Docker image

We poublish a docker image on docker hub. The size of this image is about 14.7GB.


```
$ sudo docker pull yiruzhaozhao/alphuzz:artifact

$ sudo docker run --privileged -it yiruzhaozhao/alphuzz:artifact /bin/bash
```

We put the datasets binaries and initial seeds under the root directory.

We put the datasets binaries and initial seeds under the */home* directory.

Example

Take Unifuzz dataset for example.

Alphuzz

```
$ /home/Alphuzz-main/afl-fuzz -i /unifuzz/seeds/exiv2 -o /home/out -Q --
/unifuzz/binaries/exiv2 @@
```

```
root@e17cfb1ea93f:/# /home/Alphuzz-main/afl-fuzz -i /unifuzz/seeds/exiv2/ -o /ho
me/out  -Q -- /unifuzz/binaries/exiv2 @@
afl-fuzz 2.52b by <lcamtuf@google.com>
[+] You have 8 CPU cores and 2 runnable tasks (utilization: 25%).
[+] Try parallel jobs - see docs/parallel_fuzzing.txt.
[*] Checking CPU core loadout...
[+] Found a free CPU core, binding to #0.
[*] Checking core_pattern...
[*] Setting up output directories...
[+] Output directory exists but deemed OK to reuse.
[*] Deleting old session data...
[+] Output dir cleanup successful.
[*] Scanning '/unifuzz/seeds/exiv2/'...
[+] No auto-generated dictionary tokens to reuse.
[*] Creating hard links for all input files...
[*] Validating target binary...
[*] Attempting dry run with 'id:000000,orig:103.jpg'...
```

```
                      american fuzzy lop 2.52b (exiv2)
┌─ process timing ──────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 0 hrs, 3 min, 30 sec │    cycles done : 0   │
│   last new path : none seen yet             │    total paths : 100 │
│ last uniq crash : none seen yet             │ uniq crashes : 0     │
│  last uniq hang : none seen yet             │    uniq hangs : 0    │
├─ cycle progress ─────────────┬─ map coverage ┤
│  now processing : 0* (0.00%)        │     map density : 12.44% / 12.50% │
│ paths timed out : 0 (0.00%)         │ count coverage : 1.02 bits/tuple │
├─ stage progress ─────────────┼─ findings in depth ─┤
│  now trying : trim 256/256          │ favored paths : 0 (0.00%)   │
│ stage execs : 109/122 (89.34%)      │  new edges on : 1 (1.00%)   │
│ total execs : 1048                  │ total crashes : 0 (0 unique) │
│  exec speed : 5.08/sec (zzzz...)    │  total tmouts : 0 (0 unique) │
├─ fuzzing strategy yields ───────────┴─ path geometry ─┤
│   bit flips : 0/0, 0/0, 0/0         │     levels : 1   │
│  byte flips : 0/0, 0/0, 0/0         │    pending : 100 │
│ 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         │   imported : n/a │
│       havoc : 0/0, 0/0              │  stability : 96.63% │
│        trim : n/a, n/a              └──────────────────┤
│                                         [cpu000: 54%] │
└───────────────────────────────────────────────────────┘
```

Alphuzzplusplus

```
$ /home/Alphuzzplusplus-main/afl-fuzz -i /unifuzz/seeds/cflow -o /home/out -Q -t
3000+ -- /unifuzz/binaries/cflow @@
```

```
root@e17cfb1ea93f:/# /home/Alphuzzplusplus-main/afl-fuzz -i /unifuzz/seeds/cflow
/ -o /home/out -t 3000+ -Q -- /unifuzz/binaries/cflow @@
afl-fuzz++3.14a based on afl by Michal Zalewski and a large online community
[+] afl++ is maintained by Marc "van Hauser" Heuse, Heiko "hexcoder" Eißfeldt, A
ndrea Fioraldi and Dominik Maier
[+] afl++ is open source, get it at https://github.com/AFLplusplus/AFLplusplus
[+] NOTE: This is v3.x which changes defaults and behaviours - see README.md
[+] No -M/-S set, autoconfiguring for "-S default"
[*] Getting to work...
[+] Using exponential power schedule (FAST)
[+] Enabled testcache with 50 MB
[*] Checking core_pattern...
[!] WARNING: Could not check CPU scaling governor
[+] You have 8 CPU cores and 4 runnable tasks (utilization: 50%).
[+] Try parallel jobs - see docs/parallel_fuzzing.md.
[*] Setting up output directories...
[*] Checking CPU core loadout...
[+] Found a free CPU core, try binding to #0.
[*] Scanning '/unifuzz/seeds/cflow/'...
[+] Loaded a total of 100 seeds
```

```
                  american fuzzy lop ++3.14a (default) [fast] {0}
┌─ process timing ─────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 0 min, 2 sec │    cycles done : 0   │
│   last new path : 0 days, 0 hrs, 0 min, 0 sec │    total paths : 117 │
│ last uniq crash : none seen yet          │   uniq crashes : 0   │
│  last uniq hang : none seen yet          │     uniq hangs : 0   │
├─ cycle progress ─────────────┬─ map coverage┤
│  now processing : 18.0 (15.4%)  │      map density : 1.91% / 4.52%    │
│ paths timed out : 0 (0.00%)     │   count coverage : 4.66 bits/tuple  │
├─ stage progress ─────────────┼─ findings in depth ─────────────┤
│  now trying : havoc             │   favored paths : 34 (29.06%)       │
│ stage execs : 30/1824 (1.64%)   │    new edges on : 45 (38.46%)       │
│ total execs : 1532              │   total crashes : 0 (0 unique)      │
│  exec speed : 167.4/sec         │    total tmouts : 0 (0 unique)      │
├─ fuzzing strategy yields ──────────────────┬─ path geometry ──────┤
│   bit flips : disabled (default, enable with -D) │     levels : 2    │
│  byte flips : disabled (default, enable with -D) │    pending : 117  │
│ arithmetics : disabled (default, enable with -D) │   pend fav : 34   │
│  known ints : disabled (default, enable with -D) │  own finds : 16   │
│  dictionary : n/a                                │   imported : 0    │
│ havoc/splice : 0/0, 0/0                          │  stability : 97.33% │
│ py/custom/rq : unused, unused, unused, unused    │                   │
│    trim/eff : 0.00%/13, disabled                 │   [cpu000: 37%]   │
└──────────────────────────────────────────────────┴───────────────────┘
```