# 《软件安全》实验报告

姓名：曹瑜　　学号：2212794　　班级：密码科学与技术

**实验名称：**

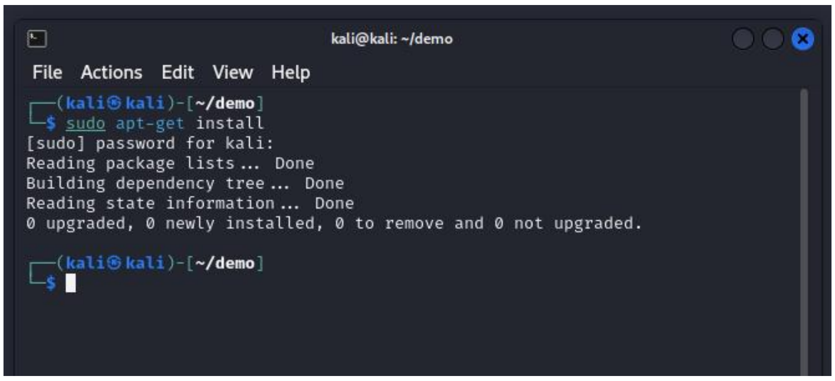
AFL 模糊测试实验

**实验要求：**

根据课本 7.4.5 章节，复现 AFL 在 KALI 下的安装、应用查阅资料理解覆盖引导和文件变异的概念和含义。

**实验过程：**

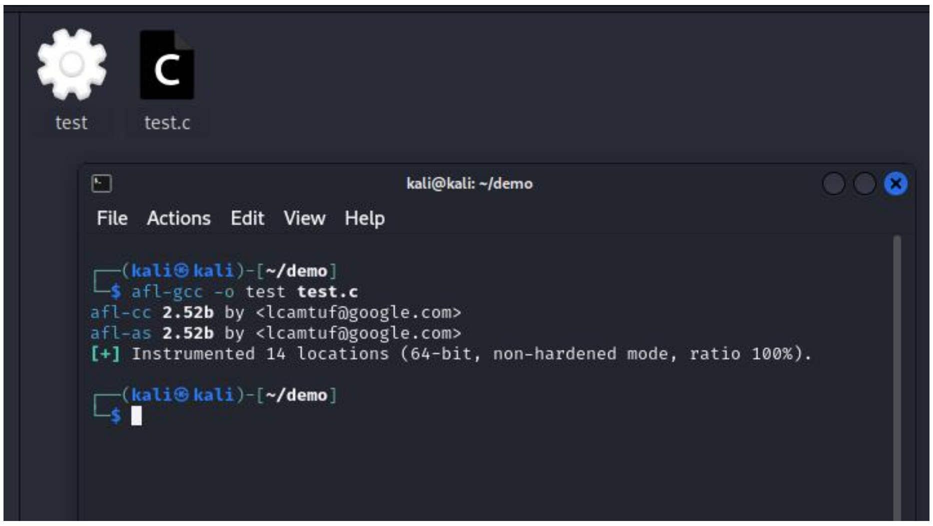
进入 kali 虚拟机完成 AFL 安装：



```
┌──(kali㉿kali)-[~/demo]
└─$ sudo apt-get install
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

┌──(kali㉿kali)-[~/demo]
└─$ 
```

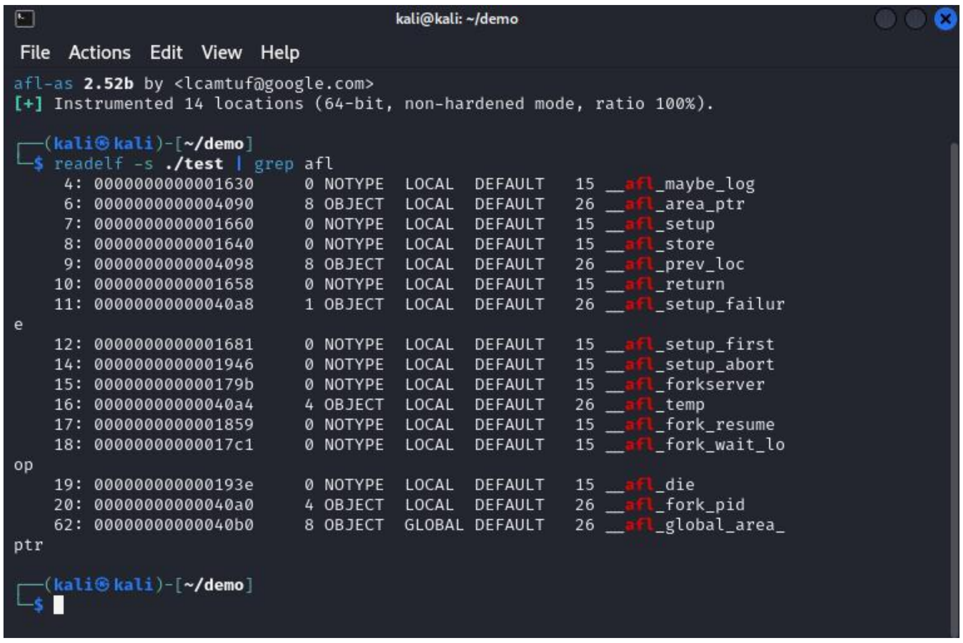
使用 afl 编译器进行编译，输入命令：afl-gcc -o test test.c

