Assignment 4 report

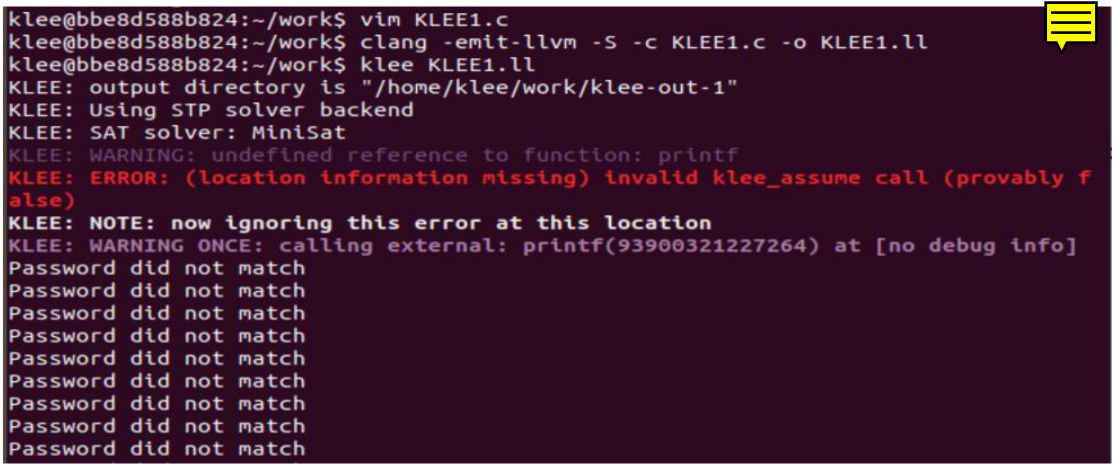
Yan Duan & Qinhuai Xu

2. PasswordCheck.c for KLEE

B) *Run KLEE on the modified program, with default options*

i) How many bugs are detected by KLEE? Explain the nature of all detected bug(s). If not, explain why no bugs were found.

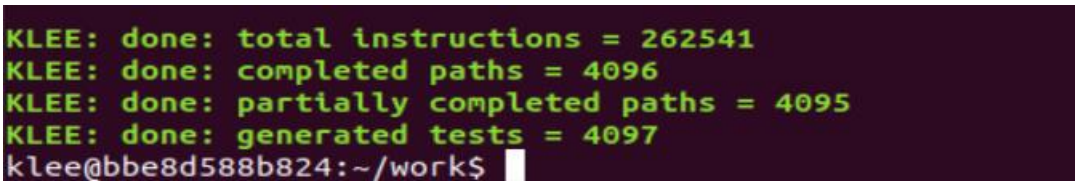
- There is 1 bug in detected by KLEE, which is invalid klee_assume call. The bug is triggered because the assertion (pwdChar >= 'a' && pwdChar <= 'z') is evaluated as false in running.



```
klee@bbe8d588b824:~/work$ vim KLEE1.c
klee@bbe8d588b824:~/work$ clang -emit-llvm -S -c KLEE1.c -o KLEE1.ll
klee@bbe8d588b824:~/work$ klee KLEE1.ll
KLEE: output directory is "/home/klee/work/klee-out-1"
KLEE: Using STP solver backend
KLEE: SAT solver: MiniSat
KLEE: WARNING: undefined reference to function: printf
KLEE: ERROR: (location information missing) invalid klee_assume call (provably f
alse)
KLEE: NOTE: now ignoring this error at this location
KLEE: WARNING ONCE: calling external: printf(93900321227264) at [no debug info]
Password did not match
Password did not match
Password did not match
Password did not match
Password did not match
Password did not match
Password did not match
Password did not match
Password did not match
```

ii) How many total paths were explored by KLEE?

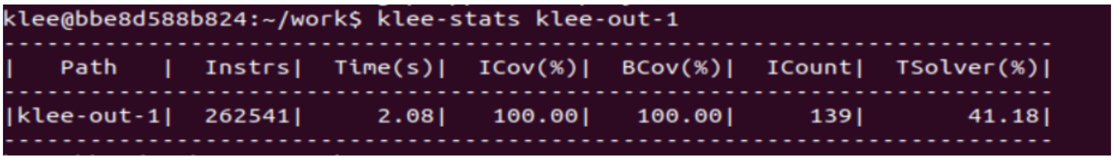
- There are totally 4096 completed paths and 4095 partially completed paths were explored by KLEE.



```
KLEE: done: total instructions = 262541
KLEE: done: completed paths = 4096
KLEE: done: partially completed paths = 4095
KLEE: done: generated tests = 4097
klee@bbe8d588b824:~/work$
```

iii) Approximately, how long did KLEE take to run?

- 2.08 seconds.



```
klee@bbe8d588b824:~/work$ klee-stats klee-out-1
-----------------------------------------------------------------------------
|   Path    |  Instrs|  Time(s)|  ICov(%)|  BCov(%)|  ICount|  TSolver(%)|
-----------------------------------------------------------------------------
|klee-out-1|  262541|     2.08|   100.00|   100.00|     139|       41.18|
-----------------------------------------------------------------------------
```

iv) Why is KLEE considered a concolic execution tool rather than a pure symbolic execution tool? What are the differences?

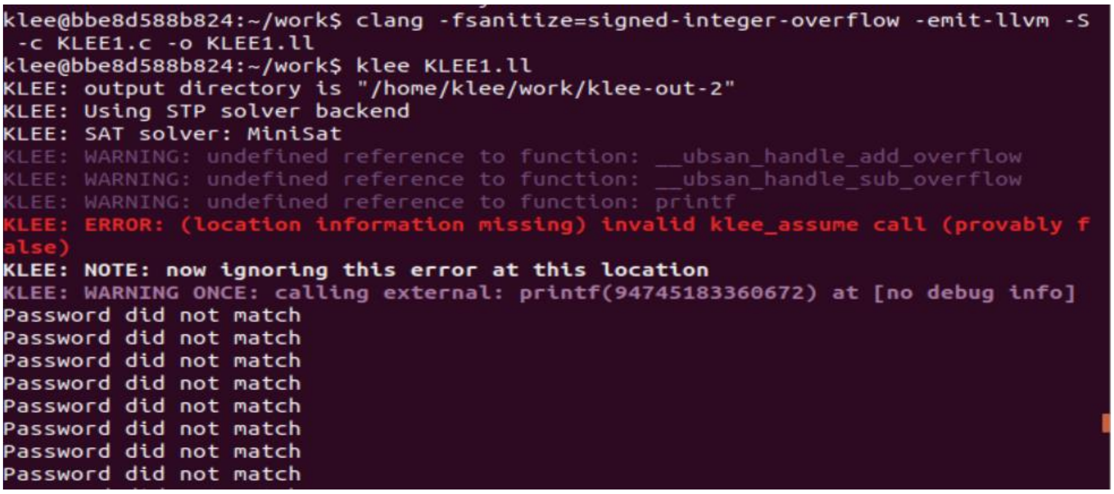
- Concolic execution is a combination of both concrete (real-world values) and symbolic (variables represented as symbols) execution, whereas pure symbolic execution relies solely on symbolic values.
- KLEE is considered as a concolic execution tool because it combines concrete execution and

symbolic execution in its analysis. KLEE starts its analysis by executing the program using concrete input values. As the program executes, KLEE collects symbolic constraints on the input variables by symbolic execution.

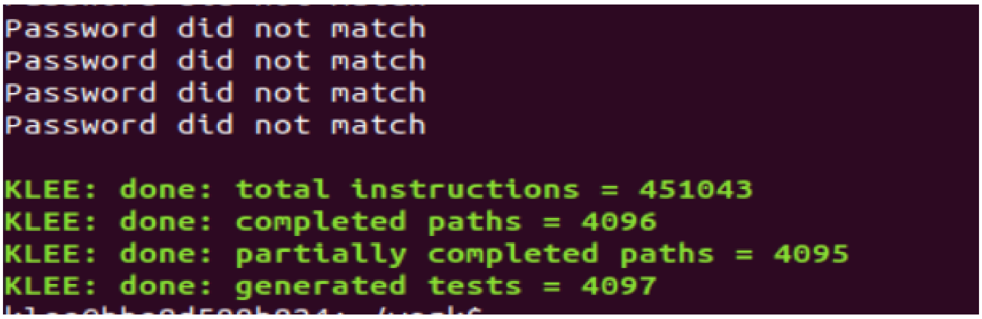
C) *Now apply a sanitizer, by passing in fsanitize=signed-integer-overflow when building KLEE1.c*

i) What is the effect on KLEE by applying this sanitizer, compared to your previous observations without sanitizers? Is KLEE able to find any new bugs? Does KLEE explore additional paths? Does KLEE take longer to run?

- Compared to previous observations without sanitizers, the number of total instructions is larger, with 451043 than 262541.
- KLEE does not find any new bugs.
- KLEE does not explore additional paths.
- KLEE takes longer to run than previous observation, with 3.30 seconds.

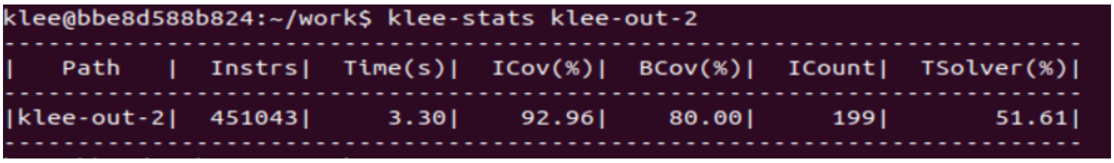


```
klee@bbe8d588b824:~/work$ clang -fsanitize=signed-integer-overflow -emit-llvm -S
 -c KLEE1.c -o KLEE1.ll
klee@bbe8d588b824:~/work$ klee KLEE1.ll
KLEE: output directory is "/home/klee/work/klee-out-2"
KLEE: Using STP solver backend
KLEE: SAT solver: MiniSat
KLEE: WARNING: undefined reference to function: __ubsan_handle_add_overflow
KLEE: WARNING: undefined reference to function: __ubsan_handle_sub_overflow
KLEE: WARNING: undefined reference to function: printf
KLEE: ERROR: (location information missing) invalid klee_assume call (provably f
alse)
KLEE: NOTE: now ignoring this error at this location
KLEE: WARNING ONCE: calling external: printf(94745183360672) at [no debug info]
Password did not match
Password did not match
Password did not match
Password did not match
Password did not match
Password did not match
Password did not match
Password did not match
```



```
Password did not match
Password did not match
Password did not match
Password did not match

KLEE: done: total instructions = 451043
KLEE: done: completed paths = 4096
KLEE: done: partially completed paths = 4095
KLEE: done: generated tests = 4097
```



```
klee@bbe8d588b824:~/work$ klee-stats klee-out-2
-------------------------------------------------------------------------
|   Path    |  Instrs|  Time(s)|  ICov(%)|  BCov(%)|  ICount|  TSolver(%)|
-------------------------------------------------------------------------
|klee-out-2|  451043|     3.30|    92.96|    80.00|     199|       51.61|
-------------------------------------------------------------------------
```

ii) How does a sanitizer work?

