

SectionA 截图:

```
american fuzzy lop 1.96b (flaw)

process timing : 0 days, 0 hrs, 0 min, 43 sec
last new path  : none yet (odd, check syntax!)
last uniq crash: 0 days, 0 hrs, 0 min, 41 sec
last uniq hang  : 0 days, 0 hrs, 0 min, 39 sec
cycle progress  :
now processing  : 0 (0.00%)
paths timed out : 0 (0.00%)
stage progress  :
now trying      : havoc
stage execs     : 2016/5000 (40.32%)
total execs     : 37.5k
exec speed      : 917.7/sec
fuzzing strategy yields
bit flips      : 0/32, 0/31, 0/29
byte flips     : 0/4, 0/3, 0/1
arithmetics    : 0/224, 0/0, 0/0
known ints     : 0/27, 0/84, 0/44
dictionary     : 0/0, 0/0, 0/0
havoc          : 2/35.0k, 0/0
trim           : 20.00%/1, 0.00%

overall results
cycles done : 7
total paths : 1
uniq crashes : 2
uniq hangs  : 1

map coverage
map density : 4 (0.01%)
count coverage : 1.00 bits/tuple
findings in depth
favored paths : 1 (100.00%)
new edges on  : 1 (100.00%)
total crashes : 3 (2 unique)
total hangs   : 1 (1 unique)
path geometry
levels : 1
pending : 0
pend fav : 0
own finds : 0
imported : n/a
variable : 0

[cpu:132%]
```

SectionB:

实验的源代码在 test.c 中, test.c 中有变异输入、将输入写入文件、输出输入、将输入灌入 base64\_encode、输出输出等几个部分

变异的策略是产生两个随机数 a1 和 a2, 在 input 中随机的位置 (a1) 处开始插入随机个数 (a2-a1) 个字符;

编译 test.c 文件后就可以运行:

```
cx418y@ubuntu:~/Desktop/lab4/Data/sectionB$ gcc test.c -o test
cx418y@ubuntu:~/Desktop/lab4/Data/sectionB$ ./test
```

crash 截图:

```
INPUT IS: Welcome to mvfqo
OUTPUT IS :V2VsY29tZSB0byBtdmZxbw==
INPUT IS: WelcomEZfVZbQ`rU
CRASHED!
Segmentation fault
cx418y@ubuntu:~/Desktop/lab4/Data/sectionB$
```