编译后完成插桩



```
┌──(kali㉿kali)-[~/demo]
└─$ afl-gcc -o test test.c
afl-cc 2.52b by <lcamtuf@google.com>
afl-as 2.52b by <lcamtuf@google.com>
[+] Instrumented 14 locations (64-bit, non-hardened mode, ratio 100%).

┌──(kali㉿kali)-[~/demo]
└─$ 
```

运行命令 readelf -s ./test | grep afl 后，可见 test 文件中插桩

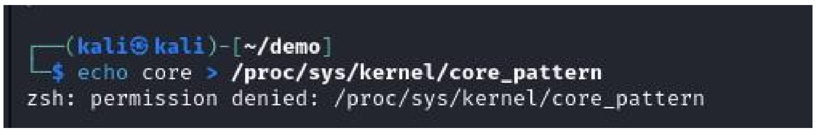


```
afl-as 2.52b by <lcamtuf@google.com>
[+] Instrumented 14 locations (64-bit, non-hardened mode, ratio 100%).

┌──(kali㉿kali)-[~/demo]
└─$ readelf -s ./test | grep afl
     4: 0000000000001630     0 NOTYPE  LOCAL  DEFAULT   15 __afl_maybe_log
     6: 0000000000004090     8 OBJECT  LOCAL  DEFAULT   26 __afl_area_ptr
     7: 0000000000001660     0 NOTYPE  LOCAL  DEFAULT   15 __afl_setup
     8: 0000000000001640     0 NOTYPE  LOCAL  DEFAULT   15 __afl_store
     9: 0000000000004098     8 OBJECT  LOCAL  DEFAULT   26 __afl_prev_loc
    10: 0000000000001658     0 NOTYPE  LOCAL  DEFAULT   15 __afl_return
    11: 00000000000040a8     1 OBJECT  LOCAL  DEFAULT   26 __afl_setup_failur
e
    12: 0000000000001681     0 NOTYPE  LOCAL  DEFAULT   15 __afl_setup_first
    14: 0000000000001946     0 NOTYPE  LOCAL  DEFAULT   15 __afl_setup_abort
    15: 000000000000179b     0 NOTYPE  LOCAL  DEFAULT   15 __afl_forkserver
    16: 00000000000040a4     4 OBJECT  LOCAL  DEFAULT   26 __afl_temp
    17: 0000000000001859     0 NOTYPE  LOCAL  DEFAULT   15 __afl_fork_resume
    18: 00000000000017c1     0 NOTYPE  LOCAL  DEFAULT   15 __afl_fork_wait_lo
op
    19: 000000000000193e     0 NOTYPE  LOCAL  DEFAULT   15 __afl_die
    20: 00000000000040a0     4 OBJECT  LOCAL  DEFAULT   26 __afl_fork_pid
    62: 00000000000040b0     8 OBJECT  GLOBAL DEFAULT   26 __afl_global_area_
ptr

┌──(kali㉿kali)-[~/demo]
└─$ 
```

