:latest' locally
latest: Pulling from vulnerables/web-dvwa
3e17c6eae66c: Pull complete
0c57df616dbf: Pull complete
eb05d18be401: Pull complete
e9968e5981d2: Pull complete
```

- let's run: `ffuf -u http://localhost:80/FUZZ -w Web-Content/big.txt -r`
    - **`-u http://localhost:80/FUZZ`** : target URL with `FUZZ` as a placeholder for wordlist entries
    - **`-w Web-Content/big.txt`** : uses `big.txt` as input for `FUZZ`
    - `-r`: follows HTTP redirects
- which endpoints/files from big.txt were accessible?

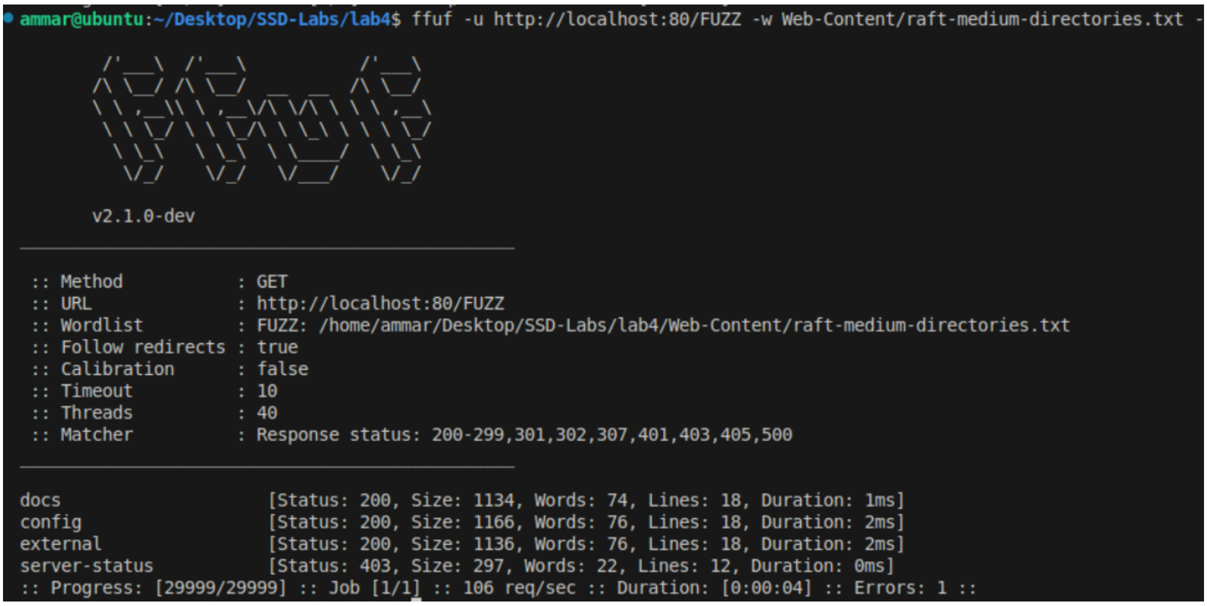
  - 

```
ammar@ubuntu:~/Desktop/SSD-Labs/lab4$ ffuf -u http://localhost:80/FUZZ -w Web-Content/big.txt -r

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://localhost:80/FUZZ
 :: Wordlist         : FUZZ: /home/ammar/Desktop/SSD-Labs/lab4/Web-Content/big.txt
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