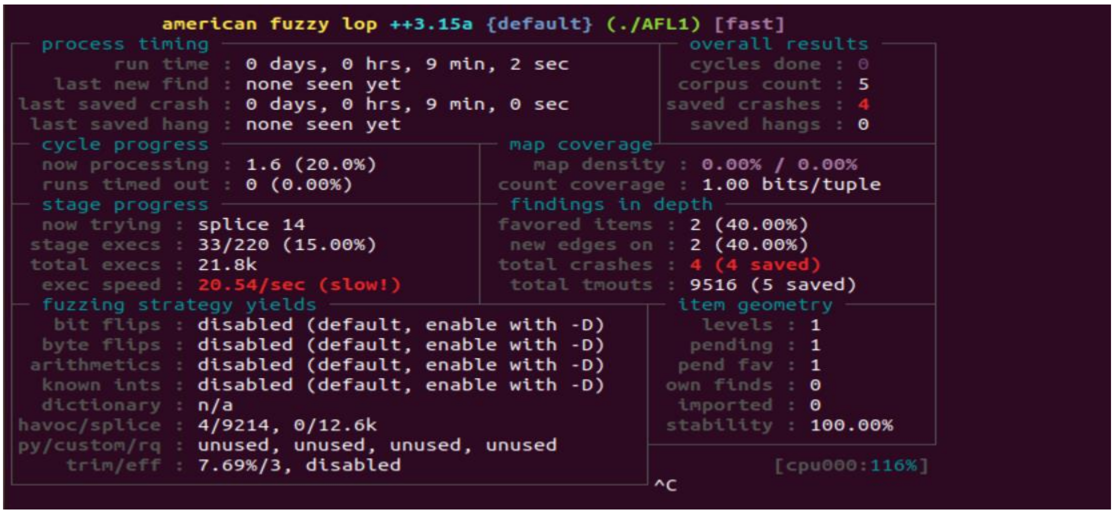
The sanitizer works by adding checks to detect specific problematic condition. For example, when you enable *fsanitize=signed-integer-overflow*, the compiler adds checks to the program's signed integer operations. During program execution, the instrumented code will perform runtime checks to detect signed integer overflows. If an overflow occurs, the sanitizer will intercept it and generate an alert and report relevant information.

3. PasswordCheck.c for AFL

B) *Run AFL on the modified program, with default options. You must supply an input folder with seed files. Start with minimal number of seed files.*

i) How many crashes and hangs were encountered by AFL?

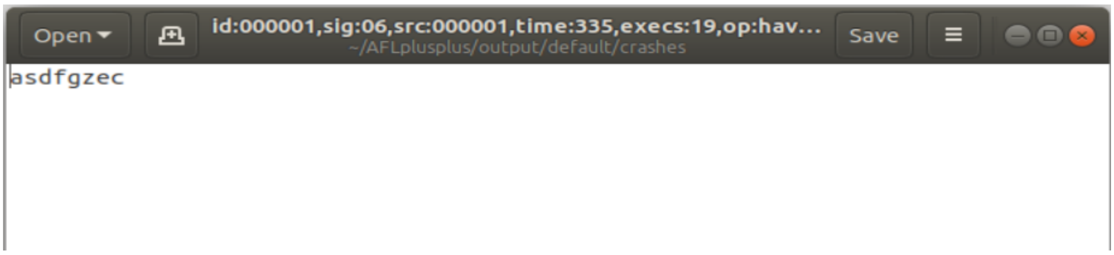
- There are 4 crashes, and 0 hangs were encountered by AFL.



```
             american fuzzy lop ++3.15a {default} (./AFL1) [fast]
┌─ process timing ────────────────────────────────────┬─ overall results ──┐
│        run time : 0 days, 0 hrs, 9 min, 2 sec        │  cycles done : 0    │
│   last new find : none seen yet                      │ corpus count : 5    │
│last saved crash : 0 days, 0 hrs, 9 min, 0 sec        │saved crashes : 4    │
│ last saved hang : none seen yet                      │  saved hangs : 0    │
├─ cycle progress ─────────────────────┬─ map coverage─┴────────────────────┤
│  now processing : 1.6 (20.0%)        │    map density : 0.00% / 0.00%     │
│  runs timed out : 0 (0.00%)          │ count coverage : 1.00 bits/tuple   │
├─ stage progress ─────────────────────┼─ findings in depth ────────────────┤
│  now trying : splice 14              │ favored items : 2 (40.00%)         │
│ stage execs : 33/220 (15.00%)        │  new edges on : 2 (40.00%)         │
│ total execs : 21.8k                  │ total crashes : 4 (4 saved)        │
│  exec speed : 20.54/sec (slow!)      │  total tmouts : 9516 (5 saved)     │
├─ fuzzing strategy yields ────────────┴───────────┬─ item geometry ────────┤
│   bit flips : disabled (default, enable with -D)  │    levels : 1          │
│  byte flips : disabled (default, enable with -D)  │   pending : 1          │
│ arithmetics : disabled (default, enable with -D)  │  pend fav : 1          │
│  known ints : disabled (default, enable with -D)  │ own finds : 0          │
│  dictionary : n/a                                 │  imported : 0          │
│havoc/splice : 4/9214, 0/12.6k                     │ stability : 100.00%    │
│py/custom/rq : unused, unused, unused, unused      ├────────────────────────┘
│    trim/eff : 7.69%/3, disabled                   │          [cpu000:116%]
└───────────────────────────────────────────────────┘^C
```

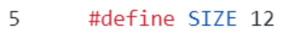
ii) How many bugs are detected by AFL? Explain the nature of all detected bug(s). If not, explain why no bugs were found.

- To find bugs, we try to check crashes files and find the crash is triggered by “asdfgzec” input. Because the length of “asdfgzec” is 8, which is less than the defined SIZE 12 in line 5 and cause password array index out of bound bug in line 14.

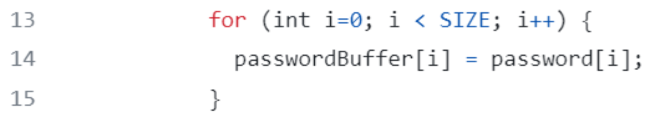


id:000001,sig:06,src:000001,time:335,execs:19,op:hav...
~/AFLplusplus/output/default/crashes

```
asdfgzec
```



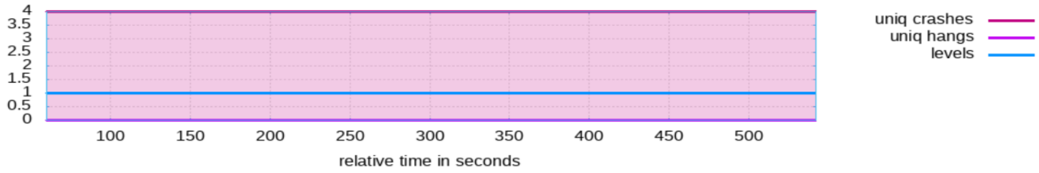
```
5      #define SIZE 12
```



```
13          for (int i=0; i < SIZE; i++) {
14            passwordBuffer[i] = password[i];
15          }
```

iii) How long did AFL take to encounter its first crash or hang, if one was ever found?

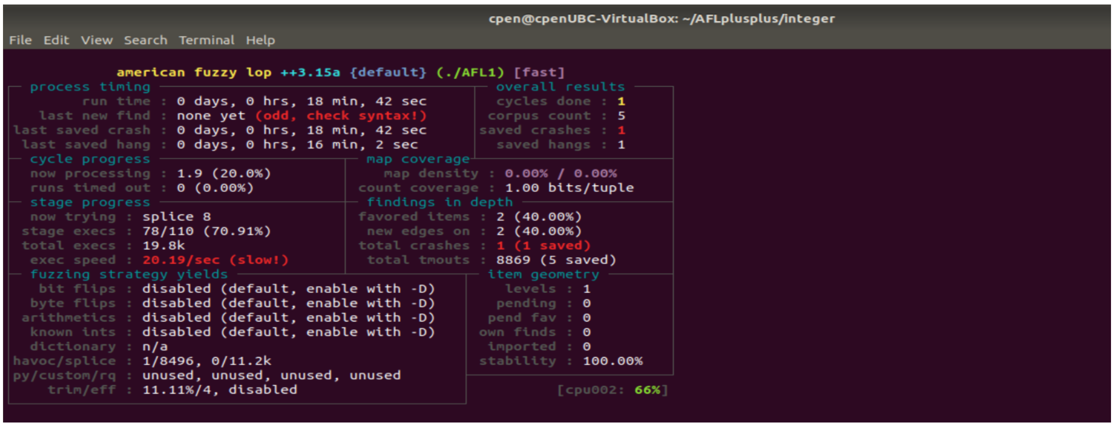
- AFL encountered its first crash very soon, in a few seconds.



C) *Now apply sanitizers to AFL, by passing in fsanitize=signed-integer-overflow, address, undefined*

i) How many crashes and hangs were detected by AFL, with sanitizers, compared to the program without sanitzers?

