Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

Not all bytes are equal: Neural byte sieve for fuzzing

硕士研究生　袁晓筱

2019年1月20日

内容提要

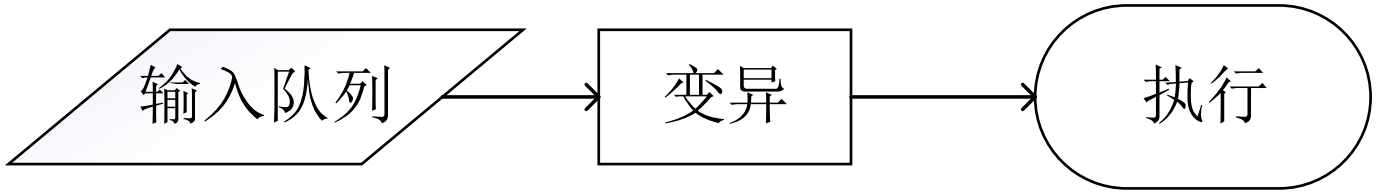
- 基本概念
  - AFL模糊测试器
- 算法原理
  - 流程和框架
- 实验结果
- 参考文献

预期收获

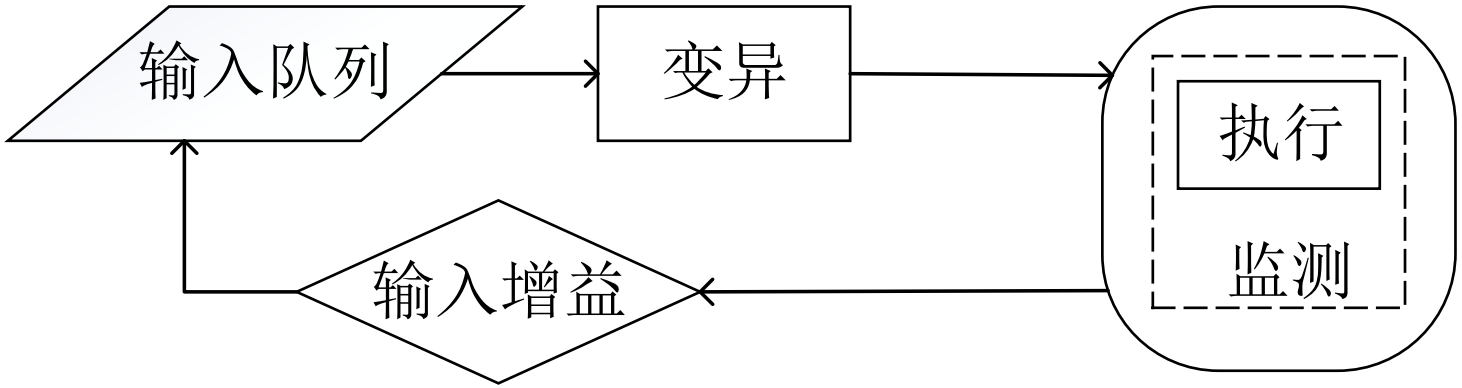
- 预期收获
  - 1. 了解AFL模糊测试器的工作流程
  - 2. 了解神经网络在模糊测试中的应用方式

基本概念

- AFL模糊测试器
  - AFL模糊测试器是一种先进的模糊测试器



简单的模糊测试器



AFL模糊测试器

基本概念

- 伪代码
  - AFL额外计算了输入的路径覆盖(cov)

```
for 输入∈输入队列 do:
|   for 执行次数←0 to 上限 do:
|   |   变异
|   |   执行结果←执行（被测程序，变异输入）
|   |   if 执行结果 is 崩溃 then
|   |       Append 变异输入 to 恶意输入文件夹
|   |   end
|   |
|   |
|   |
|   end
end
```

```
for 输入∈输入队列 do:
|   for 执行次数←0 to 上限 do:
|   |   变异
|   |   执行结果，路径覆盖←执行（被测程序，变异输入）
|   |   if 执行结果 is 崩溃 then
|   |   |   Append 变异输入 to 恶意输入文件夹
|   |   end
|   |   if 有新的路径 then
|   |   |   Append 变异输入 to 输入队列
|   |   end
|   end
end
```

