一、AFL——产生输入

1. 基础内容

AFL官方文档

MULL官方文档

AFL基础知识

1. AFL相关

1.1. 输入

对于直接从 stdin 接受输入的目标二进制文件，通常的语法是：

```
$ ./afl-fuzz -i testcase_dir -o findings_dir /path/to/program [...params...]
```

对于从文件获取输入的程序，使用“@@”标记目标命令行中应放置输入文件名的位置。模糊器将为您替换它：

```
$ ./afl-fuzz -i testcase_dir -o findings_dir /path/to/program @@
```

1.2. 输出

AFL输出文件：

- crashes：

  导致目标接收致命signal而崩溃的独特测试用例

- queue：

  存放所有具有独特执行路径的测试用例。

- crashes/README.txt：

  保存了目标执行这些crashes文件的命令行参数。

- hangs：

  导致目标超时的独特测试用例。

- fuzzer_stats：

  afl-fuzz的运行状态。

- plot_data：

  用于afl-plot绘图。

2. 实现过程

AFL测试binutils/readelf

2.1. binutils

源码包编译指令：

```
cd /opt/Mull/Subjects/binutils-2.25
CC=afl-gcc ./configure
make
```

① binutils cxxfilt测试流程

```
cd /opt/Mull/Subjects/binutils-2.25
mkdir afl_in_cxxfilt afl_out_cxxfilt
echo main > afl_in_cxxfilt/input01.txt
echo _Z1fv > afl_in_cxxfilt/input02.txt
echo _ZNSt22condition_variable_anyD2Ev > afl_in_cxxfilt/input03.txt
echo N3foo12BarExceptionE > afl_in_cxxfilt/input04.txt
echo _ZQQ > afl_in_cxxfilt/input05.txt
afl-fuzz -i afl_in_cxxfilt -o afl_out_cxxfilt ./binutils/cxxfilt
```

② binutils nm测试流程

```
cd /opt/Mull/Subjects/binutils-2.25
mkdir afl_in_nm afl_out_nm
echo auto > afl_in_nm/input01.txt
echo gnu > afl_in_nm/input02.txt
echo lucid > afl_in_nm/input03.txt
echo arm > afl_in_nm/input04.txt
echo hp > afl_in_nm/input05.txt
echo edg > afl_in_nm/input06.txt
echo gnu-v3 > afl_in_nm/input07.txt
echo java > afl_in_nm/input08.txt
echo gnat > afl_in_nm/input09.txt
echo compaq > afl_in_nm/input10.txt
afl-fuzz -i afl_in_nm -o afl_out_nm ./binutils/nm @@
```

③ binutils size 测试流程

```
cd /opt/Mull/Subjects/binutils-2.25
mkdir afl_in_size afl_out_size
echo noargs > afl_in_size/input01.txt
afl-fuzz -i afl_in_size -o afl_out_size ./binutils/size @@
```

④ binutils strip 测试流程

```
cd /opt/Mull/Subjects/binutils-2.25
mkdir afl_in_strip afl_out_strip afl_strip_tmpout
echo noargs > afl_in_strip/input01.txt
echo "hello world" > afl_in_strip/input02.txt
echo "  a,b,c,d,e" > afl_in_strip/input03.txt
echo "  iii " > afl_in_strip/input04.txt
echo "to be or not to be   " > afl_in_strip/input05.txt
afl-fuzz -i afl_in_strip -o afl_out_strip ./binutils/strip-new @@ -o
./afl_strip_tmpout
```

