PyFuzzer：一种自动化高效内存模糊测试方法

华中科技大学 李伟明

1.介绍

- 在漏洞挖掘领域，fuzz testing（模糊测试）是一种重要的技术。它通过向程序输入大量畸形数据来使执行程序出现问题，发现可能的漏洞。
- 优点：
- 简单方便的黑盒测试，并且已经有大量经验可以发掘各种漏洞。SPIKE，PEACH。
- 缺点：
- 测试数据量大，无法突破应用程序中的验证或检查，无法进行更为深入的测试。

1.介绍

- 在2003年的BlackHat Federal 和Blackhat USA安全会议上提出概念。2010在BlackHat安全大会上，提出内存fuzz testing原型系统,提高效率。
- 内存 Fuzzing 的方式：循环变异和快照恢复变异。循环变异是指在被测试代码片段的末尾插入一个无条件跳转，跳转到代码开头，并继续修改数据，循环测试。而快照恢复变异是指在被测试代码开头，保存进程快照，到末尾再把进程快照进行恢复，再继续变异数据，进行下一轮测试。

1.介绍

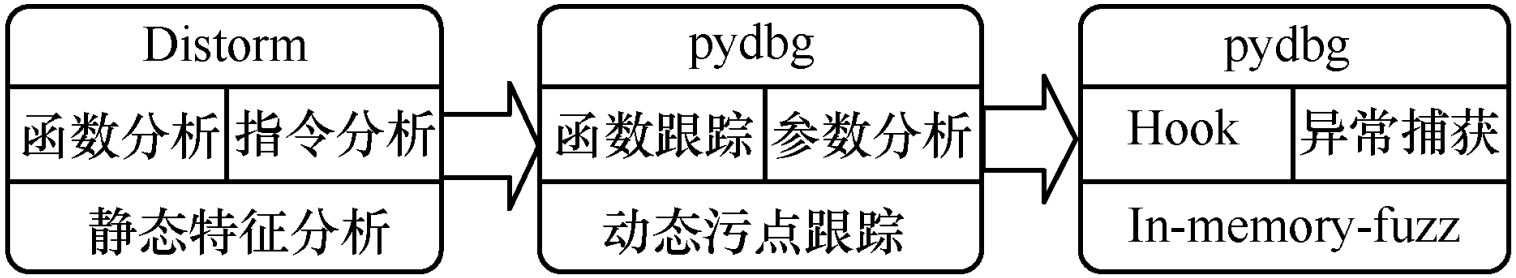
- 内存fuzz testing的优缺点：
- 优点：

  相对普通fuzz testing，不需要多次启动程序，一次启动程序可以完成大量测试，提高了测试效率。
- 缺点：

  需要较为复杂的配置，要找到合适的快照点和恢复点，手工参与较多。

2.PyFuzzer

- 本文中提出了一种基于内存fuzz testing系统PyFuzzer，综合利用了静态特征分析和动态污点跟踪的技术。在函数级别，提高内存fuzz testing的自动化。



2.1静态特征分析

- 利用了Distorm反汇编引擎对二进制程序静态解析和发掘危险函数。
- 在可执行程序的函数解析中主要使用了基于call\ret、jmp指令的函数调用方法，并利用了PE文件中的导入表来得到二进制程序的所有函数列表。
- 得到所有的函数后，并不是直接对函数进行fuzz testing测试，而是利用静态分析方法来为fuzz testing器找出潜在的高危函数，节省时间并提高效率。

2.1静态特征分析

- 主要是利用模板比对发现危险函数。
- 操作符用Operation表示。例如mov记作M，inc记作I，test记作T，JMP记作J。
- *Operator*={*x*, *i*, *r*, *a*, *m*, *f*}
- 图中的典型结构就可以记作*MrmMmrIrTrrJi*。

```
Mov     Reg , [Memory];
Mov     [Memory], Reg;
Inc     Reg;
Test    Reg;
Jxx     Addr;
```

2.1静态特征分析

- 通过Needleman算法计算出危险指令模板与指令串的Needleman距离，其值越小，相似度越高。通过对大量循环结构观察分析，选取以15作为阈值，当Needleman值小于15时，则认为是危险指令序列。包含危险指令序列的为危险函数。
- 获取危险函数的快照点和恢复点，作为内存fuzz testing的基础。

2.2动态参数分析

- 在静态分析后，会得到危险函数记录。利用Pydbg函数库对所有危险函数进行Hook。当发现被Hook 的函数对输入数据进行处理，就触发了回调函数记录该函数的入口地址，返回地址及函数参数信息。
- 通过分析危险函数的每个参数，判断该参数是否从外界接收数据。如果从外部接收数据则作为内存fuzz testing的对象。

2.3内存Fuzz testing

- fuzz testing通常作用于程序外部，根据一定策略构造畸形数据并发送给目标程序。
- 而内存fuzz testing几乎全部作用于程序内部，只需在开始进行测试时对程序发送一次数据，内存模糊测试方法就可以在目标进程内自动生成测试用例的畸形数据，其循环测试过程也不需外界干预，这样就可以减少程序I/O，提高效率。

2.3内存Fuzz testing

1. 确定模糊测试的快照点和恢复点
2. 定位输入数据位置，生成测试用例
3. 循环进行内存模糊测试
4. 记录所有出错现场信息

3.测试

- PyFuzzer对WarFTPD和Serv-U进行了测试，首先通过静态分析，实验结果如表所示。

表 2　静态分析结果

测试程序	函数数目	危险函数数目	JXX 循环拷贝结构	Rep mov
WarFTPD 1.65	1 987	362	6	432
Serv-U build 4.0.0.4	2 392	580	13	719

3.测试

- 采用了网络FTP测试工具4n FTP Fuzzer与PyFuzzer测试对比。在测试效率上，PyFuzzer测试效率为113次/s，而4n FTP Fuzzer的测试效率只有48次/s。

表 3　　对比测试结果

程序/(工具)	测试结果	漏洞类型
WarFTPD1.65 (PyFuzzer)	CWD, CDUP, DELE,NLST, LIST	DoS
	USER	Buffer overflow
Serv-U build 4.0.0.4 (PyFuzzer)	SITE CHMOD,MDTM, LIST	Buffer overflow
	XCRC, STOU, DSIZ	DoS
Serv-U build 4.0.0.4 (4n FTP Fuzzer)	SITE CHMOD,MDTM, SMNT	Buffer overflow
	STOU	DoS

4.总结

- 本文提出了综合利用静态分析和动态分析的In-memory fuzz testing测试方法。设计并实现了该方法的原型Pyfuzzer。
- 相对传统fuzz testing提高了测试效率。
- 增强了内存fuzz testing的测试过程自动化程度。