基本概念

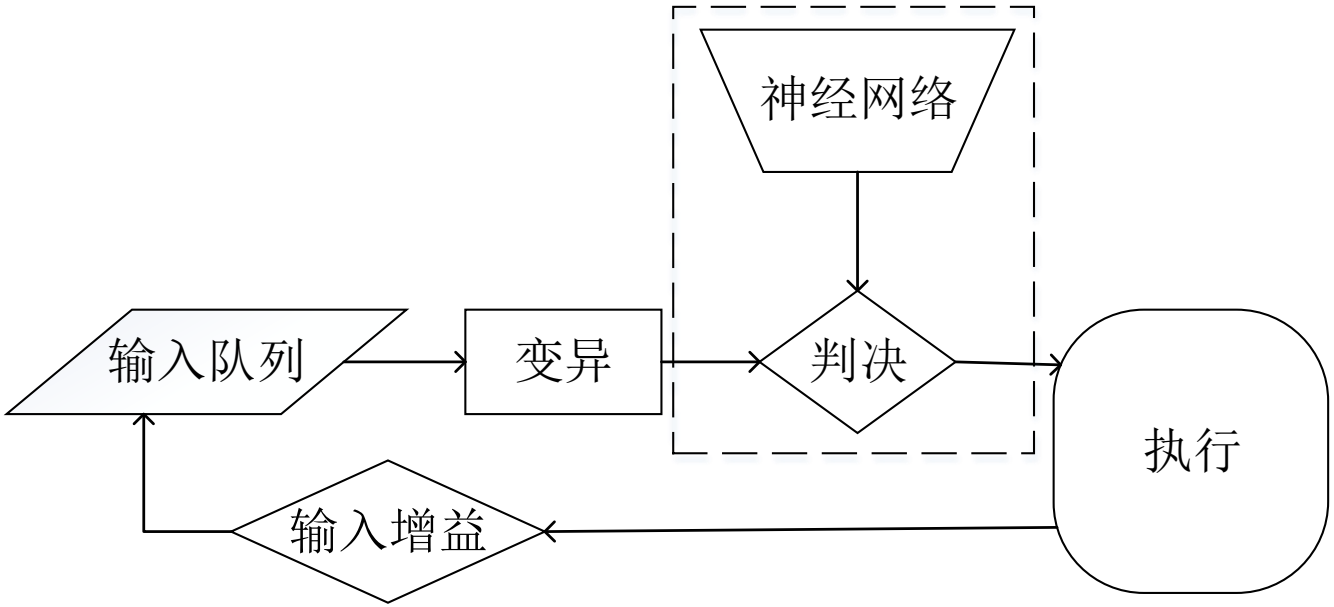
- 存在的问题
  - 针对文件的变异是随机的，随机体现在变异的位置和变异的方式上
  - 文件的格式是异构的，一般文件头或其他关键部位的变异，更容易产生输入增益

P(问题)	对输入数据随机位置进行变异难以高效触发程序漏洞
C(条件)	基于现有模糊测试器
D(难点)	自动识别变异数据的变异是否发生在关键位置
L(级别)	微软研究院作，有产品

