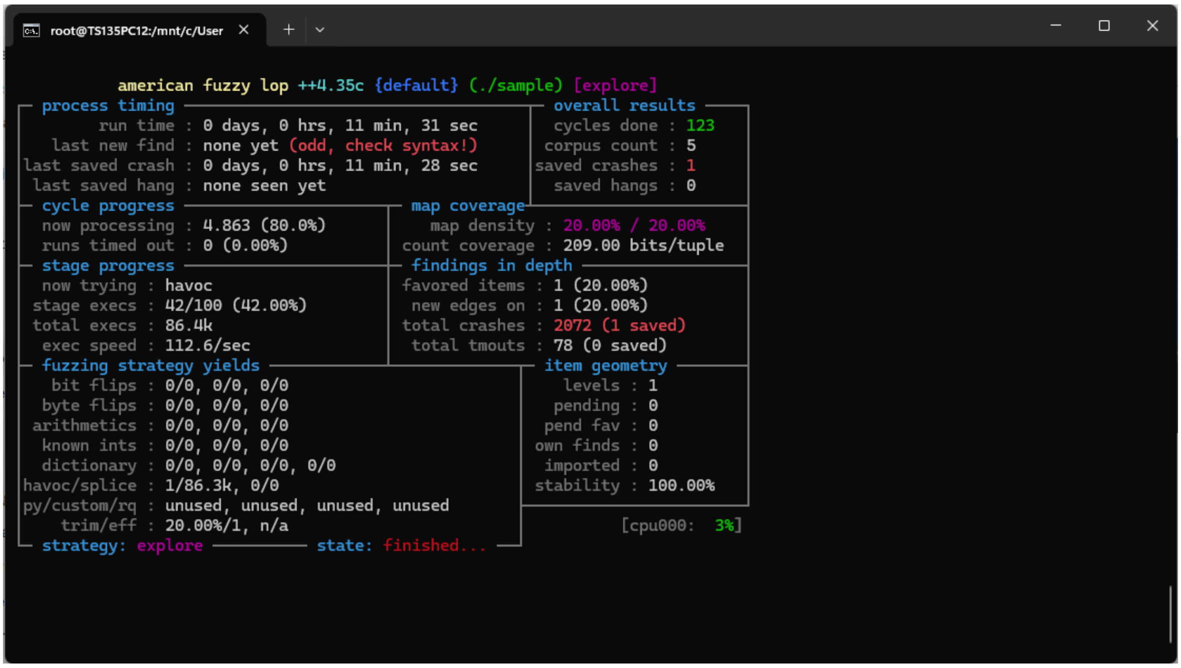
Task 2

Input :

```
cd /mnt/c/Users/pvinit25/Desktop/CyberSecurityTesting101-main/6.Fuzzing/
AFL_USE_ASAN=1 afl-cc -o sample sample.c
```

Output :



```
          american fuzzy lop ++4.35c {default} (./sample) [explore]
┌─ process timing ────────────────────────────────────┬─ overall results ────┐
│        run time : 0 days, 0 hrs, 11 min, 31 sec     │  cycles done : 123   │
│   last new find : none yet (odd, check syntax!)     │ corpus count : 5     │
│last saved crash : 0 days, 0 hrs, 11 min, 28 sec     │saved crashes : 1     │
│ last saved hang : none seen yet                     │  saved hangs : 0     │
├─ cycle progress ─────────────────────┬─ map coverage┴──────────────────────┤
│  now processing : 4.863 (80.0%)      │    map density : 20.00% / 20.00%    │
│  runs timed out : 0 (0.00%)          │ count coverage : 209.00 bits/tuple  │
├─ stage progress ─────────────────────┼─ findings in depth ─────────────────┤
│  now trying : havoc                  │ favored items : 1 (20.00%)          │
│ stage execs : 42/100 (42.00%)        │  new edges on : 1 (20.00%)          │
│ total execs : 86.4k                  │ total crashes : 2072 (1 saved)      │
│  exec speed : 112.6/sec              │  total tmouts : 78 (0 saved)        │
├─ fuzzing strategy yields ────────────┴─────────────┬─ item geometry ───────┤
│   bit flips : 0/0, 0/0, 0/0                        │    levels : 1         │
│  byte flips : 0/0, 0/0, 0/0                        │   pending : 0         │
│ 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, 0/0                   │  imported : 0         │
│havoc/splice : 1/86.3k, 0/0                         │ stability : 100.00%   │
│py/custom/rq : unused, unused, unused, unused       ├───────────────────────┘
│    trim/eff : 20.00%/1, n/a                        │          [cpu000:  3%]
└─ strategy: explore ────────── state: finished... ──┘
```

