Lab 4 - Fuzzing

Task 1 - WebApp Fuzzing

Install `ffuf` and SecLists

ffuf installation

```
sudo apt install ffuf
```

```
mohamad@mohamad-HP-ProBook-430-G7:~/Desktop/thirdYear/second-semester/secure-system-development/lab4/ffuf_2.1.0_linux_amd64$ sudo apt install ffuf
[sudo] password for mohamad:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ffuf
0 upgraded, 1 newly installed, 0 to remove and 9 not upgraded.
Need to get 2 118 kB of archives.
After this operation, 6 935 kB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 ffuf amd64 1.1.0-1 [2 118 kB]
Fetched 2 118 kB in 1s (1 942 kB/s)
Selecting previously unselected package ffuf.
(Reading database ... 227865 files and directories currently installed.)
Preparing to unpack .../ffuf_1.1.0-1_amd64.deb ...
Unpacking ffuf (1.1.0-1) ...
Setting up ffuf (1.1.0-1) ...
Processing triggers for man-db (2.10.2-1) ...
```

SecLists

Download SecLists from GitHub, unzip it, and remvove the zipped file:

```
wget -c https://github.com/danielmiessler/SecLists/archive/master.zip -O
SecList.zip && unzip SecList.zip && rm -f SecList.zip
```

```
mohamad@mohamad-HP-ProBook-430-G7:~/Downloads$ wget -c https://github.com/danielmiessler/SecLists/archive/master.zip -O SecList.zip && unzip SecList.zip && rm -f SecList.zip
--2025-03-26 15:19:51--  https://github.com/danielmiessler/SecLists/archive/master.zip
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/danielmiessler/SecLists/zip/refs/heads/master [following]
--2025-03-26 15:19:51--  https://codeload.github.com/danielmiessler/SecLists/zip/refs/heads/master
Resolving codeload.github.com (codeload.github.com)... 140.82.121.9
Connecting to codeload.github.com (codeload.github.com)|140.82.121.9|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 714968970 (682M) [application/zip]
Saving to: ‘SecList.zip’

SecList.zip             100%[===================================>] 681,85M  9,03MB/s    in 4m 43s
```

Run DVWA with Docker

```
docker run -d -p 127.0.0.1:80:80 vulnerables/web-dvwa
```

- `-d`: Run container in detached mode

- `-p`: Map host port 80 to container port 80

```
mohamad@mohamad-HP-ProBook-430-G7:~/Downloads$ 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
2cd72dba8257: Pull complete
6cff5f35147f: Pull complete
098cffd43466: Pull complete
b3d64a33242d: Pull complete
Digest: sha256:dae203fe11646a86937bf04db0079adef295f426da68a92b40e3b181f337daa7
Status: Downloaded newer image for vulnerables/web-dvwa:latest
a5fa158037331be772ef9924ebb41c20e200dae5ebbc26a15ebef5585afc93b3
mohamad@mohamad-HP-ProBook-430-G7:~/Downloads$
```

Fuzz Endpoints using `big.txt`

```
ffuf -u http://localhost:80/FUZZ -w ./SecLists-master/Discovery/Web-Content/big.txt
```

- `-u`: Target URL where `FUZZ` is replaced by each word
- `-w`: Wordlist path

```
mohamad@mohamad-HP-ProBook-430-G7:~/Desktop/thirdYear/second-semester/secure-system-development/lab4$ ffuf -u http://localhost:80/FUZZ -w ./SecLists-master/Discovery/Web-Content/big.txt

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

       v1.1.0
________________________________________________

 :: Method           : GET
 :: URL              : http://localhost:80/FUZZ
 :: Wordlist         : FUZZ: ./SecLists-master/Discovery/Web-Content/big.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
________________________________________________

.htpasswd               [Status: 403, Size: 293, Words: 22, Lines: 12]
.htaccess               [Status: 403, Size: 293, Words: 22, Lines: 12]
config                  [Status: 301, Size: 307, Words: 20, Lines: 10]
docs                    [Status: 301, Size: 305, Words: 20, Lines: 10]
external                [Status: 301, Size: 309, Words: 20, Lines: 10]
favicon.ico             [Status: 200, Size: 1405, Words: 5, Lines: 2]
robots.txt              [Status: 200, Size: 26, Words: 3, Lines: 2]
server-status           [Status: 403, Size: 297, Words: 22, Lines: 12]
:: Progress: [20478/20478] :: Job [1/1] :: 20478 req/sec :: Duration: [0:00:01] :: Errors: 0 ::
```