算法原理

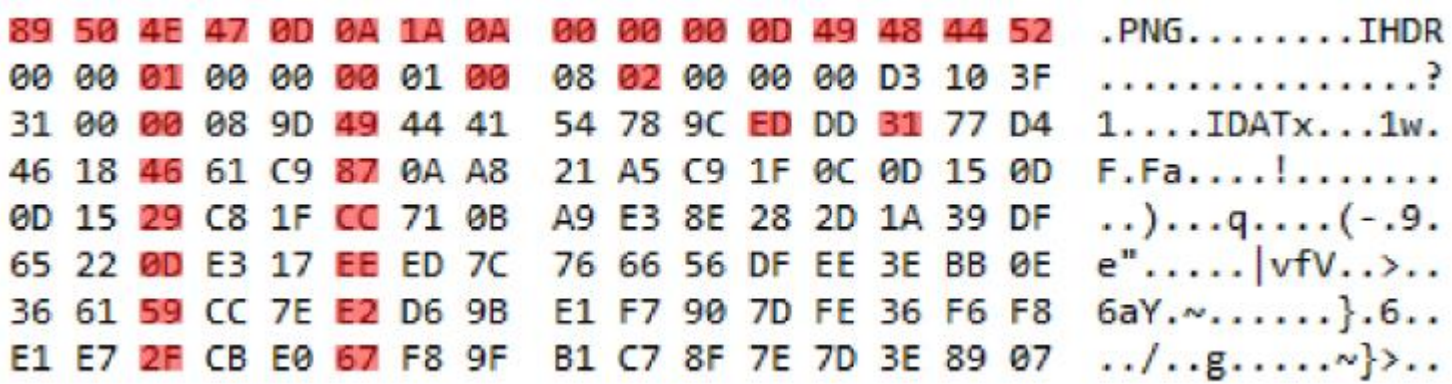
算法原理

- 核心思想
  - 神经网络：输入一条数据，输出这条数据中的“关键位置”
  - 判决：输入变异的数据，比较变异的位置与“关键位置”的重合程度，剔除重合程度低的数据



算法原理

- 神经网络
  - 目标：
    - 输入一条数据，输出这条数据的关键位置（简称热图）
      - 对应到输出数据的所有位置，指输出一组数值，代表不同位置的重要程度
    - 不同长度的输入数据对应不同长度的热图



算法原理

- 输入：
  - 由于输入数据长度可以是任意字节，需要先对数据做切分
  - 按字节切分$\{f_k:\{0\text{x}00,0\text{x}01,\dots,0\text{x}FF\}^k \mapsto [0,1]^k | k\epsilon\mathbb{N}\}$
  - 按比特切分$\{f_k:\{0,1\}^{8k} \mapsto [0,1]^{8k} | k\epsilon\mathbb{N}\}$
- 输出：
  - 输入数据中每个位置的重要程度
  - 重要程度：以代码覆盖率为标准
    - 如果变异后的代码覆盖率提高，代表变异发生在了“关键位置”，这个位置更重要
    - 记录这个变异位置，和对应的输入，作为一条训练样本，即筛选条件是$(x, x\oplus x') | s(b,b') > 0$

算法原理

- 根据代码覆盖率，筛选用于训练的样本
  - 代码覆盖率的变化表示为：$s(b,b')=\sum_{1\le i\le|b|}[b_i<b_i']$
    - b是变异前的文件执行的代码
    - b'是变异后的文件执行的代码
    - $b_i<b_i'$表示存在b_i中有，但b_i'没有的代码段

算法原理

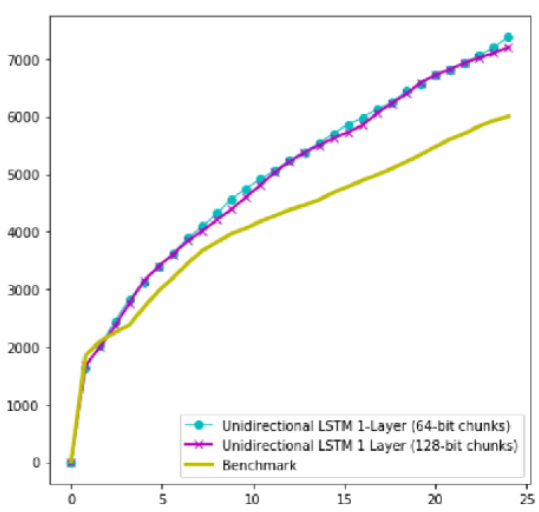
- 具体过程
  - 进行一定量的模糊测试，获得一批变异前的数据x 、变异后的数据x'、变异前后的代码覆盖率增量$s(b,b')$
  - 筛选出其中$s(b,b')$大于0的变异前后数据对(x,x'),并处理得到训练用数据对$(x,x\oplus x')$
    - 筛选条件$(x,x\oplus x')|s(b,b')>0$
  - 以变异前数据x作为模型的输入，$x\oplus x'$作为模型的输出，训练神经网络模型
    - LSTM，双向LSTM，seq2seq，seq2seq+Attn