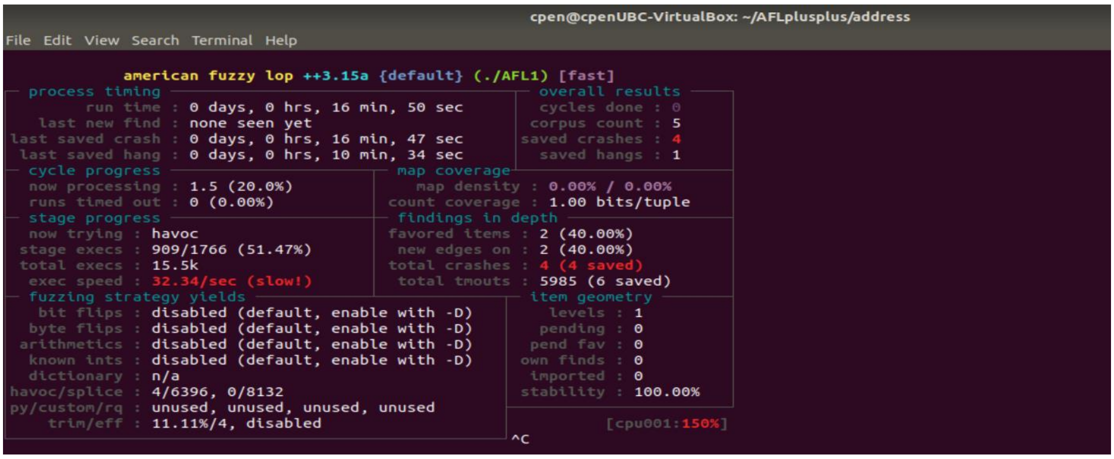
- 1 crash and 1 hang were detected by AFL with sanitizer *signed-integer-overflow*



```
cpen@cpenUBC-VirtualBox: ~/AFLplusplus/integer

         american fuzzy lop ++3.15a {default} (./AFL1) [fast]
┌─ process timing ────────────────────────────────────┬─ overall results ─┐
│        run time : 0 days, 0 hrs, 18 min, 42 sec     │  cycles done : 1  │
│   last new find : none yet (odd, check syntax!)     │ corpus count : 5  │
│last saved crash : 0 days, 0 hrs, 18 min, 42 sec     │saved crashes : 1  │
│ last saved hang : 0 days, 0 hrs, 16 min, 2 sec      │  saved hangs : 1  │
├─ cycle progress ─────────────────────┬─ map coverage┴───────────────────┤
│  now processing : 1.9 (20.0%)        │    map density : 0.00% / 0.00%  │
│  runs timed out : 0 (0.00%)          │ count coverage : 1.00 bits/tuple │
├─ stage progress ─────────────────────┼─ findings in depth ──────────────┤
│  now trying : splice 8               │ favored items : 2 (40.00%)       │
│ stage execs : 78/110 (70.91%)        │  new edges on : 2 (40.00%)       │
│ total execs : 19.8k                  │ total crashes : 1 (1 saved)      │
│  exec speed : 20.19/sec (slow!)      │  total tmouts : 8869 (5 saved)   │
├─ fuzzing strategy yields ────────────┴───────────┬─ item geometry ───────┤
│   bit flips : disabled (default, enable with -D) │    levels : 1         │
│  byte flips : disabled (default, enable with -D) │   pending : 0         │
│ arithmetics : disabled (default, enable with -D) │  pend fav : 0         │
│  known ints : disabled (default, enable with -D) │ own finds : 0         │
│  dictionary : n/a                                │  imported : 0         │
│havoc/splice : 1/8496, 0/11.2k                    │ stability : 100.00%   │
│py/custom/rq : unused, unused, unused, unused     ├───────────────────────┘
│    trim/eff : 11.11%/4, disabled                 │          [cpu002: 66%]
└──────────────────────────────────────────────────┘
```

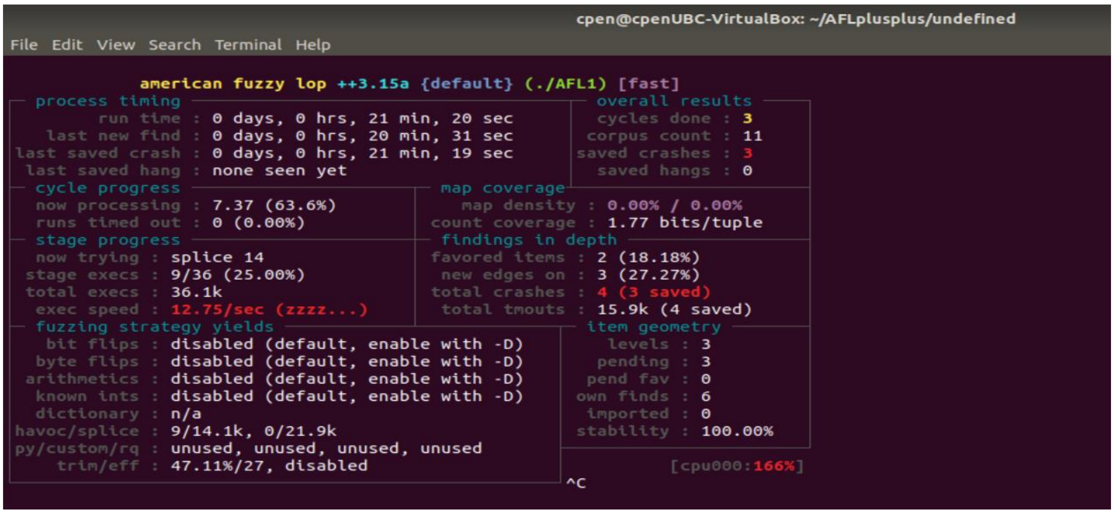
- 4 crashes and 1 hang were detected by AFL with sanitizer *address*



```
cpen@cpenUBC-VirtualBox: ~/AFLplusplus/address

         american fuzzy lop ++3.15a {default} (./AFL1) [fast]
┌─ process timing ────────────────────────────────────┬─ overall results ─┐
│        run time : 0 days, 0 hrs, 16 min, 50 sec     │  cycles done : 0  │
│   last new find : none seen yet                     │ corpus count : 5  │
│last saved crash : 0 days, 0 hrs, 16 min, 47 sec     │saved crashes : 4  │
│ last saved hang : 0 days, 0 hrs, 10 min, 34 sec     │  saved hangs : 1  │
├─ cycle progress ─────────────────────┬─ map coverage┴───────────────────┤
│  now processing : 1.5 (20.0%)        │    map density : 0.00% / 0.00%  │
│  runs timed out : 0 (0.00%)          │ count coverage : 1.00 bits/tuple │
├─ stage progress ─────────────────────┼─ findings in depth ──────────────┤
│  now trying : havoc                  │ favored items : 2 (40.00%)       │
│ stage execs : 909/1766 (51.47%)      │  new edges on : 2 (40.00%)       │
│ total execs : 15.5k                  │ total crashes : 4 (4 saved)      │
│  exec speed : 32.34/sec (slow!)      │  total tmouts : 5985 (6 saved)   │
├─ fuzzing strategy yields ────────────┴───────────┬─ item geometry ───────┤
│   bit flips : disabled (default, enable with -D) │    levels : 1         │
│  byte flips : disabled (default, enable with -D) │   pending : 0         │
│ arithmetics : disabled (default, enable with -D) │  pend fav : 0         │
│  known ints : disabled (default, enable with -D) │ own finds : 0         │
│  dictionary : n/a                                │  imported : 0         │
│havoc/splice : 4/6396, 0/8132                     │ stability : 100.00%   │
│py/custom/rq : unused, unused, unused, unused     ├───────────────────────┘
│    trim/eff : 11.11%/4, disabled                 │          [cpu001:150%]
└──────────────────────────────────────────────────┘^C
```

- 3 crashes and 0 hang were detected by AFL with sanitizer *undefined*


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/undefined
File Edit View Search Terminal Help

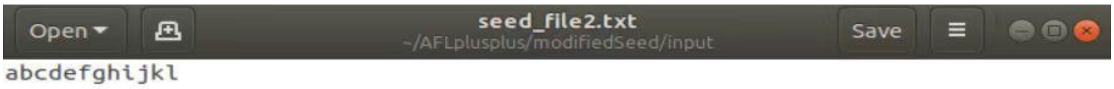
```
          american fuzzy lop ++3.15a {default} (./AFL1) [fast]
┌─ process timing ─────────────────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 21 min, 20 sec      │  cycles done : 3     │
│   last new find : 0 days, 0 hrs, 20 min, 31 sec      │ corpus count : 11    │
│last saved crash : 0 days, 0 hrs, 21 min, 19 sec      │saved crashes : 3     │
│ last saved hang : none seen yet                      │  saved hangs : 0     │
├─ cycle progress ─────────────────────┬─ map coverage─┴──────────────────────┤
│  now processing : 7.37 (63.6%)       │    map density : 0.00% / 0.00%      │
│  runs timed out : 0 (0.00%)          │ count coverage : 1.77 bits/tuple    │
├─ stage progress ─────────────────────┼─ findings in depth ──────────────────┤
│  now trying : splice 14              │ favored items : 2 (18.18%)          │
│ stage execs : 9/36 (25.00%)          │  new edges on : 3 (27.27%)          │
│ total execs : 36.1k                  │ total crashes : 4 (3 saved)         │
│  exec speed : 12.75/sec (zzzz...)    │  total tmouts : 15.9k (4 saved)     │
├─ fuzzing strategy yields ────────────┴───────────┬─ item geometry ──────────┤
│   bit flips : disabled (default, enable with -D) │    levels : 3            │
│  byte flips : disabled (default, enable with -D) │   pending : 3            │
│ arithmetics : disabled (default, enable with -D) │  pend fav : 0            │
│  known ints : disabled (default, enable with -D) │ own finds : 6            │
│  dictionary : n/a                                │  imported : 0            │
│havoc/splice : 9/14.1k, 0/21.9k                   │ stability : 100.00%      │
│py/custom/rq : unused, unused, unused, unused     ├──────────────────────────┘
│    trim/eff : 47.11%/27, disabled                │          [cpu000:166%]
└──────────────────────────────────────────────────┘^C
```

ii) How many bugs are detected by AFL? Explain the nature of all detected bug(s). If not, explain why no bugs were found.

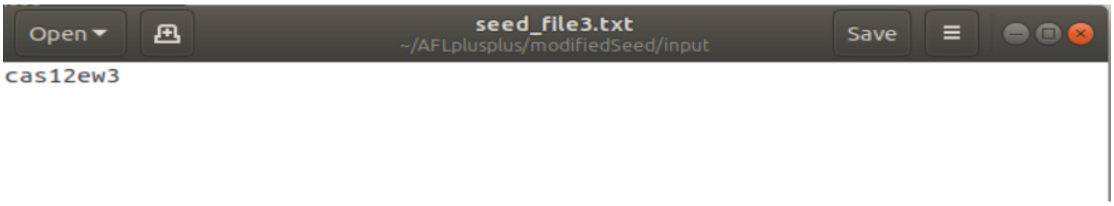
- We try to check the crashes files and they are all same issues as previous. Therefore, no new bugs are detected by AFL.

iii) Try modifying the seed input files, and/or adding more input seed files. Does this change the AFL results in any way?