进行准备工作：输入输出文件的创建

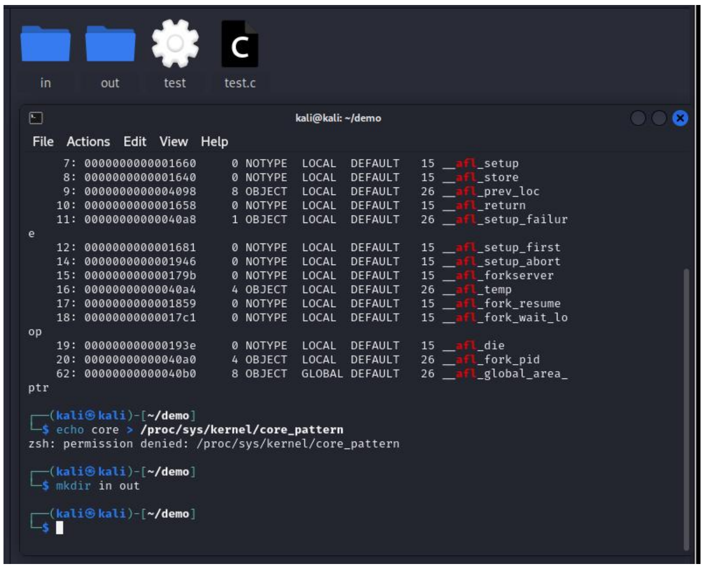
运行命令：echo core > /proc/sys/kernel/core_pattern

将 coredumps 输出为文件



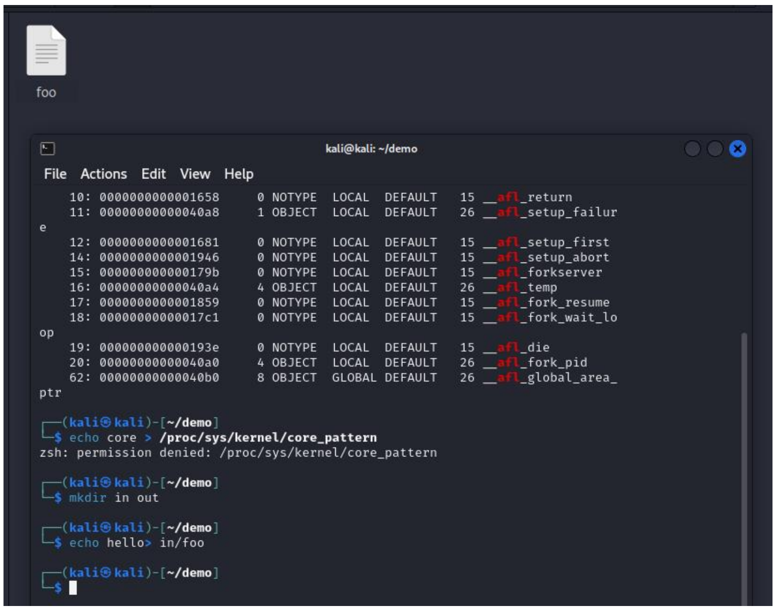
```
┌──(kali㉿kali)-[~/demo]
└─$ echo core > /proc/sys/kernel/core_pattern
zsh: permission denied: /proc/sys/kernel/core_pattern
```