⑤ binutils readelf 测试流程

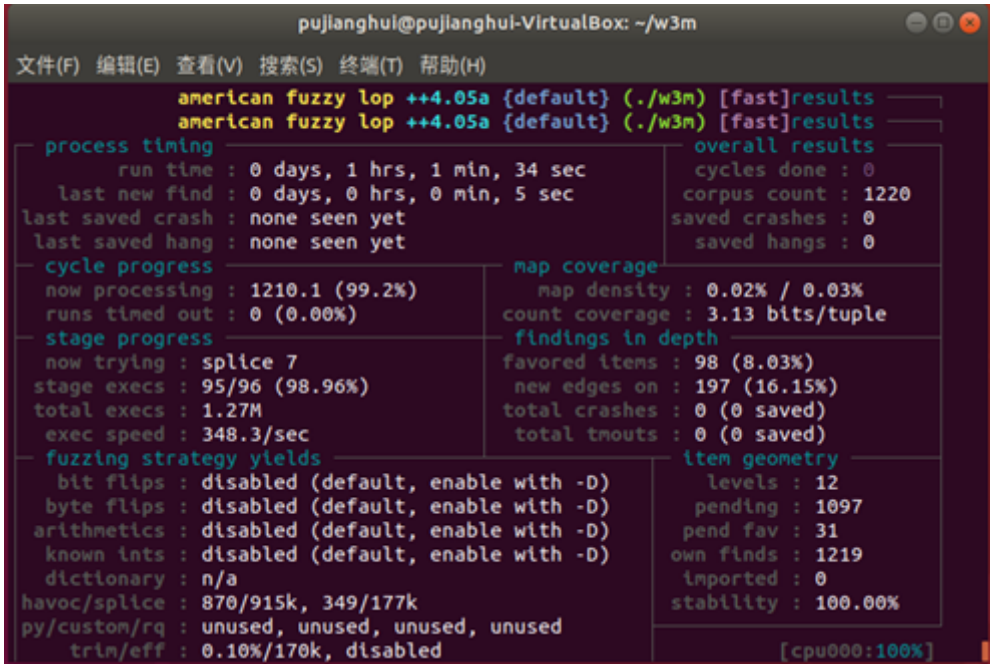
```
cd /opt/Mull/Subjects/binutils-2.25
mkdir afl_in afl_out
sudo bash -c 'echo core >/proc/sys/kernel/core_pattern'
cp /bin/ps afl_in/
afl-fuzz -i afl_in_size -o afl_out_size ./binutils/readelf -a @@
```

⑥ binutils objdump 测试流程

```
cd /opt/Mull/Subjects/binutils-2.25
mkdir afl_in afl_out
sudo bash -c 'echo core >/proc/sys/kernel/core_pattern'
cp /bin/ps afl_in/
afl-fuzz -i afl_in_size -o afl_out_size ./binutils/objdump -SD @@
```