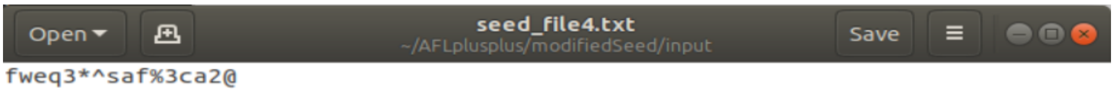
- We try to add 3 more input seed files, which is seed_file2.txt, seed_file3.txt, and seed_file4.txt.
  - seed_file2.txt contains characters.
  - seed_file3.txt contains characters and digits.
  - seed_file4.txt contains more special characters.



```
abcdefghijkl
```

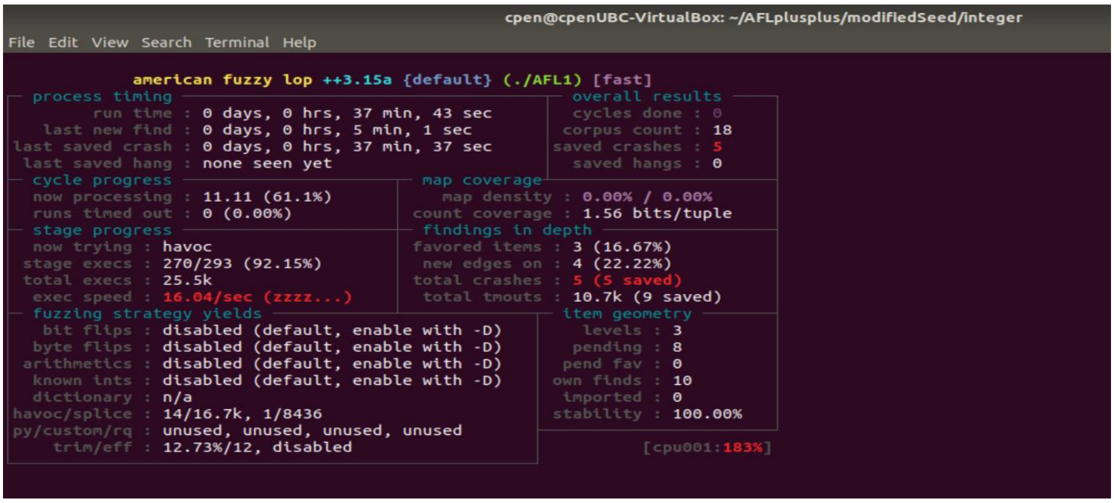


```
cas12ew3
```



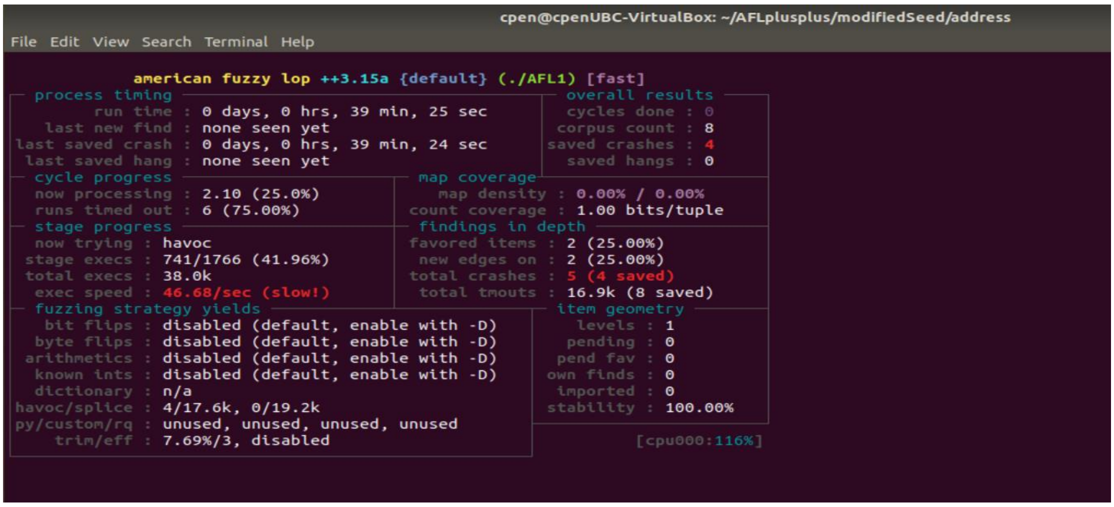
```
fweq3*^saf%3ca2@
```

- After adding more seeds, 5 crashes and 0 hang were detected by AFL with sanitizer *signed-integer-overflow*.


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/modifiedSeed/integer
File Edit View Search Terminal Help

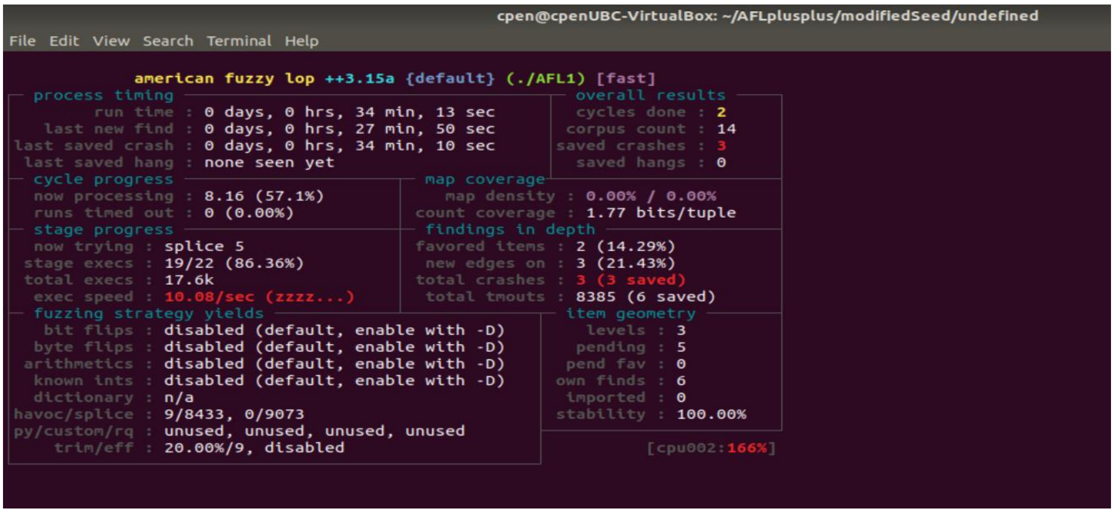
```
            american fuzzy lop ++3.15a {default} (./AFL1) [fast]
  process timing                                       overall results
        run time : 0 days, 0 hrs, 37 min, 43 sec        cycles done : 0
   last new find : 0 days, 0 hrs, 5 min, 1 sec         corpus count : 18
 last saved crash : 0 days, 0 hrs, 37 min, 37 sec     saved crashes : 5
  last saved hang : none seen yet                       saved hangs : 0
  cycle progress                          map coverage
   now processing : 11.11 (61.1%)            map density : 0.00% / 0.00%
   runs timed out : 0 (0.00%)             count coverage : 1.56 bits/tuple
  stage progress                          findings in depth
   now trying : havoc                     favored items : 3 (16.67%)
  stage execs : 270/293 (92.15%)           new edges on : 4 (22.22%)
  total execs : 25.5k                     total crashes : 5 (5 saved)
   exec speed : 16.04/sec (zzzz...)        total tmouts : 10.7k (9 saved)
  fuzzing strategy yields                              item geometry
    bit flips : disabled (default, enable with -D)        levels : 3
   byte flips : disabled (default, enable with -D)       pending : 8
  arithmetics : disabled (default, enable with -D)      pend fav : 0
   known ints : disabled (default, enable with -D)     own finds : 10
   dictionary : n/a                                     imported : 0
 havoc/splice : 14/16.7k, 1/8436                       stability : 100.00%
 py/custom/rq : unused, unused, unused, unused
     trim/eff : 12.73%/12, disabled                           [cpu001:183%]
```

- After adding more seeds, 4 crashes and 0 hang were detected by AFL with sanitizer *address*.


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/modifiedSeed/address
File Edit View Search Terminal Help

```
            american fuzzy lop ++3.15a {default} (./AFL1) [fast]
  process timing                                       overall results
        run time : 0 days, 0 hrs, 39 min, 25 sec        cycles done : 0
   last new find : none seen yet                       corpus count : 8
 last saved crash : 0 days, 0 hrs, 39 min, 24 sec     saved crashes : 4
  last saved hang : none seen yet                       saved hangs : 0
  cycle progress                          map coverage
   now processing : 2.10 (25.0%)             map density : 0.00% / 0.00%
   runs timed out : 6 (75.00%)            count coverage : 1.00 bits/tuple
  stage progress                          findings in depth
   now trying : havoc                     favored items : 2 (25.00%)
  stage execs : 741/1766 (41.96%)          new edges on : 2 (25.00%)
  total execs : 38.0k                     total crashes : 5 (4 saved)
   exec speed : 46.68/sec (slow!)          total tmouts : 16.9k (8 saved)
  fuzzing strategy yields                              item geometry
    bit flips : disabled (default, enable with -D)        levels : 1
   byte flips : disabled (default, enable with -D)       pending : 0
  arithmetics : disabled (default, enable with -D)      pend fav : 0
   known ints : disabled (default, enable with -D)     own finds : 0
   dictionary : n/a                                     imported : 0
 havoc/splice : 4/17.6k, 0/19.2k                       stability : 100.00%
 py/custom/rq : unused, unused, unused, unused
     trim/eff : 7.69%/3, disabled                             [cpu000:116%]
```

- After adding more seeds, 3 crashes and 0 hang were detected by AFL with sanitizer *undefined*.


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/modifiedSeed/undefined
File Edit View Search Terminal Help

```
            american fuzzy lop ++3.15a {default} (./AFL1) [fast]
  process timing                                       overall results
        run time : 0 days, 0 hrs, 34 min, 13 sec        cycles done : 2
   last new find : 0 days, 0 hrs, 27 min, 50 sec       corpus count : 14
 last saved crash : 0 days, 0 hrs, 34 min, 10 sec     saved crashes : 3
  last saved hang : none seen yet                       saved hangs : 0
  cycle progress                          map coverage
   now processing : 8.16 (57.1%)             map density : 0.00% / 0.00%
   runs timed out : 0 (0.00%)             count coverage : 1.77 bits/tuple
  stage progress                          findings in depth
   now trying : splice 5                  favored items : 2 (14.29%)
  stage execs : 19/22 (86.36%)             new edges on : 3 (21.43%)
  total execs : 17.6k                     total crashes : 3 (3 saved)
   exec speed : 10.08/sec (zzzz...)        total tmouts : 8385 (6 saved)
  fuzzing strategy yields                              item geometry
    bit flips : disabled (default, enable with -D)        levels : 3
   byte flips : disabled (default, enable with -D)       pending : 5
  arithmetics : disabled (default, enable with -D)      pend fav : 0
   known ints : disabled (default, enable with -D)     own finds : 6
   dictionary : n/a                                     imported : 0
 havoc/splice : 9/8433, 0/9073                         stability : 100.00%
 py/custom/rq : unused, unused, unused, unused
     trim/eff : 20.00%/9, disabled                            [cpu002:166%]
```

After checking all new crashes files, they are same issues as previous. Therefore, no new bugs are detected by AFL.

iv) Having used both KLEE and AFL on the same program, which tool do you find more effective in finding bugs in this scenario? What are the advantages and disadvantages of using each tool respectively?