.htpasswd               [Status: 403, Size: 293, Words: 22, Lines: 12, Duration: 157ms]
.htaccess               [Status: 403, Size: 293, Words: 22, Lines: 12, Duration: 162ms]
config                  [Status: 200, Size: 1166, Words: 76, Lines: 18, Duration: 11ms]
docs                    [Status: 200, Size: 1134, Words: 74, Lines: 18, Duration: 3ms]
external                [Status: 200, Size: 1136, Words: 76, Lines: 18, Duration: 7ms]
favicon.ico             [Status: 200, Size: 1406, Words: 5, Lines: 2, Duration: 0ms]
robots.txt              [Status: 200, Size: 26, Words: 3, Lines: 2, Duration: 0ms]
server-status           [Status: 403, Size: 297, Words: 22, Lines: 12, Duration: 1ms]
:: Progress: [20478/20478] :: Job [1/1] :: 58 req/sec :: Duration: [0:00:05] :: Errors: 0 ::
```

  - according to the result of the command, `favicon.ico` and `robots.txt` files and `external`, `docs`, and `config` directories
- which ones gave interesting error codes (not 404)?
  - 403: `.htaccess`, `.thpasswd`, and `server-status`, which indicates that this file exists, but it's forbidden to be accessed
- now let's run: `ffuf -u http://localhost:80/FUZZ -w Web-Content/big.txt -r`
  - `-u http://localhost:80/indexFUZZ` : targets `index` files with dynamic extensions
  - **`-w Web-Content/web-extensions.txt`** : wordlist containing extensions
  - `-r`: follows HTTP redirects
- what file extensions from web-extensions.txt are available for the index page?
  - 

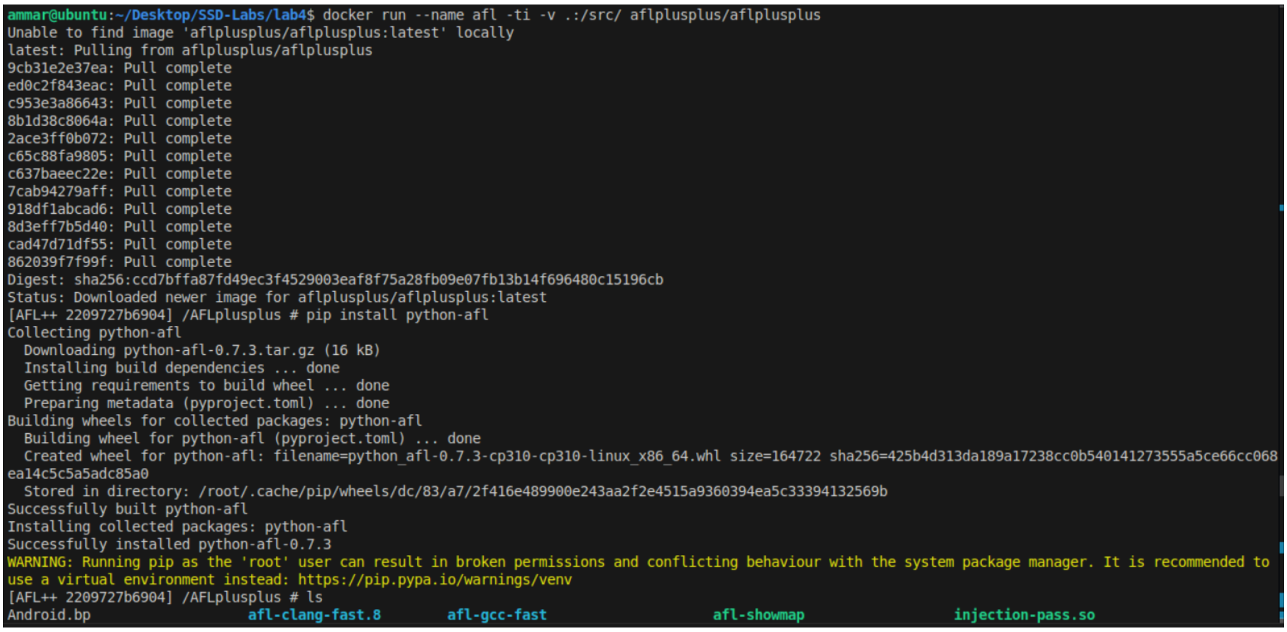
```
ammar@ubuntu:~/Desktop/SSD-Labs/lab4$ ffuf -u http://localhost:80/indexFUZZ -w Web-Content/web-extensions.txt -r