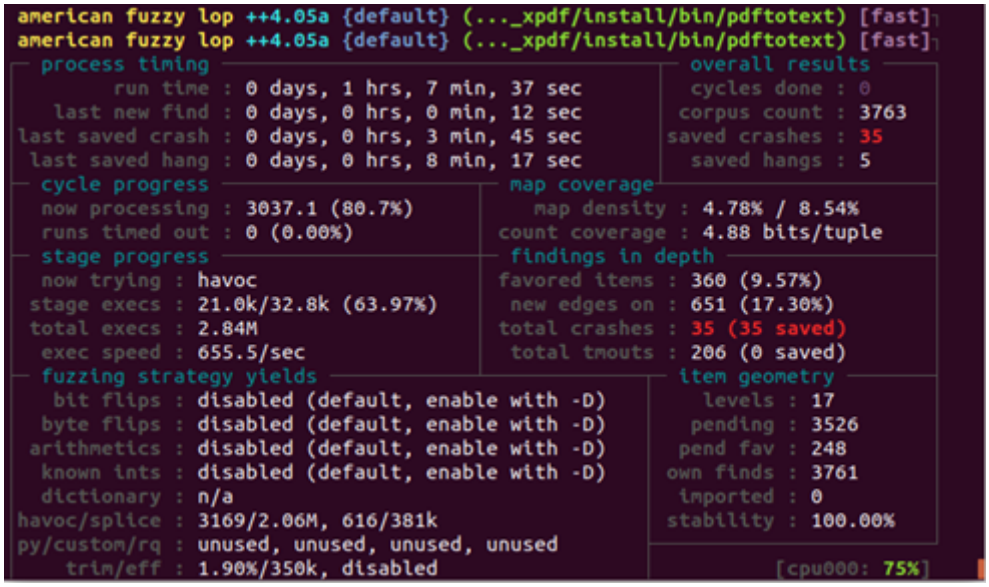
2.2. W3m

W3m AFL结果 可以看到代码维护的非常好 无crash。（通过对in的index.html进行fuzz）



2.3. XPDF

XPDF的AFL测试结果 可以看到还是有不少crash的（仅测试1h）



二、Mull——复现

1. 基础内容

编写 `mull.yml` 并在 `/etc/profile` 设置环境变量 `MULL_CONFIG` 来指定配置文件的路径

```
mutators: # 使用的变异算子，参考官方文档"Supported Mutation Operators"页面
  - cxx_add_to_sub
  - cxx_logical
excludePaths: # 被指定的路径下所有文件不会产生变异体，支持正则表达式，也可以直接写需要排除的代码文件的路径+文件名。参考官方文档Tutorials/Keeping #mutants under control/File Path Filters
  - gtest
  - gmock
timeout: # 设置每个变异体的超时时间，默认单位为毫秒
  10000 # 10 seconds
quiet: false # 静默模式开关，若设为true，则编译时不会输出编译日志
```

2. 实现过程

2.1. libxml2

利用Python语言，借助pandas、numpy数据分析包以及sqlite3库，从sqlite结果文件中，提取mutant数据库中的内容到csv文件，代码以及提取数据中前五条部分信息如下所示：

In [1]:
```
import sqlite3 as sl
import pandas as pd
import numpy as np
```

In [2]:
```
PATH_PREFIX = "../data/"
SQLITE_FILE_PATH = PATH_PREFIX + "report.sqlite"
OUTPUT_CSV_PATH = PATH_PREFIX + "report.csv"

# 建立连接
connection = sl.connect(SQLITE_FILE_PATH)
```

In [3]:
```
# 创建游标cursor来执行SQL语句
cursor=connection.cursor()

# 查询表名
cursor.execute("SELECT * FROM mutant;")
tables=cursor.fetchall()

# 获取列名
col_name_list = [tuple[0] for tuple in cursor.description]

# 保存到文件
df = pd.DataFrame(tables, columns=col_name_list)
df.to_csv(OUTPUT_CSV_PATH)
df.head()
```

Out[3]:

	mutant_id	mutator	filename	direc
0	cxx_sub_to_add:/opt/Mull/Subjects/libxml2/./ti...	cxx_sub_to_add	/opt/Mull/Subjects/libxml2/./timsort.h	
1	cxx_sub_to_add:/opt/Mull/Subjects/libxml2/./ti...	cxx_sub_to_add	/opt/Mull/Subjects/libxml2/./timsort.h	
2	cxx_sub_to_add:/opt/Mull/Subjects/libxml2/./ti...	cxx_sub_to_add	/opt/Mull/Subjects/libxml2/./timsort.h	
3	cxx_add_to_sub:/opt/Mull/Subjects/libxml2/./ti...	cxx_add_to_sub	/opt/Mull/Subjects/libxml2/./timsort.h	
4	cxx_sub_to_add:/opt/Mull/Subjects/libxml2/./ti...	cxx_sub_to_add	/opt/Mull/Subjects/libxml2/./timsort.h	