Input :

```
./sample output/default/crashes/id:000000*
```

Output :

```
=================================================================
==89433==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x72c85c3de052
at pc 0x5b8bdc7398d4 bp 0x7ffd90e45810 sp 0x7ffd90e44fd0
WRITE of size 58 at 0x72c85c3de052 thread T0
    #0 0x5b8bdc7398d3 in scanf_common(void*, int, bool, char const*, __va_list_tag*)
asan_interceptors.cpp.o
    #1 0x5b8bdc781bb8 in __isoc99_vfscanf
(/mnt/c/Users/pvinit25/Desktop/CyberSecurityTesting101-main/6.Fuzzing/sample+0xfebb8
) (BuildId: 4bb0c0afc99f3c15510e5a9fb936bfd83371f0da)
    #2 0x5b8bdc7821cd in __isoc99_fscanf
(/mnt/c/Users/pvinit25/Desktop/CyberSecurityTesting101-main/6.Fuzzing/sample+0xff1cd)
(BuildId: 4bb0c0afc99f3c15510e5a9fb936bfd83371f0da)
```

```
    #3 0x5b8bdc805136 in main
/mnt/c/Users/pvinit25/Desktop/CyberSecurityTesting101-main/6.Fuzzing/sample.c:16:9
    #4 0x76c85e0eb634  (/usr/lib/libc.so.6+0x27634) (BuildId:
5e2075850f8de86da4eead11213c59d926ca3796)
    #5 0x76c85e0eb6e8 in __libc_start_main (/usr/lib/libc.so.6+0x276e8) (BuildId:
5e2075850f8de86da4eead11213c59d926ca3796)
    #6 0x5b8bdc6b0194 in _start
(/mnt/c/Users/pvinit25/Desktop/CyberSecurityTesting101-main/6.Fuzzing/sample+0x2d194
) (BuildId: 4bb0c0afc99f3c15510e5a9fb936bfd83371f0da)

Address 0x72c85c3de052 is located in stack of thread T0 at offset 82 in frame
    #0 0x5b8bdc804fcf in main
/mnt/c/Users/pvinit25/Desktop/CyberSecurityTesting101-main/6.Fuzzing/sample.c:5

  This frame has 1 object(s):
    [32, 82) 'buffer' (line 6) <== Memory access at offset 82 overflows this variable
HINT: this may be a false positive if your program uses some custom stack unwind
mechanism, swapcontext or vfork
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow asan_interceptors.cpp.o in
scanf_common(void*, int, bool, char const*, __va_list_tag*)
Shadow bytes around the buggy address:
  0x72c85c3ddd80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x72c85c3dde00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x72c85c3dde80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x72c85c3ddf00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x72c85c3ddf80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x72c85c3de000: f1 f1 f1 f1 00 00 00 00 00 00[02]f3 f3 f3 f3 f3
  0x72c85c3de080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x72c85c3de100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x72c85c3de180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x72c85c3de200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x72c85c3de280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==89433==ABORTING
```

Yes, the AddressSanitizer pinpoints the exact line causing the crash: line 16 in main(), where fscanf writes into buffer without size checking. It also reports the vulnerability type (stack-buffer-overflow), the write size (58 bytes), and the affected variable (buffer, declared at line 6 with size 50).

The fuzzer found 2072 total crashes, but only 1 unique. All crashes triggered the same underlying vulnerability, so AFL++ only saved one.

During that process, 123 cycles have been completed. A cycle could be defined as one full pass through all corpus inputs with different mutation strategies applied.

The fuzzer was stopped when the state displayed finished and no new unique crashes had been found for many cycles, which means that all reachable paths in the program had been explored.

At screenshot time, AFL++ was running the havoc mutation stage under the global explore strategy.