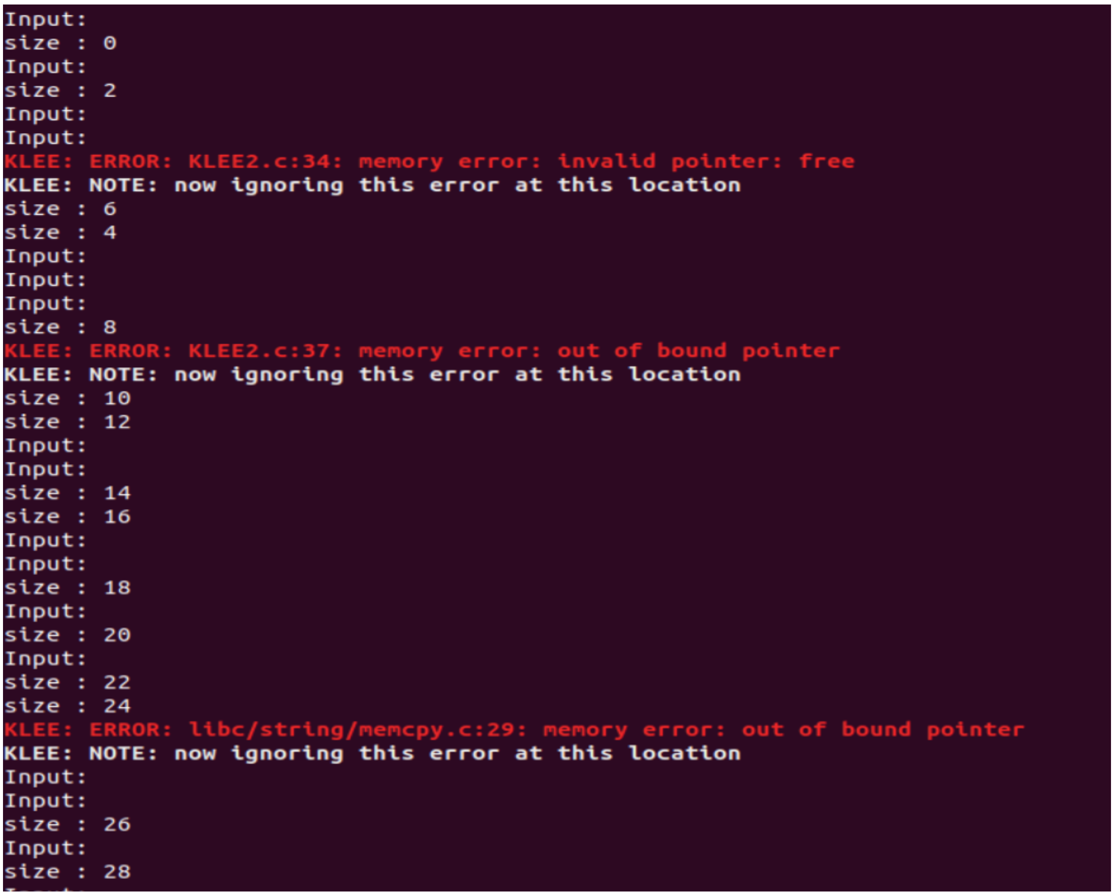
- We find that KLEE is more effective in finding bugs on *PasswordCheck.c* because KLEE takes less running time than AFL.
- KLEE:
  - Pros: Because KLEE is a concolic execution tool, it can trigger the bug with precise input values, which help developers quickly identifying and fixing issues.
  - Cons: KLEE's focus on path exploration but it may not be able to handle system calls, and external dependencies effectively like AFL.
- AFL:
  - Pros: AFL can handle system calls, and external dependencies more effectively than KLEE.
  - Cons: AFL heavily relies on random mutations of inputs, which may lead to limited path coverage. Also, the large number of test inputs will lead to longer running time.

4. Vulnerable.c for KLEE

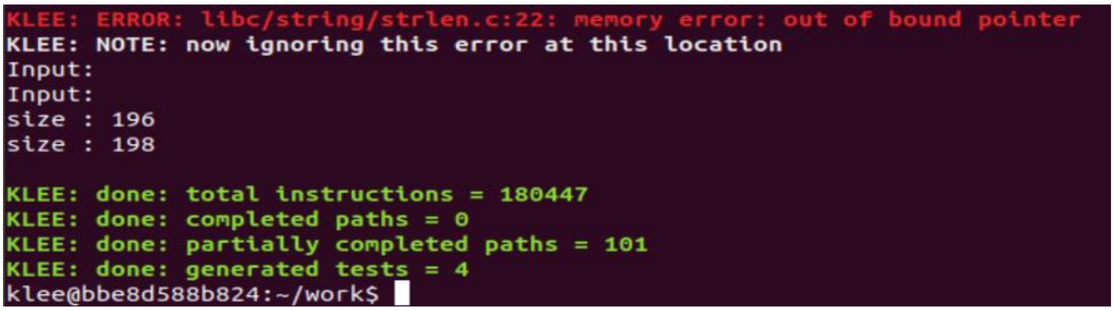
B) *Run KLEE on the modified program, with the (uClibc) C standard library enabled*

i) Explain your observations from using KLEE on this program.

- There are 4 bugs found, including invalid pointer type bugs and out of bound pointer type bugs.
- There are 101 partially completed paths and 0 completed path.
- 1.56 seconds were taken.

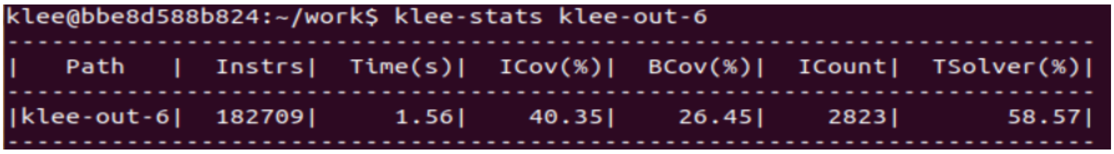


```
Input:
size : 0
Input:
size : 2
Input:
Input:
KLEE: ERROR: KLEE2.c:34: memory error: invalid pointer: free
KLEE: NOTE: now ignoring this error at this location
size : 6
size : 4
Input:
Input:
Input:
size : 8
KLEE: ERROR: KLEE2.c:37: memory error: out of bound pointer
KLEE: NOTE: now ignoring this error at this location
size : 10
size : 12
Input:
Input:
size : 14
size : 16
Input:
Input:
size : 18
Input:
size : 20
Input:
size : 22
size : 24
KLEE: ERROR: libc/string/memcpy.c:29: memory error: out of bound pointer
KLEE: NOTE: now ignoring this error at this location
Input:
Input:
size : 26
Input:
size : 28
```



```
KLEE: ERROR: libc/string/strlen.c:22: memory error: out of bound pointer
KLEE: NOTE: now ignoring this error at this location
Input:
Input:
size : 196
size : 198

KLEE: done: total instructions = 180447
KLEE: done: completed paths = 0
KLEE: done: partially completed paths = 101
KLEE: done: generated tests = 4
klee@bbe8d588b824:~/work$
```

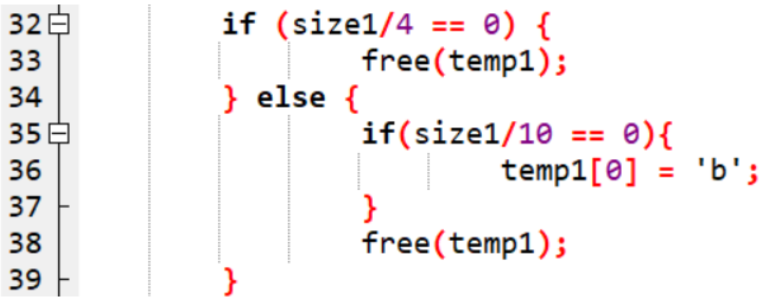


```
klee@bbe8d588b824:~/work$ klee-stats klee-out-6
------------------------------------------------------------------------
|   Path    |  Instrs|  Time(s)|  ICov(%)|  BCov(%)|  ICount|  TSolver(%)|
------------------------------------------------------------------------
|klee-out-6|  182709|     1.56|    40.35|    26.45|    2823|       58.57|
------------------------------------------------------------------------
```

ii) Try fixing a few bug(s) in the program, and recompiling the program again. Then, run KLEE on it again. Do you discover any new bugs or paths?

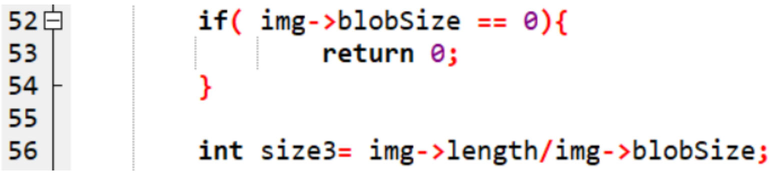
- Bug 1: free(temp1)

  Move "*free(temp1)*" on line 31 into the if sentence (line 38) to avoid being free twice.



```
	if (size1/4 == 0) {
		free(temp1);
	} else {
		if(size1/10 == 0){
			temp1[0] = 'b';
		}
		free(temp1);
	}
```

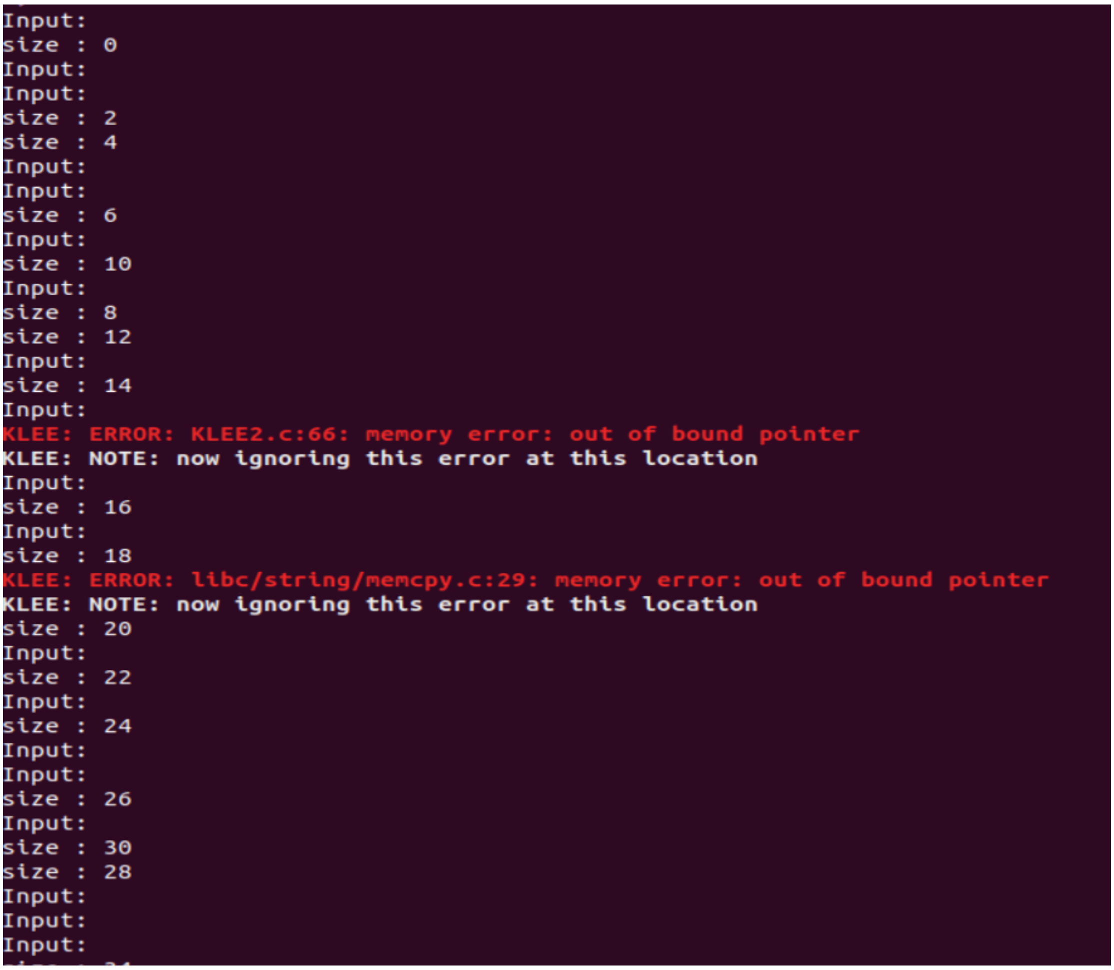
- Bug 2: 0 as divisor
  Check blobSize value before division to avoid 0 as divisor.