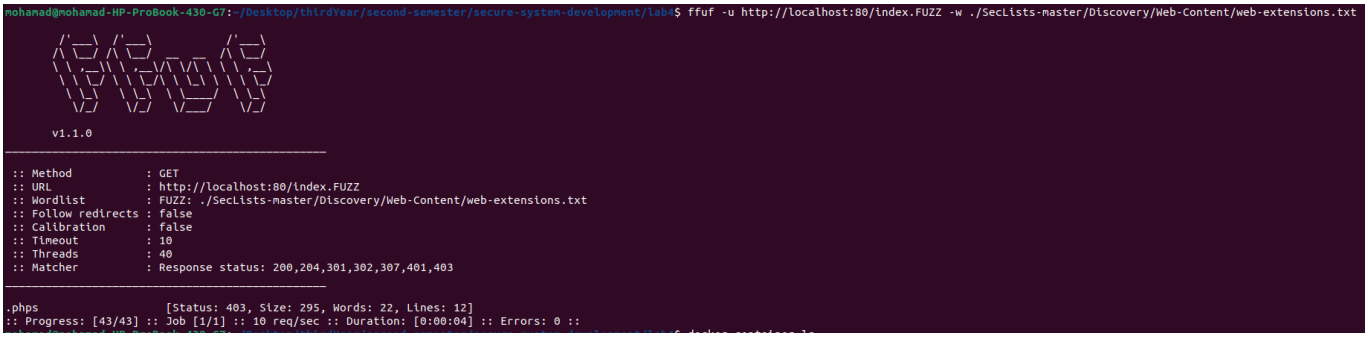
Observations

```
.htpasswd               [Status: 403, Size: 293, Words: 22, Lines: 12]
.htaccess               [Status: 403, Size: 293, Words: 22, Lines: 12]
config                  [Status: 301, Size: 307, Words: 20, Lines: 10]
docs                    [Status: 301, Size: 305, Words: 20, Lines: 10]
external                [Status: 301, Size: 309, Words: 20, Lines: 10]
favicon.ico             [Status: 200, Size: 1405, Words: 5, Lines: 2]
robots.txt              [Status: 200, Size: 26, Words: 3, Lines: 2]
server-status           [Status: 403, Size: 297, Words: 22, Lines: 12]
```

- `200`: Accessible → `robots.txt`, `favicon.ico`
- `301`: Redirected → `config`, `docs`, `external`
- `403`: Forbidden (interesting) → `.htaccess`, `.htpasswd`, `server-status`

What file extensions are available for `index`?

```
ffuf -u http://localhost:80/index.FUZZ -w ./SecLists-master/Discovery/Web-Content/web-extensions.txt
```

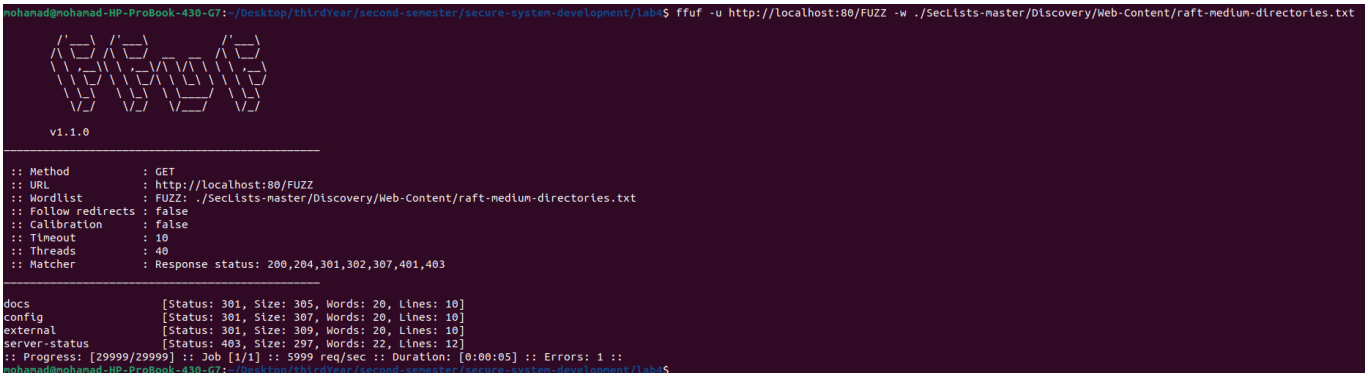


Result

- `index.php` found, but returns `403` (forbidden)

Directory Fuzzing using `raft-medium-directories.txt`

```
ffuf -u http://localhost:80/FUZZ -w ./SecLists-master/Discovery/Web-Content/raft-medium-directories.txt
```