Task 3

In this task, we had to retrieve the language file that never made it into production from Juice Shop.

I used the command ffuf to enumerate possible language files under assets/i18n/ by replacing FUZZ with entries from a wordlist. The -fs 75055 flag helped me to filter out responses of that specific size, which corresponds to the default page returned for non-existing files, it allows us to keep only the relevant results.

Input :

```
ffuf -u http://localhost:3000/assets/i18n/FUZZ.json \
    -w /usr/share/seclists/Discovery/Web-Content/common.txt \
    -fs 75055
```

Output :

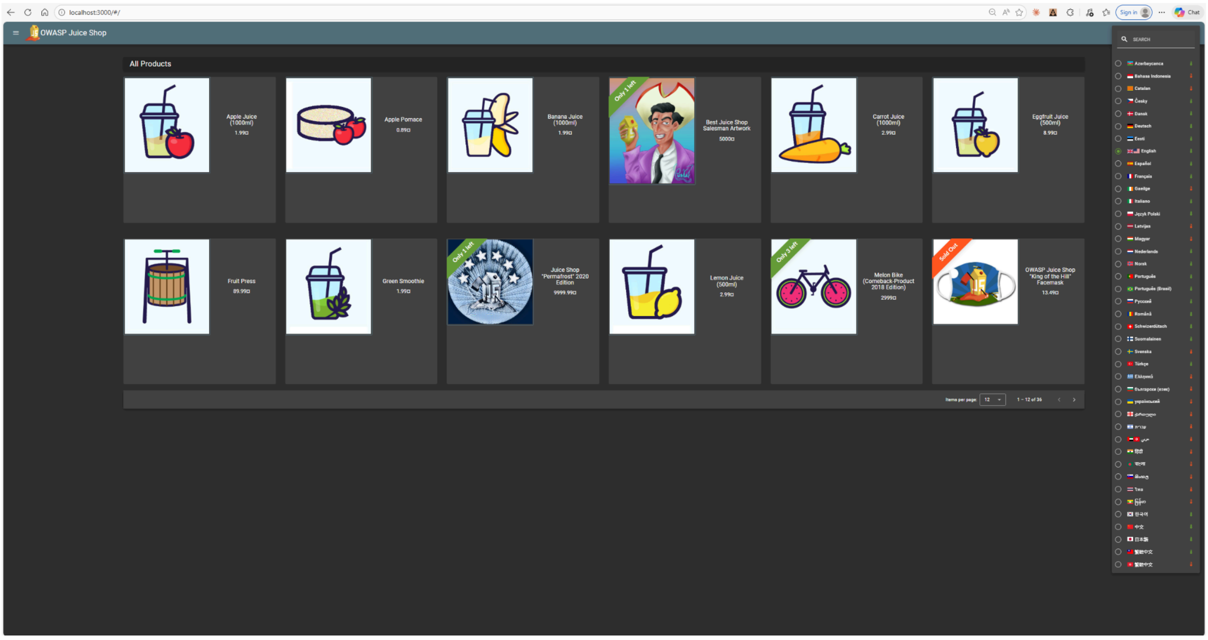
```
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________
```