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://localhost:80/indexFUZZ
 :: Wordlist         : FUZZ: /home/ammar/Desktop/SSD-Labs/lab4/Web-Content/web-extensions.txt
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

.phps                   [Status: 403, Size: 294, Words: 22, Lines: 12, Duration: 4546ms]
.php                    [Status: 200, Size: 1523, Words: 89, Lines: 77, Duration: 12ms]
:: Progress: [43/43] :: Job [1/1] :: 9 req/sec :: Duration: [0:00:04] :: Errors: 0 ::
```

  - according to the result of the command, `.php` (redirection) and `.phps` (forbidden)
- now let's run `fuf -u http://localhost:80/FUZZ -w Web-Content/raft-medium-directories.txt -r`
  - `-u http://localhost:80/FUZZ`: `FUZZ` is a placeholder where **ffuf** inserts entries from the wordlist
  - **`-w Web-Content/raft-medium-directories.txt`** : uses `raft-medium-directories.txt` (a curated list of common dirs) for `FUZZ`
  - `-r`: follows HTTP redirects

- which directories from raft-medium-directories.txt are accessible?
  - 

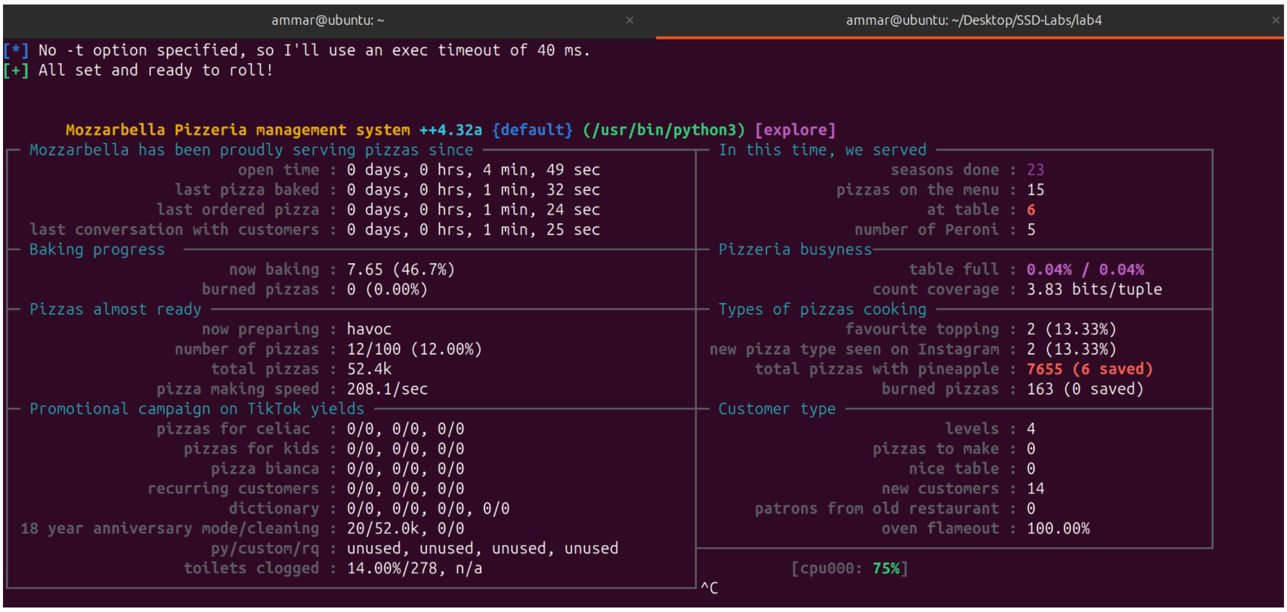
    ```
    ammar@ubuntu:~/Desktop/SSD-Labs/lab4$ ffuf -u http://localhost:80/FUZZ -w Web-Content/raft-medium-directories.txt -

            /'___\  /'___\           /'___\
           /\ \__/ /\ \__/  __  __  /\ \__/
           \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
            \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
             \ \_\   \ \_\  \ \____/  \ \_\
              \/_/    \/_/   \/___/    \/_/

           v2.1.0-dev
    ________________________________________________

     :: Method           : GET
     :: URL              : http://localhost:80/FUZZ
     :: Wordlist         : FUZZ: /home/ammar/Desktop/SSD-Labs/lab4/Web-Content/raft-medium-directories.txt
     :: Follow redirects : true
     :: Calibration      : false
     :: Timeout          : 10
     :: Threads          : 40
     :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
    ________________________________________________

    docs                    [Status: 200, Size: 1134, Words: 74, Lines: 18, Duration: 1ms]
    config                  [Status: 200, Size: 1166, Words: 76, Lines: 18, Duration: 2ms]
    external                [Status: 200, Size: 1136, Words: 76, Lines: 18, Duration: 2ms]
    server-status           [Status: 403, Size: 297, Words: 22, Lines: 12, Duration: 0ms]
    :: Progress: [29999/29999] :: Job [1/1] :: 106 req/sec :: Duration: [0:00:04] :: Errors: 1 ::
    ```
  - according to the result of the command, `external`, `docs`, and `config` directories
- which ones gave interesting error codes (not 404)?
  - 403: `server-status`, which indicates that this directory exists, but it's forbidden to be accessed

Task 2

- let's install AFL++ locally using the docker image:
- 