运行命令：mkdir in out ，创建 in 和 out 文件夹



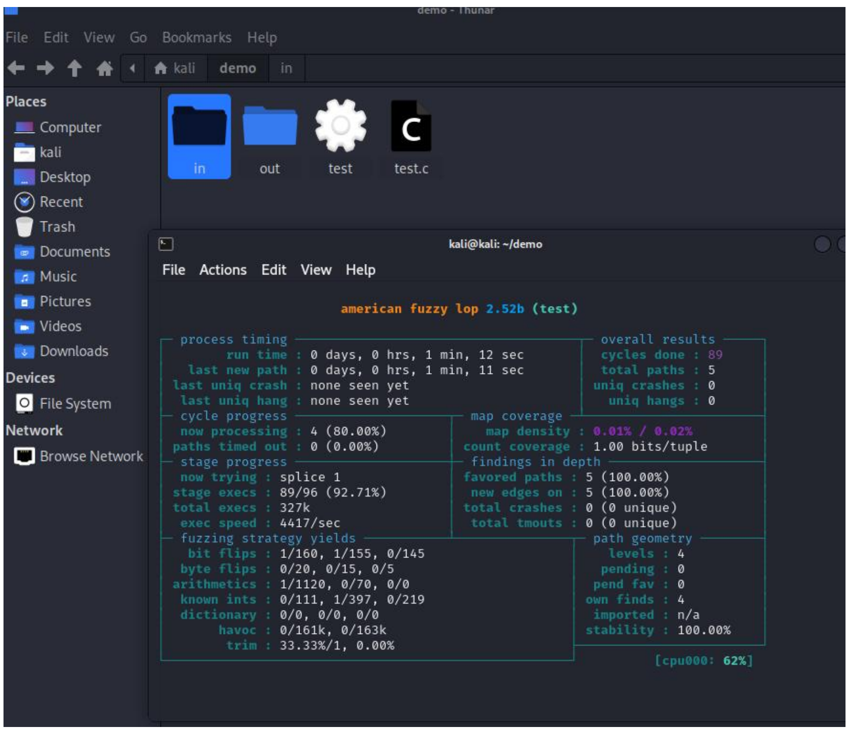
```
     7: 0000000000001660     0 NOTYPE  LOCAL  DEFAULT   15 __afl_setup
     8: 0000000000001640     0 NOTYPE  LOCAL  DEFAULT   15 __afl_store
     9: 0000000000004098     8 OBJECT  LOCAL  DEFAULT   26 __afl_prev_loc
    10: 0000000000001658     0 NOTYPE  LOCAL  DEFAULT   15 __afl_return
    11: 00000000000040a8     1 OBJECT  LOCAL  DEFAULT   26 __afl_setup_failur
e
    12: 0000000000001681     0 NOTYPE  LOCAL  DEFAULT   15 __afl_setup_first
    14: 0000000000001946     0 NOTYPE  LOCAL  DEFAULT   15 __afl_setup_abort
    15: 000000000000179b     0 NOTYPE  LOCAL  DEFAULT   15 __afl_forkserver
    16: 00000000000040a4     4 OBJECT  LOCAL  DEFAULT   26 __afl_temp
    17: 0000000000001859     0 NOTYPE  LOCAL  DEFAULT   15 __afl_fork_resume
    18: 00000000000017c1     0 NOTYPE  LOCAL  DEFAULT   15 __afl_fork_wait_lo
op
    19: 000000000000193e     0 NOTYPE  LOCAL  DEFAULT   15 __afl_die
    20: 00000000000040a0     4 OBJECT  LOCAL  DEFAULT   26 __afl_fork_pid
    62: 00000000000040b0     8 OBJECT  GLOBAL DEFAULT   26 __afl_global_area_
ptr

┌──(kali㉿kali)-[~/demo]
└─$ echo core > /proc/sys/kernel/core_pattern
zsh: permission denied: /proc/sys/kernel/core_pattern

┌──(kali㉿kali)-[~/demo]
└─$ mkdir in out

┌──(kali㉿kali)-[~/demo]
└─$ 
```