2.2. W3m

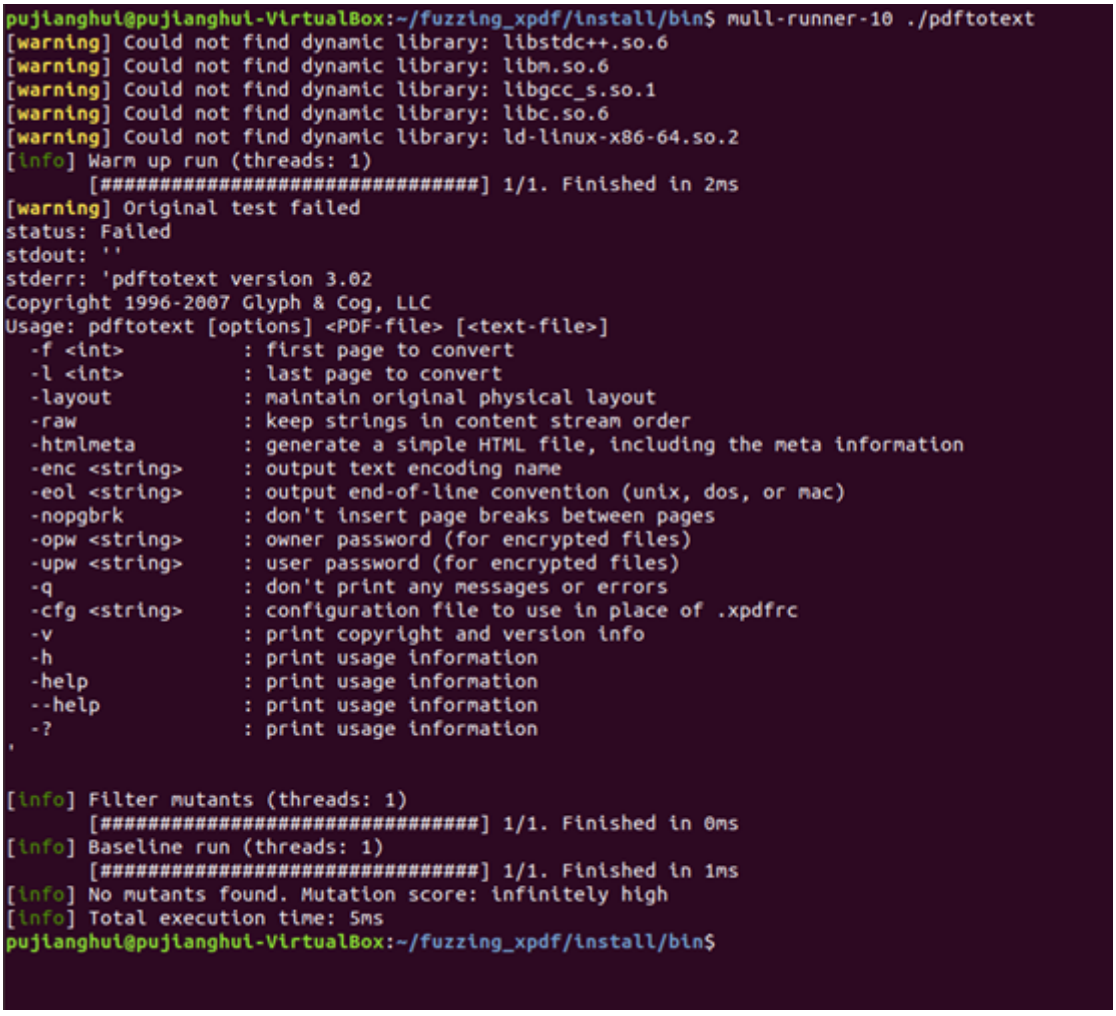
得分非常高

```
pujianghui@pujianghui-VirtualBox:~/w3m$ mull-runner-10 ./w3m
[warning] Could not find dynamic library: libm.so.6
[warning] Could not find dynamic library: libgc.so.1
[warning] Could not find dynamic library: libssl.so.1.1
[warning] Could not find dynamic library: libcrypto.so.1.1
[warning] Could not find dynamic library: libtinfo.so.5
[warning] Could not find dynamic library: libc.so.6
[info] Warm up run (threads: 1)
       [################################] 1/1. Finished in 6009ms
[warning] Original test failed
status: Timedout
stdout: ''
stderr: ''

[info] Filter mutants (threads: 1)
       [################################] 1/1. Finished in 0ms
[info] Baseline run (threads: 1)
       [################################] 1/1. Finished in 6012ms
[info] No mutants found. Mutation score: infinitely high
[info] Total execution time: 12023ms
pujianghui@pujianghui-VirtualBox:~/w3m$
```