Observations1

```
config        [301]
docs          [301]
external      [301]
server-status [403]
```

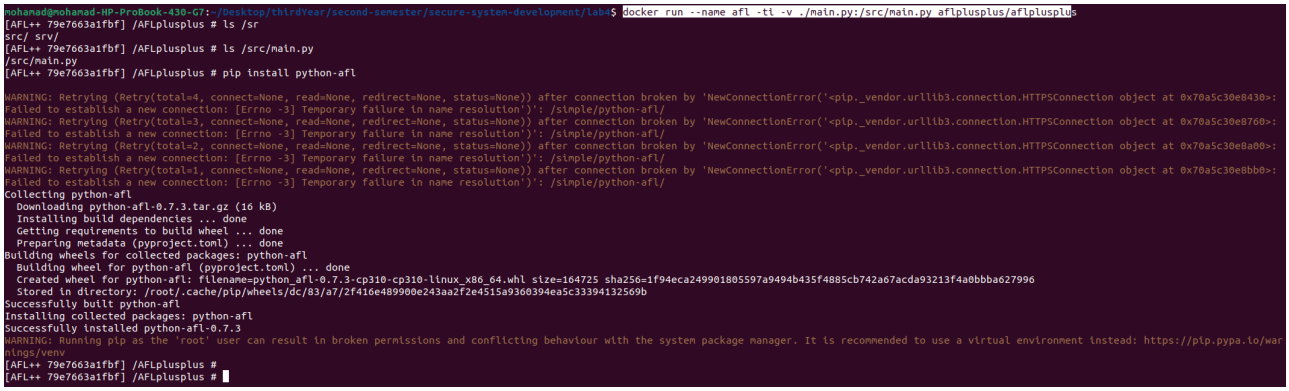
- `301`: `config`, `docs`, `external` → Redirected
- `403`: `server-status` → Forbidden access (interesting)

Task 2 - Python Fuzzing

Run AFL++ Docker & Install `python-afl`

```
docker run --name afl -ti -v ./main.py:/src/main.py aflplusplus/aflplusplus
pip install python-afl
```

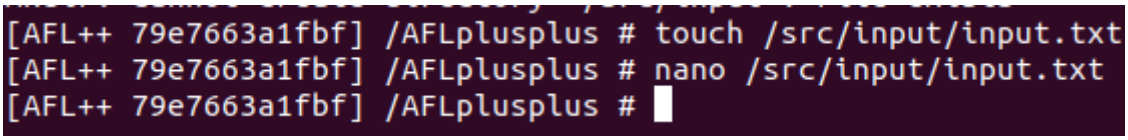
- `--name afl`: Names the container "afl"
- `-ti`: Runs in interactive terminal mode
- `-v ./main.py:/src/main.py`: Mounts `main.py` into the container
- `aflplusplus/aflplusplus`: Uses the AFL++ Docker image



Prepare Input Directory (Seed Corpus)

- Created `/src/input/` with seed input file `input.txt` containing:

```
test
```