运行命令 echo hello> in/foo ，向 in 文件夹中创建一个包含字符串“hello”的文件 foo



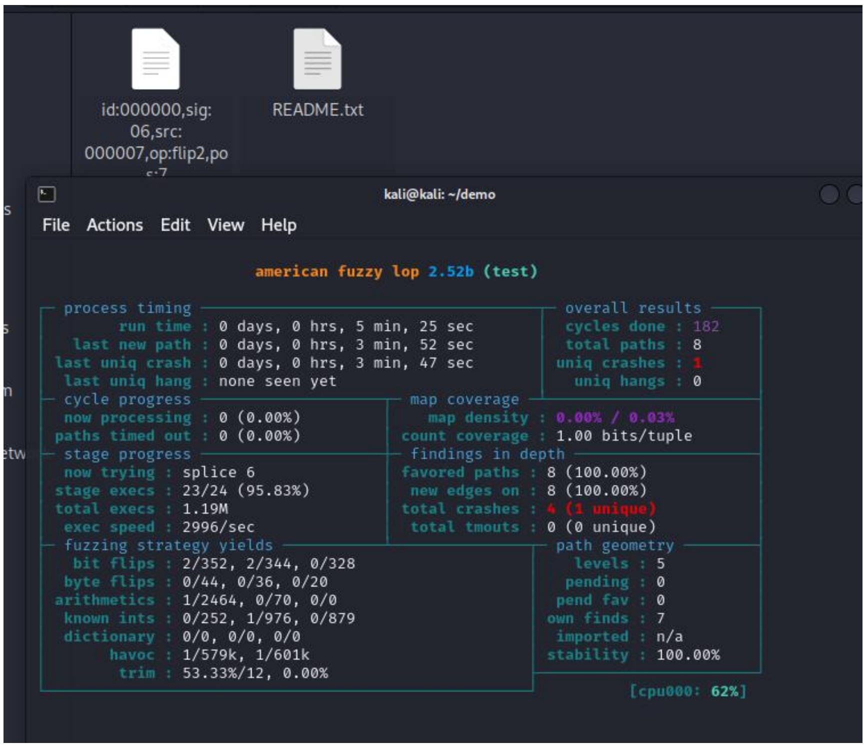
```
    10: 0000000000001658     0 NOTYPE  LOCAL  DEFAULT   15 __afl_return
    11: 00000000000040a8     1 OBJECT  LOCAL  DEFAULT   26 __afl_setup_failur
e
    12: 0000000000001681     0 NOTYPE  LOCAL  DEFAULT   15 __afl_setup_first
    14: 0000000000001946     0 NOTYPE  LOCAL  DEFAULT   15 __afl_setup_abort
    15: 000000000000179b     0 NOTYPE  LOCAL  DEFAULT   15 __afl_forkserver
    16: 00000000000040a4     4 OBJECT  LOCAL  DEFAULT   26 __afl_temp
    17: 0000000000001859     0 NOTYPE  LOCAL  DEFAULT   15 __afl_fork_resume
    18: 00000000000017c1     0 NOTYPE  LOCAL  DEFAULT   15 __afl_fork_wait_lo
op
    19: 000000000000193e     0 NOTYPE  LOCAL  DEFAULT   15 __afl_die
    20: 00000000000040a0     4 OBJECT  LOCAL  DEFAULT   26 __afl_fork_pid
    62: 00000000000040b0     8 OBJECT  GLOBAL DEFAULT   26 __afl_global_area_
ptr

┌──(kali㉿kali)-[~/demo]
└─$ echo core > /proc/sys/kernel/core_pattern
zsh: permission denied: /proc/sys/kernel/core_pattern

┌──(kali㉿kali)-[~/demo]
└─$ mkdir in out

┌──(kali㉿kali)-[~/demo]
└─$ echo hello> in/foo

┌──(kali㉿kali)-[~/demo]
└─$ █
```

运行 afl-fuzz -i in -o out -- ./test @@ ，启动模糊测试



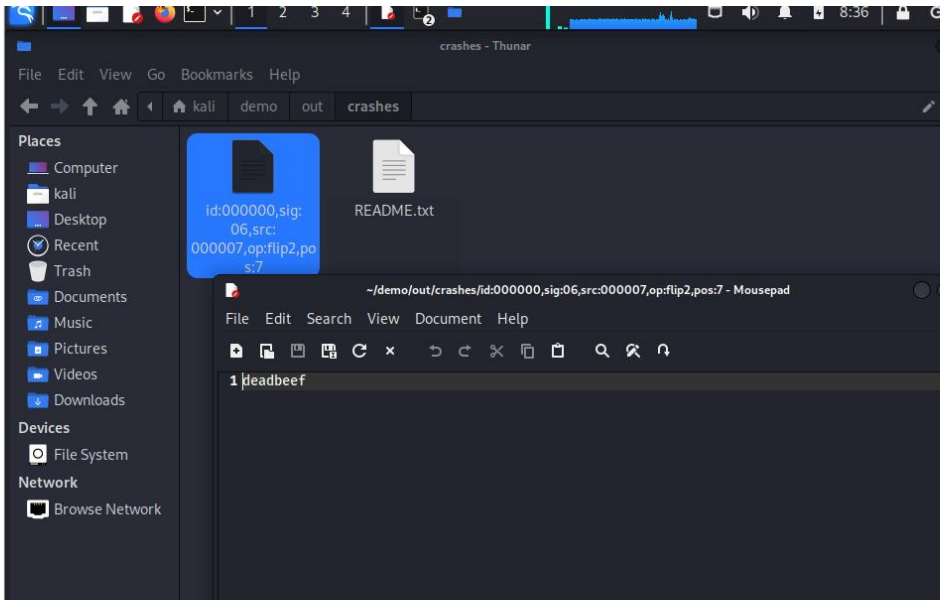
```
                     american fuzzy lop 2.52b (test)

┌─ process timing ─────────────────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 0 hrs, 1 min, 12 sec       │  cycles done : 89     │
│   last new path : 0 days, 0 hrs, 1 min, 11 sec       │  total paths : 5      │
│ last uniq crash : none seen yet                      │ uniq crashes : 0      │
│  last uniq hang : none seen yet                      │   uniq hangs : 0      │
├─ cycle progress ────────────────────┬─ map coverage ─┴───────────────────────┤
│  now processing : 4 (80.00%)        │    map density : 0.01% / 0.02%        │
│ paths timed out : 0 (0.00%)         │ count coverage : 1.00 bits/tuple      │
├─ stage progress ────────────────────┼─ findings in depth ────────────────────┤
│  now trying : splice 1              │ favored paths : 5 (100.00%)           │
│ stage execs : 89/96 (92.71%)        │  new edges on : 5 (100.00%)           │
│ total execs : 327k                  │ total crashes : 0 (0 unique)          │
│  exec speed : 4417/sec              │  total tmouts : 0 (0 unique)          │
├─ fuzzing strategy yields ───────────┴───────────────┬─ path geometry ────────┤
│   bit flips : 1/160, 1/155, 0/145                   │    levels : 4          │
│  byte flips : 0/20, 0/15, 0/5                       │   pending : 0          │
│ arithmetics : 1/1120, 0/70, 0/0                     │  pend fav : 0          │
│  known ints : 0/111, 1/397, 0/219                   │ own finds : 4          │
│  dictionary : 0/0, 0/0, 0/0                         │  imported : n/a        │
│       havoc : 0/161k, 0/163k                        │ stability : 100.00%    │
│        trim : 33.33%/1, 0.00%                       ├────────────────────────┘
└─────────────────────────────────────────────────────┘          [cpu000: 62%]
```