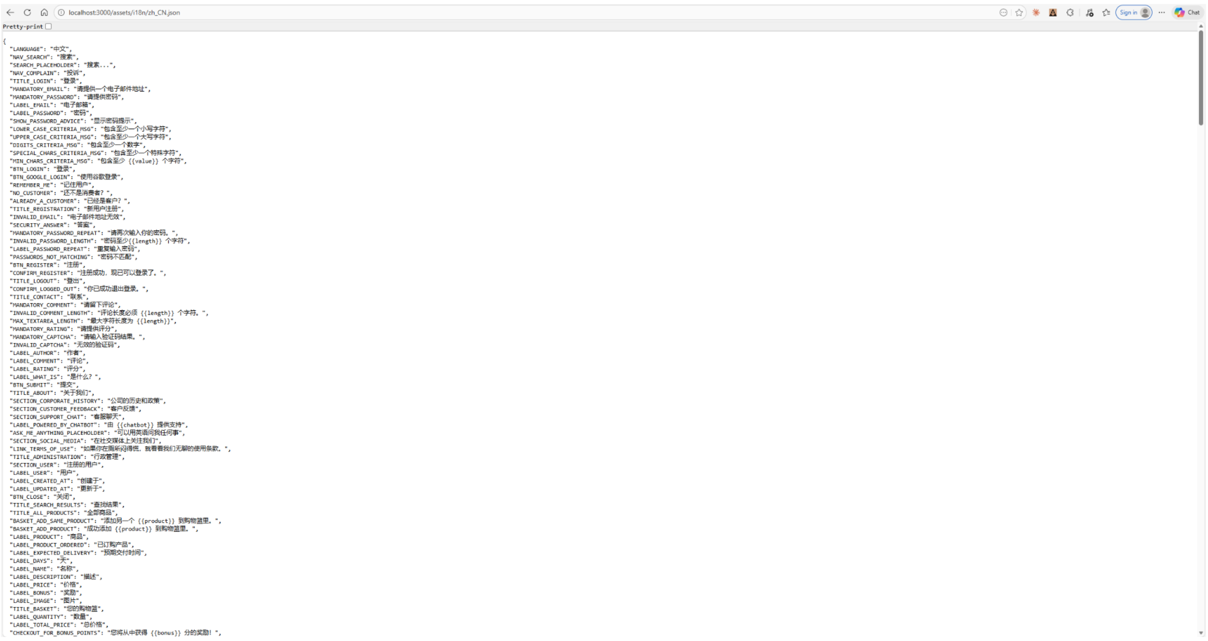
```
:: Method           : GET
:: URL              : http://localhost:3000/assets/i18n/FUZZ.json
:: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/common.txt
:: Follow redirects : false
:: Calibration      : false
:: Timeout          : 10
:: Threads          : 40
:: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
:: Filter           : Response size: 75055
________________________________________________

de_DE                   [Status: 200, Size: 36840, Words: 3703, Lines: 484, Duration: 57ms]
en                      [Status: 200, Size: 33094, Words: 3871, Lines: 484, Duration: 79ms]
es_ES                   [Status: 200, Size: 36082, Words: 4243, Lines: 484, Duration: 85ms]
fr_FR                   [Status: 200, Size: 37109, Words: 4324, Lines: 484, Duration: 49ms]
it_IT                   [Status: 200, Size: 34656, Words: 3981, Lines: 484, Duration: 91ms]
ja_JP                   [Status: 200, Size: 41505, Words: 1696, Lines: 484, Duration: 49ms]
ko_KR                   [Status: 200, Size: 37201, Words: 3296, Lines: 484, Duration: 73ms]
pt_BR                   [Status: 200, Size: 35187, Words: 4043, Lines: 484, Duration: 101ms]
zh_CN                   [Status: 200, Size: 31056, Words: 1681, Lines: 484, Duration: 59ms]
zh_TW                   [Status: 200, Size: 31334, Words: 1921, Lines: 484, Duration: 45ms]
:: Progress: [4751/4751] :: Job [1/1] :: 358 req/sec :: Duration: [0:00:15] :: Errors: 0 ::
```

The scan we did previously returned 10 valid language files, including zh_CN.json which is Simplified Chinese and this language is not available in the Juice Shop language selector.



