Android的全系统符号执行平台
Android_S2E

ksh@skyeye.org
北京迪捷数原科技有限公司

SAT和SMT

- SAT：
  - 一个*布尔可满足性问题的成员(instance)*是一个布尔表达式，或者说，一些布尔变量跟布尔逻辑运算符的组合。
  - 一个表达式是*可满足的*，如果存在某些给予布尔变量真值的方式，可以使这个表达式的值为真；
  - SAT是一个NPC问题。
- SAT and SMT
  - SAT：命题逻辑的表达式，a || b && c || d .....
  - SMT：一阶逻辑的表达式, F(x) || G(y) && H(z).....
- SMT tools： Barcelogic,CVC,MathSAT,Yices, Z3, STP
- 参考资料：
  - Satisfiability Modulo Theories: Introduction and Applications

起源

- King, CACM 1976.
- Key idea:
  - generalize testing by using unknown symbolic variables in evaluation
  - Symbolic executor executes program, tracking symbolic state.
  - If execution path depends on unknown, we fork symbolic executor
  - at least, conceptually

示例

- Lec13-symexec.pdf

符号执行的工具

- 符号执行工具：CUTE, Klee, DART, SAGE, Pex, and Yogi。
- 全系统的符号执行：S2E,基于全系统模拟器Qemu和符号执行工具Klee。

程序行为分析方法的列举和特点

- 静态分析：目前在处理大规模的软件效果最好。也有Coverity和Fortify这样的成熟商业工具。
- 程序切片：基本上属于静态分析。
- 模型检测：model checking
- 模糊测试：碰运气和靠人工经验的成分居多，暴力破解。挖漏洞的效果比较好，比较实用。目前已有成熟的开源工具。
  - 白盒Fuzz，灰盒Fuzz等；
- 符号执行：执行驱动的精确的路径遍历。适合那种逻辑不太复杂的场景。

各种程序行为分析方法结合

- Fuzz和静态结合
- Fuzz和符号执行结合
- 静态和符号执行结合
- 具体执行和Fuzz结合
- 具体执行和符号执行结合

符号执行的问题

- 路径爆炸
  - 符号变量作为循环的迭代；
  - 符号变量作为间接跳转或者内存引用的地址；
    - 比如 int a[x],其中x为符号变量
- 当前的解决思路：
  - 状态合并；
  - 并行执行
  - 各种启发式算法：带来了不精确性；

符号执行的问题（续）

- 环境依赖
  - KLEE：使用修改过的uClibC解决C语言的环境问题；
  - JPF：Java虚拟机；
  - S2E：1、使用全系统模拟器解决环境问题；2、处理符号执行和具体执行的切换问题。

Android_S2E项目

- 项目背景
  - 把S2E项目移植到Android模拟器
  - 项目主页：https://github.com/michaekang/android_s2e
- 项目成果
  - 可以对一个小的C语言程序进行符号分析，给出可以覆盖所有路径的输入；
  - 演示视频：http://v.youku.com/v_show/id_XNzMxNTE5MzE2.html
- 项目演示

未来的发展

- 我关注的可能的产品方向
  - C语言的单元测试用例自动生成；
  - Automatic debugging, 自动定位bug的工具；
  - 用于嵌入式系统可靠性设计的自动故障注入；

参考资料

- http://research.microsoft.com/en-us/events/z3dtu/dtu-jan2.pdf
- Satisfiability Modulo Theories: Introduction and Applications
- *KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs , OSDI'08*
- KINT, OSDI'12, 找内核的整数溢出，全局静态分析和函数内部的符号执行；
- *S2E: A Platform for In Vivo Multi-Path Analysis of Software Systems*， ASPLOS'11
- SymDrive: Testing Drivers without Devices. In OSDI'12：为没有真实外设的驱动进行测试
- Path-exploration lifting: hi-fi tests for lo-fi emulators, ASPLOS '12

欢迎大家参与到符号执行的研究中，一起讨论交流。

Q & A