2.3. XPDF

XPDF的pdftotext文件的mull执行结果。



三、绘制统计图

借助Python语言，利用numpy、pandas、matplotlib等数据分析与可视化库，绘制利用AFL进行模糊测试时#path随时间（单位：分钟）变化的折线图，代码如下：

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

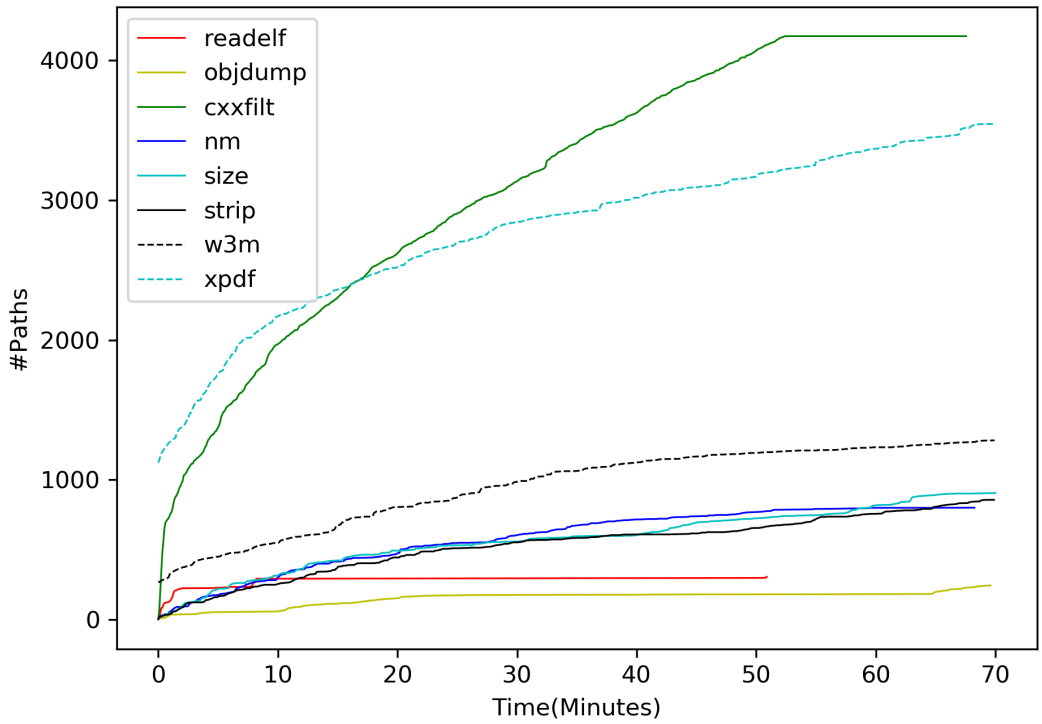
In [2]:
```
DURATION = 70
PATH_PREFIX = "../data/"
FILE_PATH_READELF = PATH_PREFIX + "plot_data_readelf.csv"
FILE_PATH_OBJDUMP = PATH_PREFIX + "plot_data_objdump.csv"
FILE_PATH_CXXFILT = PATH_PREFIX + "plot_data_cxxfilt.csv"
FILE_PATH_NM = PATH_PREFIX + "plot_data_nm.csv"
FILE_PATH_SIZE = PATH_PREFIX + "plot_data_size.csv"
FILE_PATH_STRIP = PATH_PREFIX + "plot_data_strip.csv"
FILE_PATH_W3M = PATH_PREFIX + "plot_data_w3m.csv"
FILE_PATH_XPDF = PATH_PREFIX + "plot_data_xpdf.csv"
```