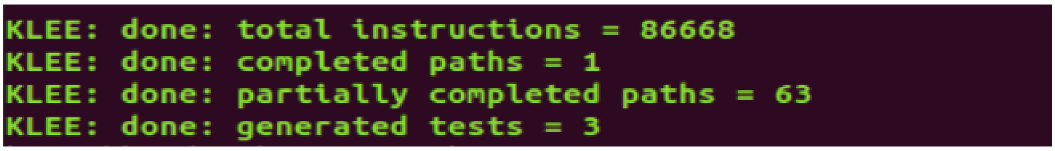
```
	if( img->blobSize == 0){
		return 0;
	}

	int size3= img->length/img->blobSize;
```

- After fixing the above 2 bugs and recompiling the program, we run KLEE on it again. We find 1 new bug and 1 remaining bug. There are 1 completed path and 63 partially completed paths.



```
Input:
size : 0
Input:
Input:
size : 2
size : 4
Input:
Input:
size : 6
Input:
size : 10
Input:
size : 8
size : 12
Input:
size : 14
Input:
KLEE: ERROR: KLEE2.c:66: memory error: out of bound pointer
KLEE: NOTE: now ignoring this error at this location
Input:
size : 16
Input:
size : 18
KLEE: ERROR: libc/string/memcpy.c:29: memory error: out of bound pointer
KLEE: NOTE: now ignoring this error at this location
size : 20
Input:
size : 22
Input:
size : 24
Input:
Input:
size : 26
Input:
size : 30
size : 28
Input:
Input:
Input:
```



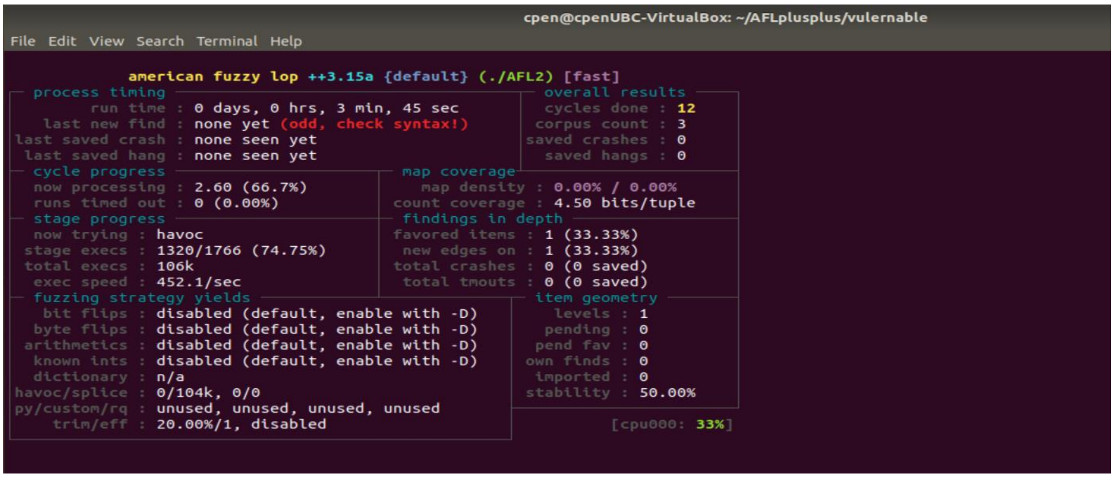
```
KLEE: done: total instructions = 86668
KLEE: done: completed paths = 1
KLEE: done: partially completed paths = 63
KLEE: done: generated tests = 3
```

### 5.	Vulnerable.c for AFL

*B)	Run AFL on the modified program*

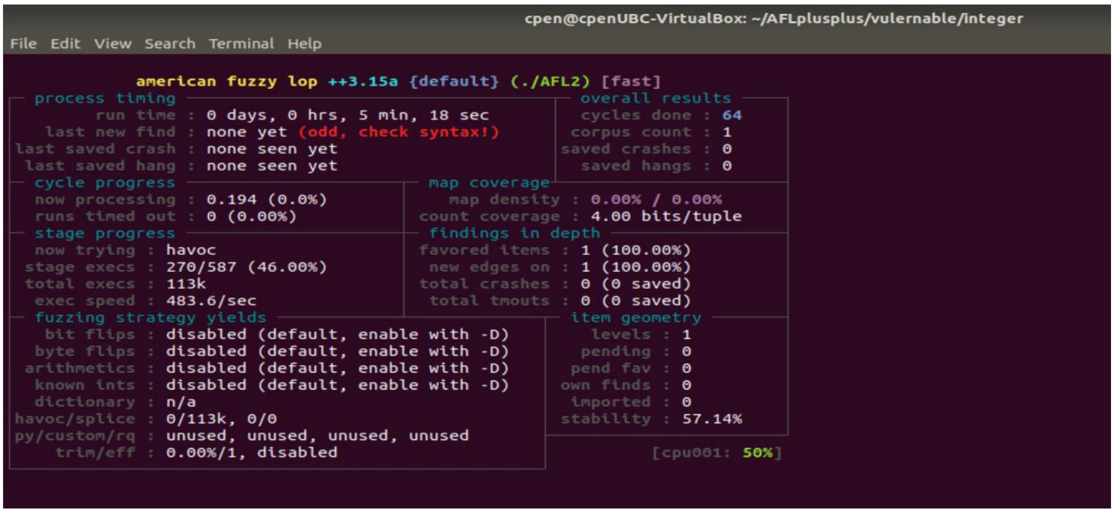
i) Explain your observations from using AFL on this program and try different sanitizers

- 0 crash and 0 hang were detected by AFL without sanitizer


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/vulernable
File Edit View Search Terminal Help

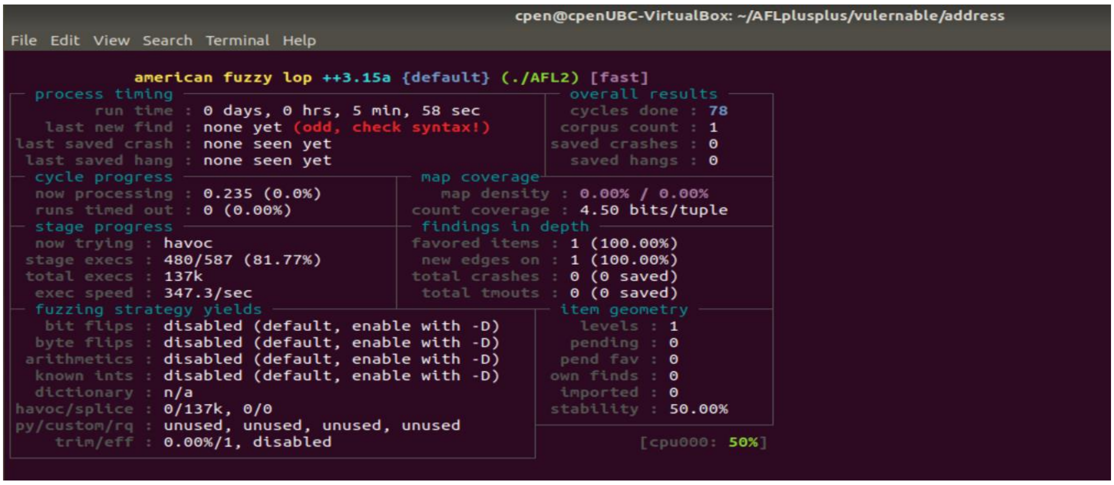
```
            american fuzzy lop ++3.15a {default} (./AFL2) [fast]
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 3 min, 45 sec      │  cycles done : 12    │
│   last new find : none yet (odd, check syntax!)     │ corpus count : 3     │
│last saved crash : none seen yet                     │saved crashes : 0     │
│ last saved hang : none seen yet                     │  saved hangs : 0     │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 2.60 (66.7%)       │    map density : 0.00% / 0.00%      │
│  runs timed out : 0 (0.00%)          │ count coverage : 4.50 bits/tuple    │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : havoc                  │ favored items : 1 (33.33%)          │
│ stage execs : 1320/1766 (74.75%)     │  new edges on : 1 (33.33%)          │
│ total execs : 106k                   │ total crashes : 0 (0 saved)         │
│  exec speed : 452.1/sec              │  total tmouts : 0 (0 saved)         │
├─ fuzzing strategy yields ────────────┴─────────────┬─ item geometry ───────┤
│   bit flips : disabled (default, enable with -D)   │    levels : 1         │
│  byte flips : disabled (default, enable with -D)   │   pending : 0         │
│ arithmetics : disabled (default, enable with -D)   │  pend fav : 0         │
│  known ints : disabled (default, enable with -D)   │ own finds : 0         │
│  dictionary : n/a                                  │  imported : 0         │
│havoc/splice : 0/104k, 0/0                          │ stability : 50.00%    │
│py/custom/rq : unused, unused, unused, unused       ├───────────────────────┘
│    trim/eff : 20.00%/1, disabled                   │          [cpu000: 33%]
└────────────────────────────────────────────────────┘
```

- 0 crash and 0 hang were detected by AFL with sanitizer *signed-integer-overflow*


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/vulernable/integer
File Edit View Search Terminal Help

