Hw 2. Fuzzing

Task1 – CVE-2014-0160 (openssl)[1]

Heartbleed was a security bug in the OpenSSL cryptography library (CVE-2014-0160), which implements the Transport Layer Security (TLS) protocol.
In this task, you need to find the bug with AFL.

Reference steps

1. Configure and build with ASAN
   ```
   CC=afl-clang-fast CXX=afl-clang-fast++ ./config -d
   AFL_USE_ASAN=1 make
   ```
2. Complete the harness (`harness.cc`)
   (Of course, you can write your own harness.)
3. Compile the harness
   (e.g.)
   ```
   AFL_USE_ASAN=1 afl-clang-fast++ -g harness.cc openssl-1.0.1f/libssl.a openssl-1.0.1f/libcrypto.a -o handshake -I openssl-1.0.1f/include -ldl
   ```
4. Fuzzing
5. Analyze the crashes

Task2 – CVE-2009-0159 (ntpq)[2]

ntpq is a utility included in NTP Reference Implementation suite of tools. It queries the NTP server (e.g. ntpd) to synchronize the local clock.
In this task, you need to trigger the CVE-2009-0159 bug in ntpq with AFL.

Hint

- See https://xorl.wordpress.com/2009/04/13/cve-2009-0159-ntp-remote-stack-overflow for more information about CVE-2009-0159
- In this task, instead of trying to have AFL's output simulate a remote ntpd, you can directly replace `ntpq/ntpq.c`'s `main()` function with code that calls `cookedprint` with `datatype`, `status`, and `data` all read in from stdin, and the output file as stdout.

Reference steps

[1] https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-0160
[2] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0159

1. Write the customized `ntpq/ntpq.c`'s `main()` function (see Hint)
2. Configure and build ntpq

```
CC=afl-clang-fast ./configure
```

`make -C ntpq` (compile ntpq itself rather than compiling whole source code)

3. Fuzzing
4. Analyze the crashes

Include the following in your homework submission

Program code

- In each task, you should submit the harness that you have used for the fuzzing.
- The format of filename is: `harness_task1.c` / `harness_task2.c`

Seed

- In each task, you should submit the seeds that you have used for the fuzzing.
- Please compress them as a zip file.
- The format of filename is: `seed_task1.zip` / `seed_task2.zip`

Report

For each task, you should indicate the following items in your report:

- Key steps in the fuzzing process. You can use screenshots along with short descriptions.
- Briefly explain the code structure and usage of your harness code.
- How do you create the seeds? Are some of the seeds more effective than the other (i.e., allowing AFL to find more paths in a shorter time)?
- Verify that the crash you found correspond to the CVE number given in the homework description. If you find a new crash site (a new bug), please describe the bug you find.
- Techniques or tricks you used to improve performance of fuzzing (if any).

The format of filename is: `report_studentID.pdf`