模糊测试执行中，若找到触发异常的 crash，会存储在 out 文件夹的 crash 子文件夹中


id:000000,sig:06,src:000007,op:flip2,pos:7
README.txt
kali@kali: ~/demo
File Actions Edit View Help

```
              american fuzzy lop 2.52b (test)

┌─ process timing ─────────────────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 0 hrs, 5 min, 25 sec       │  cycles done : 182    │
│   last new path : 0 days, 0 hrs, 3 min, 52 sec       │  total paths : 8      │
│ last uniq crash : 0 days, 0 hrs, 3 min, 47 sec       │ uniq crashes : 1      │
│  last uniq hang : none seen yet                      │   uniq hangs : 0      │
├─ cycle progress ────────────────────┬─ map coverage ─┴───────────────────────┤
│  now processing : 0 (0.00%)         │    map density : 0.00% / 0.03%        │
│ paths timed out : 0 (0.00%)         │ count coverage : 1.00 bits/tuple      │
├─ stage progress ────────────────────┼─ findings in depth ────────────────────┤
│  now trying : splice 6              │ favored paths : 8 (100.00%)           │
│ stage execs : 23/24 (95.83%)        │  new edges on : 8 (100.00%)           │
│ total execs : 1.19M                 │ total crashes : 4 (1 unique)          │
│  exec speed : 2996/sec              │  total tmouts : 0 (0 unique)          │
├─ fuzzing strategy yields ───────────┴───────────┬─ path geometry ────────────┤
│   bit flips : 2/352, 2/344, 0/328               │    levels : 5             │
│  byte flips : 0/44, 0/36, 0/20                  │   pending : 0             │
│ arithmetics : 1/2464, 0/70, 0/0                 │  pend fav : 0             │
│  known ints : 0/252, 1/976, 0/879               │ own finds : 7             │
│  dictionary : 0/0, 0/0, 0/0                     │  imported : n/a           │
│       havoc : 1/579k, 1/601k                    │ stability : 100.00%       │
│        trim : 53.33%/12, 0.00%                  ├───────────────────────────┘
└─────────────────────────────────────────────────┘          [cpu000: 62%]
```

最后得到的 crash 样例结果为“ deadbeef ”


crashes - Thunar
File Edit View Go Bookmarks Help
kali demo out crashes
Places
Computer
kali
Desktop
Recent
Trash
Documents
Music
Pictures
Videos
Downloads
Devices
File System
Network
Browse Network
id:000000,sig:06,src:000007,op:flip2,pos:7
README.txt
~/demo/out/crashes/id:000000,sig:06,src:000007,op:flip2,pos:7 - Mousepad
File Edit Search View Document Help

```
deadbeef
```

**心得体会：**

通过本次实验，成功在 kali 虚拟机上安装了 AFL，并复现了 AFL 模糊测试的代码，完成了 AFL 模糊测试并得到了正确结果，理解了 AFL 模糊测试的原理，了解了如何指定 fuzzing 的输入输出，如何查看最终分析结果，对 AFL 模糊测试有了更深入的了解