  ```
  ammar@ubuntu:~/Desktop/SSD-Labs/lab4$ docker run --name afl -ti -v .:/src/ aflplusplus/aflplusplus
  Unable to find image 'aflplusplus/aflplusplus:latest' locally
  latest: Pulling from aflplusplus/aflplusplus
  9cb31e2e37ea: Pull complete
  ed0c2f843eac: Pull complete
  c953e3a86643: Pull complete
  8b1d38c8064a: Pull complete
  2ace3ff0b072: Pull complete
  c65c88fa9805: Pull complete
  c637baeec22e: Pull complete
  7cab94279aff: Pull complete
  918df1abcad6: Pull complete
  8d3eff7b5d40: Pull complete
  cad47d71df55: Pull complete
  862039f7f99f: Pull complete
  Digest: sha256:ccd7bffa87fd49ec3f4529003eaf8f75a28fb09e07fb13b14f696480c15196cb
  Status: Downloaded newer image for aflplusplus/aflplusplus:latest
  [AFL++ 2209727b6904] /AFLplusplus # pip install python-afl
  Collecting python-afl
    Downloading python-afl-0.7.3.tar.gz (16 kB)
    Installing build dependencies ... done
    Getting requirements to build wheel ... done
    Preparing metadata (pyproject.toml) ... done
  Building wheels for collected packages: python-afl
    Building wheel for python-afl (pyproject.toml) ... done
    Created wheel for python-afl: filename=python_afl-0.7.3-cp310-cp310-linux_x86_64.whl size=164722 sha256=425b4d313da189a17238cc0b540141273555a5ce66cc068
  ea14c5c5a5adc85a0
    Stored in directory: /root/.cache/pip/wheels/dc/83/a7/2f416e489900e243aa2f2e4515a9360394ea5c33394132569b
  Successfully built python-afl
  Installing collected packages: python-afl
  Successfully installed python-afl-0.7.3
  WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to
  use a virtual environment instead: https://pip.pypa.io/warnings/venv
  [AFL++ 2209727b6904] /AFLplusplus # ls
  Android.bp                    afl-clang-fast.8       afl-gcc-fast                  afl-showmap                   injection-pass.so
  ```
- let's prepare `input.txt` file:

  ```
  finish_him
  ```
- let's run the fuzzer using `py-afl-fuzz -i input -o output -- /usr/bin/python3 main.py`

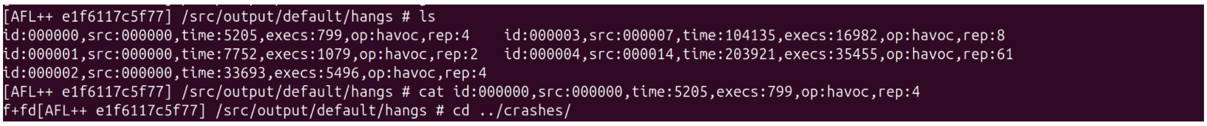
- 

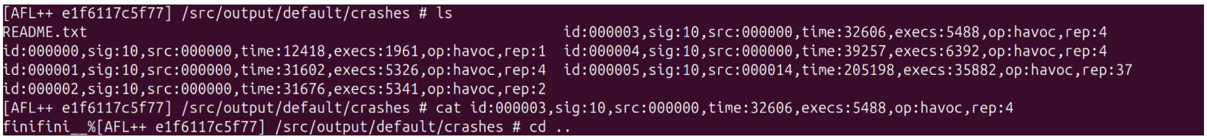
```
[*] No -t option specified, so I'll use an exec timeout of 40 ms.
[+] All set and ready to roll!


      Mozzarbella Pizzeria management system ++4.32a {default} (/usr/bin/python3) [explore]
┌─ Mozzarbella has been proudly serving pizzas since ─────────┬─ In this time, we served ────────────┐
│                         open time : 0 days, 0 hrs, 4 min, 49 sec │          seasons done : 23           │
│                  last pizza baked : 0 days, 0 hrs, 1 min, 32 sec │     pizzas on the menu : 15          │
│                last ordered pizza : 0 days, 0 hrs, 1 min, 24 sec │               at table : 6           │
│  last conversation with customers : 0 days, 0 hrs, 1 min, 25 sec │        number of Peroni : 5          │
├─ Baking progress ──────────────────────────────────────────────┼─ Pizzeria busyness──────────────────┤
│                        now baking : 7.65 (46.7%)                │            table full : 0.04% / 0.04% │
│                     burned pizzas : 0 (0.00%)                   │        count coverage : 3.83 bits/tuple │
├─ Pizzas almost ready ──────────────────────────────────────────┼─ Types of pizzas cooking ────────────┤
│                     now preparing : havoc                       │        favourite topping : 2 (13.33%) │
│                  number of pizzas : 12/100 (12.00%)             │new pizza type seen on Instagram : 2 (13.33%) │
│                      total pizzas : 52.4k                       │  total pizzas with pineapple : 7655 (6 saved) │
│                pizza making speed : 208.1/sec                   │            burned pizzas : 163 (0 saved) │
├─ Promotional campaign on TikTok yields ────────────────────────┼─ Customer type ─────────────────────┤
│                pizzas for celiac  : 0/0, 0/0, 0/0               │                   levels : 4          │
│                  pizzas for kids : 0/0, 0/0, 0/0                │            pizzas to make : 0         │
│                      pizza bianca : 0/0, 0/0, 0/0               │               nice table : 0          │
│               recurring customers : 0/0, 0/0, 0/0               │            new customers : 14         │
│                        dictionary : 0/0, 0/0, 0/0, 0/0          │ patrons from old restaurant : 0       │
│ 18 year anniversary mode/cleaning : 20/52.0k, 0/0               │            oven flameout : 100.00%    │
│                      py/custom/rq : unused, unused, unused, unused ├─────────────────────────────────────┘
│                   toilets clogged : 14.00%/278, n/a             │          [cpu000: 75%]
└─────────────────────────────────────────────────────────────────┘^C
```

- let's inspect `fuzzer_stats` file:

```
[AFL++ e1f6117c5f77] /src/output/default # cat fuzzer_stats
start_time        : 1743538971
last_update       : 1743539261
run_time          : 289
fuzzer_pid        : 154504
cycles_done       : 23
cycles_wo_finds   : 9
time_wo_finds     : 92
fuzz_time         : 287
calibration_time  : 0
cmplog_time       : 0
sync_time         : 0
trim_time         : 1
execs_done        : 52407
execs_per_sec     : 180.87
execs_ps_last_min : 192.13
corpus_count      : 15
corpus_favored    : 2
corpus_found      : 14
corpus_imported   : 0
corpus_variable   : 0
max_depth         : 4
cur_item          : 7
pending_favs      : 0
pending_total     : 0
stability         : 100.00%
bitmap_cvg        : 0.04%
saved_crashes     : 6
saved_hangs       : 5
total_tmout       : 163
last_find         : 1743539168
last_crash        : 1743539177
last_hang         : 1743539175
execs_since_crash : 16525
```

```
exec_timeout      : 40
slowest_exec_ms   : 0
peak_rss_mb       : 0
cpu_affinity      : 0
edges_found       : 23
total_edges       : 65536
var_byte_count    : 0
havoc_expansion   : 5
auto_dict_entries : 0
testcache_size    : 820
testcache_count   : 15
testcache_evict   : 0
afl_banner        : /usr/bin/python3
afl_version       : ++4.32a
target_mode       : shmem_testcase default
command_line      : afl-fuzz -i input/ -o output -- /usr/bin/python3
main.py
```

- let's inspect one `crash` and one `hang` cases
  - `hang`: f+fd
  - 

    ```
    [AFL++ e1f6117c5f77] /src/output/default/hangs # ls
    id:000000,src:000000,time:5205,execs:799,op:havoc,rep:4    id:000003,src:000007,time:104135,execs:16982,op:havoc,rep:8
    id:000001,src:000000,time:7752,execs:1079,op:havoc,rep:2   id:000004,src:000014,time:203921,execs:35455,op:havoc,rep:61
    id:000002,src:000000,time:33693,execs:5496,op:havoc,rep:4
    [AFL++ e1f6117c5f77] /src/output/default/hangs # cat id:000000,src:000000,time:5205,execs:799,op:havoc,rep:4
    f+fd[AFL++ e1f6117c5f77] /src/output/default/hangs # cd ../crashes/
    ```
  - problem: the loop gets stuck because `i` isn’t incremented when `s[i] == '+'` and this causes an infinite loop on the `+` character
  - fix: increase `i` in the elif block
  - `crash`: finifini__%
  - 

    ```
    [AFL++ e1f6117c5f77] /src/output/default/crashes # ls
    README.txt                                            id:000003,sig:10,src:000000,time:32606,execs:5488,op:havoc,rep:4
    id:000000,sig:10,src:000000,time:12418,execs:1961,op:havoc,rep:1  id:000004,sig:10,src:000000,time:39257,execs:6392,op:havoc,rep:4
    id:000001,sig:10,src:000000,time:31602,execs:5326,op:havoc,rep:4  id:000005,sig:10,src:000014,time:205198,execs:35882,op:havoc,rep:37
    id:000002,sig:10,src:000000,time:31676,execs:5341,op:havoc,rep:2
    [AFL++ e1f6117c5f77] /src/output/default/crashes # cat id:000003,sig:10,src:000000,time:32606,execs:5488,op:havoc,rep:4
    finifini__%[AFL++ e1f6117c5f77] /src/output/default/crashes # cd ..
    ```
  - problem: the code crashes when `%` is at the end of the string because it tries to access `s[i+1]` and `s[i+2]` (out-of-bounds)
  - fix: check if `%` has 2 valid digits after it

- Will the fuzzer ever terminate in the above experiment? Why/Why not?

  no, because the mutation will never end (it's a continuous process)

- How coverage-guided fuzzers work? Is AFL coverage-guided?

  they track which code paths are executed by mutations and prioritize inputs that explore new branches. yes, AFL is coverage-guided as it mutates inputs to maximize code coverage and find crashes

- How to optimize a fuzzing campaign?

  increase the quality of the seed which we start with and paralellize the fuzzing process