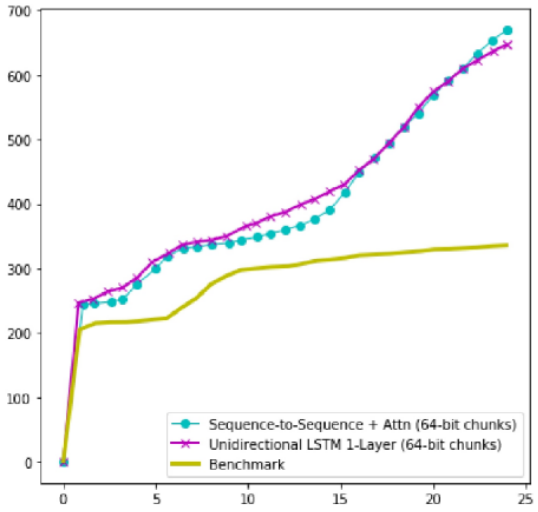
算法原理

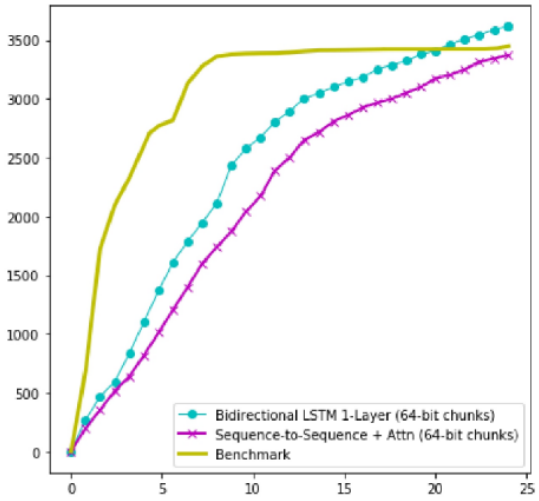
- 使用模型判决
  - 判断所有变异位置与模型给出的“热图”的重合度之和是否大于阈值
  - $\sum_k[(x \oplus x') \wedge [f(x)]] > \alpha$
    - x变异前的原始文件
    - x'变异后的输入文件
    - $\oplus$按位异或
    - $[_]$上取整函数
    - α用户定义的阈值

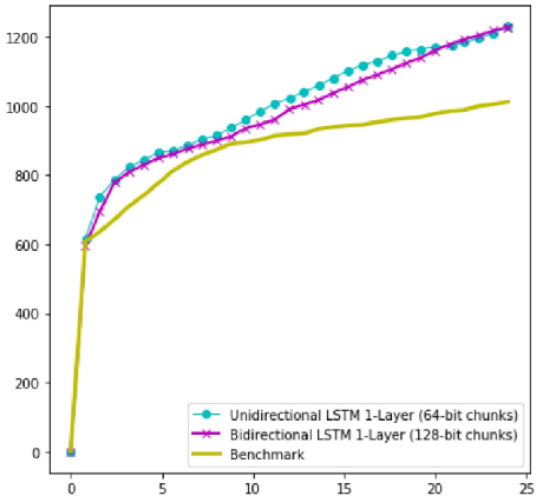
实验结果

- 四个程序readelf,readpng,mupdf,libxml的输入增益：









实验结果

- mupdf在输入增益上没有明显的改进
  - 典型的pdf文件通常非常大，影响了模型查询时间
  - 提高模型吞吐量和性能
- 未来的工作
  - 使用强化学习在线学习，以便随着模糊测试的进行，不断改进模型
  - 使用生成模型，生成应用于输入文件的变异部分

参考文献

- Not all bytes are equal: Neural byte sieve for fuzzing
  - https://arxiv.org/abs/1711.04596

知人者智，自知者明。
胜人者有力，自胜者
强。知足者富。强行
者有志。不失其所者
久。死而不亡者，寿。

谢谢！