In [3]:
```
df_readelf = pd.read_csv(FILE_PATH_READELF)
df_objdump = pd.read_csv(FILE_PATH_OBJDUMP)
df_cxxfilt = pd.read_csv(FILE_PATH_CXXFILT)
df_nm = pd.read_csv(FILE_PATH_NM)
df_size = pd.read_csv(FILE_PATH_SIZE)
df_strip = pd.read_csv(FILE_PATH_STRIP)
df_w3m = pd.read_csv(FILE_PATH_W3M)
df_xpdf = pd.read_csv(FILE_PATH_XPDF)
lst_df = [
    ['readelf', df_readelf, 'r', '-' ],
    ['objdump', df_objdump, 'y', '-' ],
    ['cxxfilt', df_cxxfilt, 'g', '-' ],
    ['nm'     , df_nm     , 'b', '-' ],
    ['size'   , df_size   , 'c', '-' ],
    ['strip'  , df_strip  , 'k', '-' ],
    ['w3m'    , df_w3m    , 'k', '--'],
    ['xpdf'   , df_xpdf   , 'c', '--'],
]
```

In [4]:
```
# 传入记录AFL生成数据的DataFrame，获取横坐标（time:0~60min）和纵坐标（total paths）
def getData(df):
    x = (df.iloc[:, 0] - df.iloc[0, 0]) / 60
    x = x[x <= DURATION]
    length = x.shape[0]
    y = list(df.iloc[:, 3])[0: length]
    return x, y
```

In [5]:
```
fig = plt.figure(figsize=(7, 5), dpi=300)
plt.xlabel('Time(Minutes)')
plt.ylabel('#Paths')
for each in lst_df:
    x, y = getData(each[1])
    plt.plot(x, y, lw=0.75, ls=each[3], c=each[2], label=each[0])
plt.legend(loc=0)
plt.show()
fig.savefig("../result/result")
```

最终生成图表如下图所示：



四、过程中遇到的问题

1. binutils版本问题

执行命令`git clone git://sourceware.org/git/binutils-gdb.git`得到binutils，使用mull-run-10无法执行binutils-gdb内下的可执行文件，发现是版本问题，后clone版本为2.25的binutils即可解决。

2. XPDF的output/queue文件无法进行Mull测试

XPDF的output/queue文件 发现无法继续

```
pujianghui@pujianghui-VirtualBox:~/fuzzing_xpdf/out/default/queue$ mull-runner-1
0 ./id:000000,time:0,execs:0,orig:small-example-pdf-file.pdf
[error] Cannot create SymbolicFile from: /home/pujianghui/fuzzing_xpdf/out/defau
lt/queue/id:000000,time:0,execs:0,orig:small-example-pdf-file.pdf
[error] Error messages are treated as fatal errors. Exiting now.
pujianghui@pujianghui-VirtualBox:~/fuzzing_xpdf/out/default/queue$
```

3. afl-fuzz下载的pdf文件出现问题

```
[-] PROGRAM ABORT : Pipe at the beginning of 'core_pattern'
         Location : check_crash_handling(), src/afl-fuzz-init.c:2197
```

后执行命令`echo core >/proc/sys/kernel/core_pattern`解决

4. 用mull对w3m的output进行测试时的问题

```
pujianghui@pujianghui-VirtualBox:~/w3m/out/default/queue$ mull-runner-10 ./id:00
0000,time:0,execs:0,orig:index.html
[error] Cannot create SymbolicFile from: /home/pujianghui/w3m/out/default/queue/
id:000000,time:0,execs:0,orig:index.html
[error] Error messages are treated as fatal errors. Exiting now.
pujianghui@pujianghui-VirtualBox:~/w3m/out/default/queue$
```

未找到解决办法。