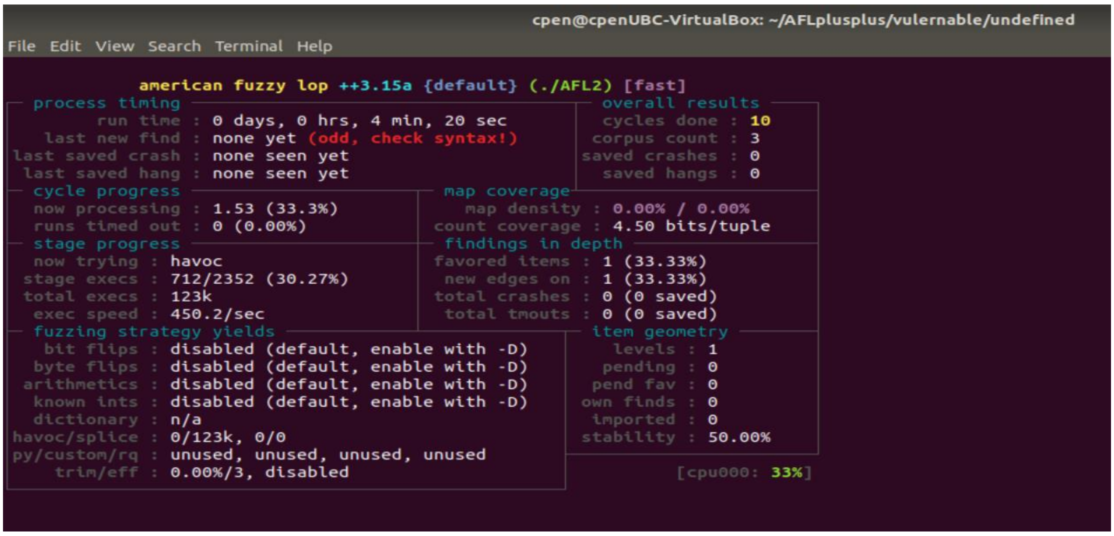
```
            american fuzzy lop ++3.15a {default} (./AFL2) [fast]
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 5 min, 18 sec      │  cycles done : 64    │
│   last new find : none yet (odd, check syntax!)     │ corpus count : 1     │
│last saved crash : none seen yet                     │saved crashes : 0     │
│ last saved hang : none seen yet                     │  saved hangs : 0     │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 0.194 (0.0%)       │    map density : 0.00% / 0.00%      │
│  runs timed out : 0 (0.00%)          │ count coverage : 4.00 bits/tuple    │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : havoc                  │ favored items : 1 (100.00%)         │
│ stage execs : 270/587 (46.00%)       │  new edges on : 1 (100.00%)         │
│ total execs : 113k                   │ total crashes : 0 (0 saved)         │
│  exec speed : 483.6/sec              │  total tmouts : 0 (0 saved)         │
├─ fuzzing strategy yields ────────────┴─────────────┬─ item geometry ───────┤
│   bit flips : disabled (default, enable with -D)   │    levels : 1         │
│  byte flips : disabled (default, enable with -D)   │   pending : 0         │
│ arithmetics : disabled (default, enable with -D)   │  pend fav : 0         │
│  known ints : disabled (default, enable with -D)   │ own finds : 0         │
│  dictionary : n/a                                  │  imported : 0         │
│havoc/splice : 0/113k, 0/0                          │ stability : 57.14%    │
│py/custom/rq : unused, unused, unused, unused       ├───────────────────────┘
│    trim/eff : 0.00%/1, disabled                    │          [cpu001: 50%]
└────────────────────────────────────────────────────┘
```

- 0 crash and 0 hang were detected by AFL with sanitizer *address*


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/vulernable/address
File Edit View Search Terminal Help

```
          american fuzzy lop ++3.15a {default} (./AFL2) [fast]
 process timing                                     overall results
        run time : 0 days, 0 hrs, 5 min, 58 sec     cycles done : 78
   last new find : none yet (odd, check syntax!)    corpus count : 1
last saved crash : none seen yet                   saved crashes : 0
 last saved hang : none seen yet                    saved hangs : 0
 cycle progress                          map coverage
  now processing : 0.235 (0.0%)            map density : 0.00% / 0.00%
  runs timed out : 0 (0.00%)            count coverage : 4.50 bits/tuple
 stage progress                          findings in depth
  now trying : havoc                    favored items : 1 (100.00%)
 stage execs : 480/587 (81.77%)          new edges on : 1 (100.00%)
 total execs : 137k                     total crashes : 0 (0 saved)
  exec speed : 347.3/sec                 total tmouts : 0 (0 saved)
 fuzzing strategy yields                               item geometry
   bit flips : disabled (default, enable with -D)        levels : 1
  byte flips : disabled (default, enable with -D)       pending : 0
 arithmetics : disabled (default, enable with -D)      pend fav : 0
  known ints : disabled (default, enable with -D)     own finds : 0
  dictionary : n/a                                     imported : 0
havoc/splice : 0/137k, 0/0                            stability : 50.00%
py/custom/rq : unused, unused, unused, unused
    trim/eff : 0.00%/1, disabled                          [cpu000: 50%]
```

- 0 crash and 0 hang were detected by AFL with sanitizer *undefined*

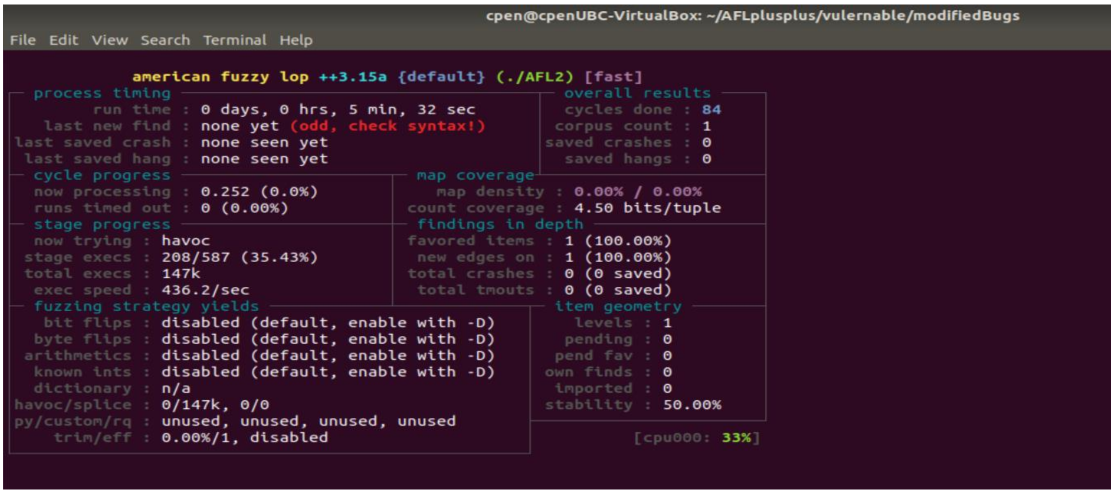

cpen@cpenUBC-VirtualBox: ~/AFLplusplus/vulernable/undefined
File Edit View Search Terminal Help

```
          american fuzzy lop ++3.15a {default} (./AFL2) [fast]
 process timing                                     overall results
        run time : 0 days, 0 hrs, 4 min, 20 sec     cycles done : 10
   last new find : none yet (odd, check syntax!)    corpus count : 3
last saved crash : none seen yet                   saved crashes : 0
 last saved hang : none seen yet                    saved hangs : 0
 cycle progress                          map coverage
  now processing : 1.53 (33.3%)            map density : 0.00% / 0.00%
  runs timed out : 0 (0.00%)            count coverage : 4.50 bits/tuple
 stage progress                          findings in depth
  now trying : havoc                    favored items : 1 (33.33%)
 stage execs : 712/2352 (30.27%)         new edges on : 1 (33.33%)
 total execs : 123k                     total crashes : 0 (0 saved)
  exec speed : 450.2/sec                 total tmouts : 0 (0 saved)
 fuzzing strategy yields                               item geometry
   bit flips : disabled (default, enable with -D)        levels : 1
  byte flips : disabled (default, enable with -D)       pending : 0
 arithmetics : disabled (default, enable with -D)      pend fav : 0
  known ints : disabled (default, enable with -D)     own finds : 0
  dictionary : n/a                                     imported : 0
havoc/splice : 0/123k, 0/0                            stability : 50.00%
py/custom/rq : unused, unused, unused, unused
    trim/eff : 0.00%/3, disabled                          [cpu000: 33%]
```

ii) Try fixing a few bugs(s) in the program, and running AFL on it again. Do you discover any new bugs or paths?

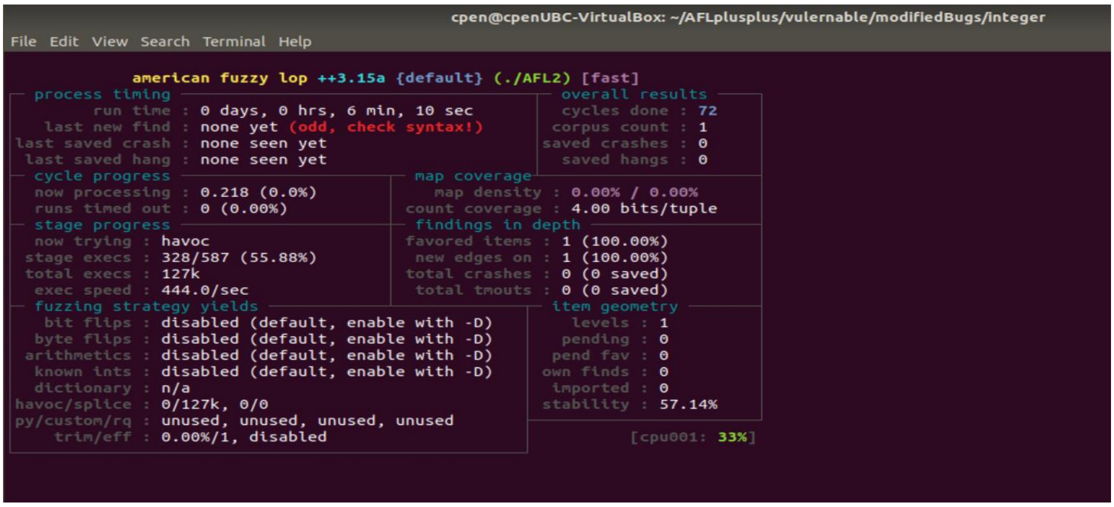
We use the bug-fixed program in Task 4 to run AFL again.