Run the Fuzzer

```
cd /src
py-afl-fuzz -i input -o output -- /usr/bin/python3 main.py
```

```
[AFL++ d8ac67853932] /src # py-afl-fuzz -i input/ -o output -- /usr/bin/python3 main.py
[!] WARNING: Mistyped AFL environment variable: AFL_SKIP_CHECKS=1
Did you mean AFL_SKIP_BIN_CHECK?
[+] Enabled environment variable AFL_SKIP_CPUFREQ with value 1
[+] Enabled environment variable AFL_I_DONT_CARE_ABOUT_MISSING_CRASHES with value 1
[+] Enabled environment variable AFL_DUMB_FORKSRV with value 1
[+] Enabled environment variable AFL_SKIP_BIN_CHECK with value 1
[+] Enabled environment variable AFL_TRY_AFFINITY with value 1
afl-fuzz++4.32a based on afl by Michal Zalewski and a large online community
[+] AFL++ is maintained by Marc "van Hauser" Heuse, Dominik Maier, Andrea Fioraldi and Heiko "hexcoder" Eißfeldt
[+] AFL++ is open source, get it at https://github.com/AFLplusplus/AFLplusplus
[+] NOTE: AFL++ >= v3 has changed defaults and behaviours - see README.md
[+] No -M/-S set, autoconfiguring for "-S default"
[*] Getting to work...
[+] Using exploration-based constant power schedule (EXPLORE)
[+] Enabled testcache with 50 MB
[+] Generating fuzz data with a length of min=1 max=1048576
[*] Checking core_pattern...
[+] You have 8 CPU cores and 5 runnable tasks (utilization: 62%).
[+] Try parallel jobs - see /usr/local/share/doc/afl/fuzzing_in_depth.md#c-using-multiple-cores
[*] Setting up output directories...
[*] Checking CPU core loadout...
[+] Found a free CPU core, try binding to #0.
[*] Validating target binary...
[*] Scanning 'input/'...
[*] Creating hard links for all input files...
[+] Loaded a total of 1 seeds.
[*] No auto-generated dictionary tokens to reuse.
[*] Attempting dry run with 'id:000000,time:0,execs:0,orig:1.txt'...
[*] Spinning up the fork server...
[!] WARNING: Old fork server model is used by the target, this still works though.
[+] All right - old fork server is up.
    len = 5, map size = 17, exec speed = 4493 us, hash = 38f851f903055c2d
[+] All test cases processed.
[+] Here are some useful stats:

    Test case count : 1 favored, 0 variable, 0 ignored, 1 total
       Bitmap range : 17 to 17 bits (average: 17.00 bits)
        Exec timing : 4493 to 4493 us (average: 4493 us)

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


       american fuzzy lop ++4.32a {default} (/usr/bin/python3) [explore]
┌─ process timing ──────────────────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 0 hrs, 1 min, 30 sec        │  cycles done : 9      │
│   last new find : 0 days, 0 hrs, 0 min, 35 sec        │ corpus count : 13     │
│last saved crash : 0 days, 0 hrs, 1 min, 11 sec        │saved crashes : 5      │
│ last saved hang : 0 days, 0 hrs, 0 min, 53 sec        │  saved hangs : 4      │
├─ cycle progress ────────────────────┬─ map coverage ─┴───────────────────────┤
│  now processing : 2*1 (15.4%)       │    map density : 0.03% / 0.04%         │
```

Let It Run (Detect Crash & Hang)

```
[*] Setting up output directories...
[*] Checking CPU core loadout...
[+] Found a free CPU core, try binding to #0.
[*] Validating target binary...
[*] Scanning 'input/'...
[*] Creating hard links for all input files...
[+] Loaded a total of 1 seeds.
[*] No auto-generated dictionary tokens to reuse.
[*] Attempting dry run with 'id:000000,time:0,execs:0,orig:1.txt'...
[*] Spinning up the fork server...
[!] WARNING: Old fork server model is used by the target, this still works though.
[+] All right - old fork server is up.
    len = 5, map size = 17, exec speed = 4493 us, hash = 38f851f903055c2d
[+] All test cases processed.
[+] Here are some useful stats:

    Test case count : 1 favored, 0 variable, 0 ignored, 1 total
       Bitmap range : 17 to 17 bits (average: 17.00 bits)
        Exec timing : 4493 to 4493 us (average: 4493 us)

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


       american fuzzy lop ++4.32a {default} (/usr/bin/python3) [explore]
─ process timing ──────────────────────────────────────────────── overall results ─
        run time : 0 days, 0 hrs, 0 min, 3 sec          cycles done : 0
   last new find : 0 days, 0 hrs, 0 min, 2 sec         corpus count : 7
last saved crash : 0 days, 0 hrs, 0 min, 1 sec        saved crashes : 1
 last saved hang : 0 days, 0 hrs, 0 min, 1 sec          saved hangs : 1
─ cycle progress ──────────────────────────── map coverage──────────────────────────
  now processing : 0.0 (0.0%)                    map density : 0.03% / 0.03%
  runs timed out : 0 (0.00%)                  count coverage : 2.76 bits/tuple
─ stage progress ──────────────────────────── findings in depth ──────────────────────
  now trying : havoc                          favored items : 1 (14.29%)
 stage execs : 585/6400 (9.14%)                new edges on : 1 (14.29%)
 total execs : 642                            total crashes : 5 (1 saved)
  exec speed : 298.4/sec                       total tmouts : 1 (0 saved)
─ fuzzing strategy yields ──────────────────────────────────────────── item geometry ─
   bit flips : 0/0, 0/0, 0/0                          levels : 2
  byte flips : 0/0, 0/0, 0/0                         pending : 7
 arithmetics : 0/0, 0/0, 0/0                        pend fav : 1
  known ints : 0/0, 0/0, 0/0                       own finds : 6
  dictionary : 0/0, 0/0, 0/0, 0/0                   imported : 0
havoc/splice : 0/0, 0/0                            stability : 100.00%
py/custom/rq : unused, unused, unused, unused
    trim/eff : 20.00%/1, n/a                                [cpu000: 50%]
─ strategy: explore ──────────── state: started :-) ──────────────────────────────
```