However, it remains accessible on the server by using the following link localhost:3000/assets/i18n/zh_CN.json



```
{
  "LANGUAGE": "中文",
  "NAV_SEARCH": "搜索",
  "SEARCH_PLACEHOLDER": "搜索...",
  "NAV_COMPLAIN": "投诉",
  "TITLE_LOGIN": "登录",
  "MANDATORY_EMAIL": "请提供一个电子邮件地址",
  "MANDATORY_PASSWORD": "请提供密码",
  "LABEL_EMAIL": "电子邮箱",
  "LABEL_PASSWORD": "密码",
  "SHOW_PASSWORD_ADVICE": "显示密码提示",
  "LOWER_CASE_CRITERIA_MSG": "包含至少一个小写字符",
  "UPPER_CASE_CRITERIA_MSG": "包含至少一个大写字符",
  "DIGITS_CRITERIA_MSG": "包含至少一个数字",
  "SPECIAL_CHARS_CRITERIA_MSG": "包含至少一个特殊字符",
  "MIN_CHARS_CRITERIA_MSG": "包含至少 {{value}} 个字符",
  "BTN_LOGIN": "登录",
  "BTN_GOOGLE_LOGIN": "使用谷歌登录",
  "REMEMBER_ME": "记住用户",
  "NO_CUSTOMER": "还不是消费者？",
  "ALREADY_A_CUSTOMER": "已经是客户？",
  "TITLE_REGISTRATION": "新用户注册",
  "INVALID_EMAIL": "电子邮件地址无效",
  "SECURITY_ANSWER": "答案",
  "MANDATORY_PASSWORD_REPEAT": "请再次输入你的密码。",
  "INVALID_PASSWORD_LENGTH": "密码至少{{length}} 个字符",
  "LABEL_PASSWORD_REPEAT": "重复输入密码",
  "PASSWORDS_NOT_MATCHING": "密码不匹配",
  "BTN_REGISTER": "注册",
  "CONFIRM_REGISTER": "注册成功，现已可以登录了。",
  "TITLE_LOGOUT": "登出",
  "CONFIRM_LOGGED_OUT": "你已成功退出登录。",
  "TITLE_CONTACT": "联系",
  "MANDATORY_COMMENT": "请留下评论",
  "INVALID_COMMENT_LENGTH": "评论长度必须 {{length}} 个字符。",
  "MAX_TEXTAREA_LENGTH": "最大字符长度为 {{length}}",
  "MANDATORY_RATING": "请提供评分",
  "MANDATORY_CAPTCHA": "请输入验证码结果。",
  "INVALID_CAPTCHA": "无效的验证码",
  "LABEL_AUTHOR": "作者",
  "LABEL_COMMENT": "评论",
  "LABEL_RATING": "评分",
  "LABEL_WHAT_IS": "是什么？",
  "BTN_SUBMIT": "提交",
  "TITLE_ABOUT": "关于我们",
  "SECTION_CORPORATE_HISTORY": "公司的历史和政策",
  "SECTION_CUSTOMER_FEEDBACK": "客户反馈",
  "SECTION_SUPPORT_CHAT": "客服聊天",
  "LABEL_POWERED_BY_CHATBOT": "由 {{chatbot}} 提供支持",
  "ASK_ME_ANYTHING_PLACEHOLDER": "可以用英语问我任何事",
  "SECTION_SOCIAL_MEDIA": "在社交媒体上关注我们",
  "LINK_TERMS_OF_USE": "如果你在厕所闷得慌，就看看我们无聊的使用条款。",
  "TITLE_ADMINISTRATION": "行政管理",
  "SECTION_USER": "注册的用户",
  "LABEL_USER": "用户",
  "LABEL_CREATED_AT": "创建于",
  "LABEL_UPDATED_AT": "更新于",
  "BTN_CLOSE": "关闭",
  "TITLE_SEARCH_RESULTS": "查找结果",
  "TITLE_ALL_PRODUCTS": "全部商品",
  "BASKET_ADD_SAME_PRODUCT": "添加另一个 {{product}} 到购物篮里。",
  "BASKET_ADD_PRODUCT": "成功添加 {{product}} 到购物篮里。",
  "LABEL_PRODUCT": "商品",
  "LABEL_PRODUCT_ORDERED": "已订购产品",
  "LABEL_EXPECTED_DELIVERY": "预期交付时间",
  "LABEL_DAYS": "天",
  "LABEL_NAME": "名称",
  "LABEL_DESCRIPTION": "描述",
  "LABEL_PRICE": "价格",
  "LABEL_BONUS": "奖励",
  "LABEL_IMAGE": "图片",
  "TITLE_BASKET": "您的购物篮",
  "LABEL_QUANTITY": "数量",
  "LABEL_TOTAL_PRICE": "总价格",
  "CHECKOUT_FOR_BONUS_POINTS": "您将从中获得 {{bonus}} 分的奖励！",
```

Then, we retrieved the file with:

Input :

```
curl -O http://localhost:3000/assets/i18n/zh_CN.json
```

Output :

```
  % Total    % Received % Xferd  Average Speed  Time    Time     Time   Current
                                 Dload  Upload  Total   Spent    Left   Speed
100  31056 100  31056   0      0  6.00M      0                          0
```