符号执行简介

符号执行入门 - 知乎 (zhihu.com)

符号执行技术总结 - 知乎 (zhihu.com)

符号执行-基于python的二进制分析框架angr - blacksunny - 博客园 (cnblogs.com)

quals-2016/baby-re at master · legitbs/quals-2016 (github.com)

2020网鼎杯青龙组部分逆向题 - 『脱壳破解区』 - 吾爱破解 - LCG - LSG |安卓破解|病毒分析|www.52pojie.cn

定义：符号执行技术指的是通过程序分析的方法，确定哪些输入向量会对应导致程序的执行结果向量的方法。顾名思义，使用符号执行分析一个程序时，该程序会使用符号值作为输入，而非一般执行程序时使用的具体值。在达到目标代码时，分析器可以得到相应的路径约束，然后通过约束求解器来得到可以触发目标代码的具体值。

目标：软件测试中的符号执行主要目标是: 在给定的探索尽可能多的、不同的程序路径(program path)。对于每一条程序路径，(1) 生成一个具体输入的集合(主要能力)；(2) 检查是否存在各种错误，包括断言违规、未捕获异常、安全漏洞和内存损坏。

符号执行->形式化/静态分析

符号执行的发展：

70年代纯静态->Concolic执行 开始发展

Concolic执行：那些符号执行不好处理的部分、求解器无法求解的部分，用实际值替换就好了。使用实际值，可以让因外部代码交互和约束求解超时造成的不精确大大降低，但付出的代价就是，会有丢失路径的缺陷，牺牲了路径探索的完全性。

面临的几个问题：路径选择（启发式、程序分析减少路径）

约束求解：不相关约束消除，增量求解。

在符号执行的约束生成过程中，尤其是在concolic执行过程中，通常会通过条件取反的方式增加约束，一个已知路径约束的分支谓词会取反，然后结果的约束集会检查可满足性以识别另一条路径是否可行。一个很重要的现象是，一个程序分支通常只依赖一小部分程序变量，所以我们可以尝试从当前路径条件中移除与识别当前分支结果不相关的约束。

几个主流的工具：KLEE、基于二进制的：S2E、python开源：angr