Analyze Results

`fuzzer_stats`

```
cmdline  crashes  fastresume.bin  fuzz_bitmap  fuzzer_setup  fuzzer_stats  hangs  plot_data  queue  target_hash
[AFL++ d8ac67853932] /src/output/default # cat fuzzer_stats
start_time        : 1743202327
last_update       : 1743202455
run_time          : 128
fuzzer_pid        : 48
cycles_done       : 18
cycles_wo_finds   : 16
time_wo_finds     : 73
fuzz_time         : 127
calibration_time  : 0
cmplog_time       : 0
sync_time         : 0
trim_time         : 0
execs_done        : 40967
execs_per_sec     : 319.01
execs_ps_last_min : 360.30
corpus_count      : 13
corpus_favored    : 2
corpus_found      : 12
corpus_imported   : 0
corpus_variable   : 0
max_depth         : 3
cur_item          : 4
pending_favs      : 0
pending_total     : 0
stability         : 100.00%
bitmap_cvg        : 0.04%
saved_crashes     : 5
saved_hangs       : 5
total_tmout       : 118
last_find         : 1743202382
last_crash        : 1743202346
last_hang         : 1743202450
execs_since_crash : 35872
exec_timeout      : 40
slowest_exec_ms   : 0
peak_rss_mb       : 12
cpu_affinity      : 0
edges_found       : 23
total_edges       : 65536
var_byte_count    : 0
havoc_expansion   : 5
auto_dict_entries : 0
testcache_size    : 524
testcache_count   : 13
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
[AFL++ d8ac67853932] /src/output/default #
```

Crash Input

```
eeeeeee%ee%
```

```
[AFL++ d8ac67853932] /src/output/default # cd crashes/
[AFL++ d8ac67853932] /src/output/default/crashes # ls
README.txt                                          id:000001,sig:10,src:000000,time:8491,execs:1928,op:havoc,rep:8   id:000003,sig:10,src:000000,time:18106,execs:4699,op:havoc,rep:4
id:000000,sig:10,src:000000,time:1555,execs:161,op:havoc,rep:6  id:000002,sig:10,src:000000,time:14773,execs:3734,op:havoc,rep:7  id:000004,sig:10,src:000000,time:19433,execs:5095,op:havoc,rep:7
[AFL++ d8ac67853932] /src/output/default/crashes # cat id:000000,sig:10,src:000000,time:1555,execs:161,op:havoc,rep:6
[AFL++ d8ac67853932] /src/output/default/crashes # cat id:000001,sig:10,src:000000,time:8491,execs:1928,op:havoc,rep:8:000000,time:8491,execs:1928,op:havoc,rep:8
[AFL++ d8ac67853932] /src/output/default/crashes # cat id:000003,sig:10,src:000000,time:18106,execs:4699,op:havoc,rep:4s:4699,op:havoc,rep:4
eeeeeee%ee%[AFL++ d8ac67853932] /src/output/default/crashes #
```

Hang Input

```
+ttpp
att%eeettt%eet$+V
```

```
[AFL++ d8ac67853932] /src/output/default # cd hangs/
[AFL++ d8ac67853932] /src/output/default/hangs # ls
id:000000,src:000000,time:1449,execs:127,op:havoc,rep:5  id:000002,src:000000,time:26734,execs:6892,op:havoc,rep:8  id:000004,src:000007,time:123248,execs:39270,op:havoc,rep:29
id:000001,src:000000,time:4264,execs:672,op:havoc,rep:4  id:000003,src:000000,time:37040,execs:9798,op:havoc,rep:6
[AFL++ d8ac67853932] /src/output/default/hangs # cat id:000000,src:000000,time:1449,execs:127,op:havoc,rep:5
+ttpp[AFL++ d8ac67853932] /src/output/default/hangs #
```

Reproduce Crash & Hang

Case	Input	Root Cause	Fix
Crash	`%`	Invalid checking input length	Check `i + 2 < len(s)` before parsing `%`
Hang	`+`	Missing `i += 1` → infinite loop	Add `i += 1` after `ret.append(' ')`

Theory & Reflection

Will the fuzzer ever terminate?

> **No.** AFL++ runs indefinitely, continuously mutating inputs unless explicitly stopped.

How do coverage-guided fuzzers work?

They monitor which code branches are triggered by each input and prioritize mutations that explore new paths.

Yes, AFL++ is coverage-guided—it uses edge coverage to guide input generation.

How to optimize a fuzzing campaign?

> Use high-quality seed inputs, run in parallel, reduce redundant/crashing inputs, and monitor coverage growth.

Link to Github Lab