- 0 crash and 0 hang were detected by AFL without sanitizer


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/vulernable/modifiedBugs
File Edit View Search Terminal Help

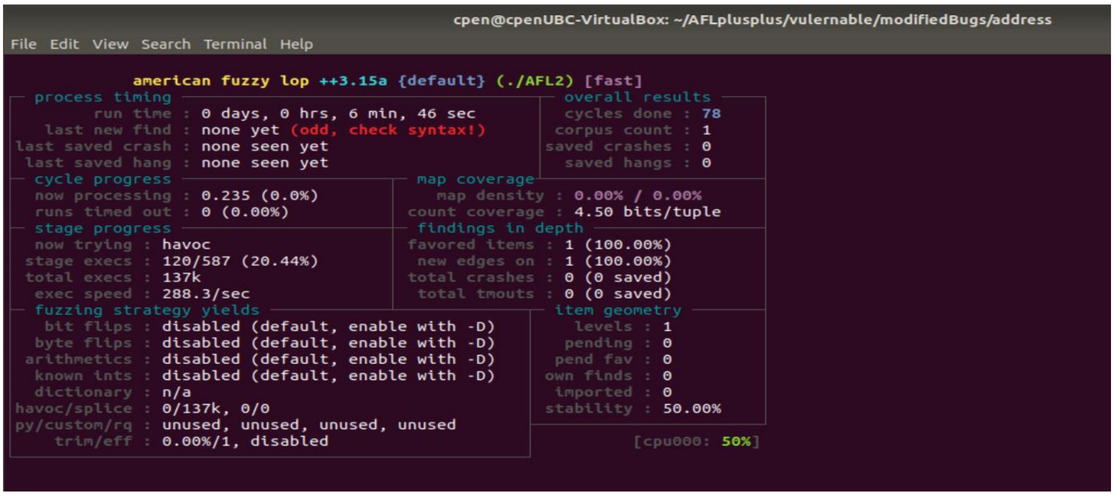
```
          american fuzzy lop ++3.15a {default} (./AFL2) [fast]
 process timing                                     overall results
        run time : 0 days, 0 hrs, 5 min, 32 sec     cycles done : 84
   last new find : none yet (odd, check syntax!)    corpus count : 1
last saved crash : none seen yet                   saved crashes : 0
 last saved hang : none seen yet                    saved hangs : 0
 cycle progress                          map coverage
  now processing : 0.252 (0.0%)            map density : 0.00% / 0.00%
  runs timed out : 0 (0.00%)            count coverage : 4.50 bits/tuple
 stage progress                          findings in depth
  now trying : havoc                    favored items : 1 (100.00%)
 stage execs : 208/587 (35.43%)          new edges on : 1 (100.00%)
 total execs : 147k                     total crashes : 0 (0 saved)
  exec speed : 436.2/sec                 total tmouts : 0 (0 saved)
 fuzzing strategy yields                               item geometry
   bit flips : disabled (default, enable with -D)        levels : 1
  byte flips : disabled (default, enable with -D)       pending : 0
 arithmetics : disabled (default, enable with -D)      pend fav : 0
  known ints : disabled (default, enable with -D)     own finds : 0
  dictionary : n/a                                     imported : 0
havoc/splice : 0/147k, 0/0                            stability : 50.00%
py/custom/rq : unused, unused, unused, unused
    trim/eff : 0.00%/1, disabled                          [cpu000: 33%]
```

- 0 crash and 0 hang were detected by AFL with sanitizer *signed-integer-overflow*


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/vulernable/modifiedBugs/integer
File Edit View Search Terminal Help

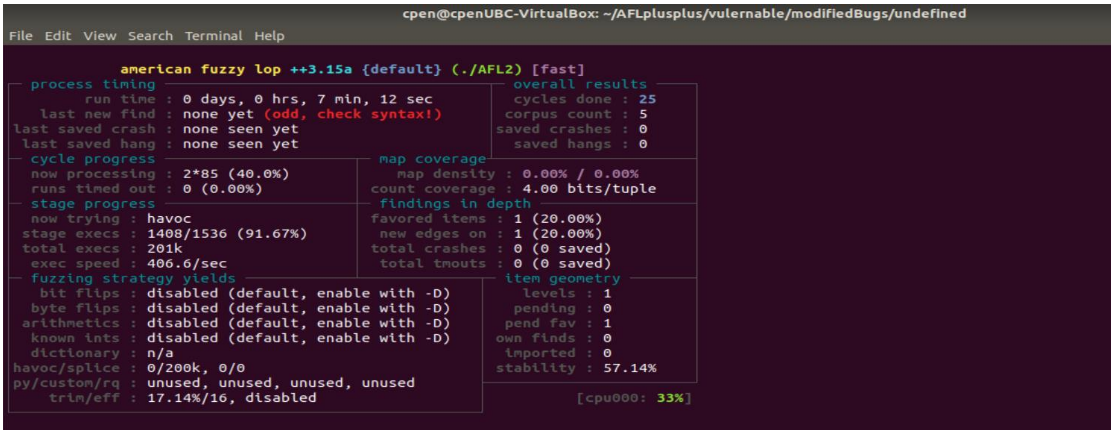
```
              american fuzzy lop ++3.15a {default} (./AFL2) [fast]
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 6 min, 10 sec      │  cycles done : 72    │
│   last new find : none yet (odd, check syntax!)     │ corpus count : 1     │
│last saved crash : none seen yet                     │saved crashes : 0     │
│ last saved hang : none seen yet                     │  saved hangs : 0     │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 0.218 (0.0%)       │    map density : 0.00% / 0.00%      │
│  runs timed out : 0 (0.00%)          │ count coverage : 4.00 bits/tuple    │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : havoc                  │ favored items : 1 (100.00%)         │
│ stage execs : 328/587 (55.88%)       │  new edges on : 1 (100.00%)         │
│ total execs : 127k                   │ total crashes : 0 (0 saved)         │
│  exec speed : 444.0/sec              │  total tmouts : 0 (0 saved)         │
├─ fuzzing strategy yields ────────────┴─────────────┬─ item geometry ───────┤
│   bit flips : disabled (default, enable with -D)   │    levels : 1         │
│  byte flips : disabled (default, enable with -D)   │   pending : 0         │
│ arithmetics : disabled (default, enable with -D)   │  pend fav : 0         │
│  known ints : disabled (default, enable with -D)   │ own finds : 0         │
│  dictionary : n/a                                  │  imported : 0         │
│havoc/splice : 0/127k, 0/0                          │ stability : 57.14%    │
│py/custom/rq : unused, unused, unused, unused       ├───────────────────────┘
│    trim/eff : 0.00%/1, disabled                    │          [cpu001: 33%]
└────────────────────────────────────────────────────┘
```

- 0 crash and 0 hang were detected by AFL with sanitizer *address*


cpen@cpenUBC-VirtualBox: ~/AFLplusplus/vulernable/modifiedBugs/address
File Edit View Search Terminal Help

```
              american fuzzy lop ++3.15a {default} (./AFL2) [fast]
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 6 min, 46 sec      │  cycles done : 78    │
│   last new find : none yet (odd, check syntax!)     │ corpus count : 1     │
│last saved crash : none seen yet                     │saved crashes : 0     │
│ last saved hang : none seen yet                     │  saved hangs : 0     │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 0.235 (0.0%)       │    map density : 0.00% / 0.00%      │
│  runs timed out : 0 (0.00%)          │ count coverage : 4.50 bits/tuple    │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : havoc                  │ favored items : 1 (100.00%)         │
│ stage execs : 120/587 (20.44%)       │  new edges on : 1 (100.00%)         │
│ total execs : 137k                   │ total crashes : 0 (0 saved)         │
│  exec speed : 288.3/sec              │  total tmouts : 0 (0 saved)         │
├─ fuzzing strategy yields ────────────┴─────────────┬─ item geometry ───────┤
│   bit flips : disabled (default, enable with -D)   │    levels : 1         │
│  byte flips : disabled (default, enable with -D)   │   pending : 0         │
│ arithmetics : disabled (default, enable with -D)   │  pend fav : 0         │
│  known ints : disabled (default, enable with -D)   │ own finds : 0         │
│  dictionary : n/a                                  │  imported : 0         │
│havoc/splice : 0/137k, 0/0                          │ stability : 50.00%    │
│py/custom/rq : unused, unused, unused, unused       ├───────────────────────┘
│    trim/eff : 0.00%/1, disabled                    │          [cpu000: 50%]
└────────────────────────────────────────────────────┘
```

- 0 crash and 0 hang were detected by AFL with sanitizer *undefined*

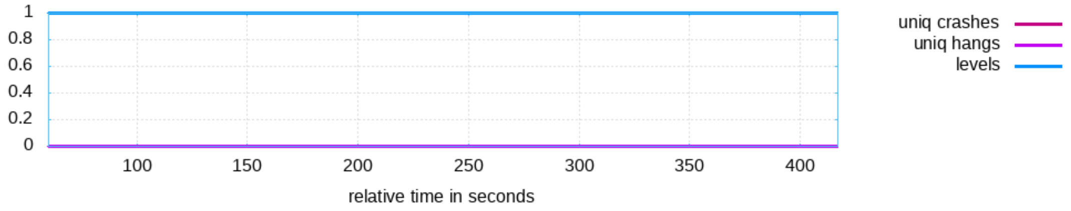

cpen@cpenUBC-VirtualBox: ~/AFLplusplus/vulernable/modifiedBugs/undefined
File Edit View Search Terminal Help

```
              american fuzzy lop ++3.15a {default} (./AFL2) [fast]
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 7 min, 12 sec      │  cycles done : 25    │
│   last new find : none yet (odd, check syntax!)     │ corpus count : 5     │
│last saved crash : none seen yet                     │saved crashes : 0     │
│ last saved hang : none seen yet                     │  saved hangs : 0     │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 2*85 (40.0%)       │    map density : 0.00% / 0.00%      │
│  runs timed out : 0 (0.00%)          │ count coverage : 4.00 bits/tuple    │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : havoc                  │ favored items : 1 (20.00%)          │
│ stage execs : 1408/1536 (91.67%)     │  new edges on : 1 (20.00%)          │
│ total execs : 201k                   │ total crashes : 0 (0 saved)         │
│  exec speed : 406.6/sec              │  total tmouts : 0 (0 saved)         │
├─ fuzzing strategy yields ────────────┴─────────────┬─ item geometry ───────┤
│   bit flips : disabled (default, enable with -D)   │    levels : 1         │
│  byte flips : disabled (default, enable with -D)   │   pending : 0         │
│ arithmetics : disabled (default, enable with -D)   │  pend fav : 1         │
│  known ints : disabled (default, enable with -D)   │ own finds : 0         │
│  dictionary : n/a                                  │  imported : 0         │
│havoc/splice : 0/200k, 0/0                          │ stability : 57.14%    │
│py/custom/rq : unused, unused, unused, unused       ├───────────────────────┘
│    trim/eff : 17.14%/16, disabled                  │          [cpu000: 33%]
└────────────────────────────────────────────────────┘
```

No new bugs were found.

iii) Plot a graph that shows the number of error cases (i.e., bugs, crashes, hangs) detected by AFL versus the elapsed time

- Because there is no bug detected by AFL, the graph is straight lines.



iv) Compare the results using KLEE and AFL on Vulnerable.c. Which tool is preferrable for bug detection? What are the advantages and drawbacks of using each tool?

- KLEE is preferrable for bug detection on *Vulnerable.c* because no bugs detected by AFL in our experiment.
- KLEE:
  - Pros: Because KLEE is a concolic execution tool, it can trigger the bug with precise input values, which help developers quickly identifying and fixing issues.
  - Cons: KLEE's focus on path exploration but it may not be able to handle system calls, and external dependencies effectively like AFL.
- AFL:
  - Pros: AFL can handle system calls, and external dependencies more effectively than KLEE.
  - Cons: AFL heavily relies on random mutations of inputs, which may lead to limited path coverage. Also, the large number of test inputs will lead to longer running time.