DOUBLETAKE: Fast and Precise Error Detection via Evidence-Based Dynamic Analysis

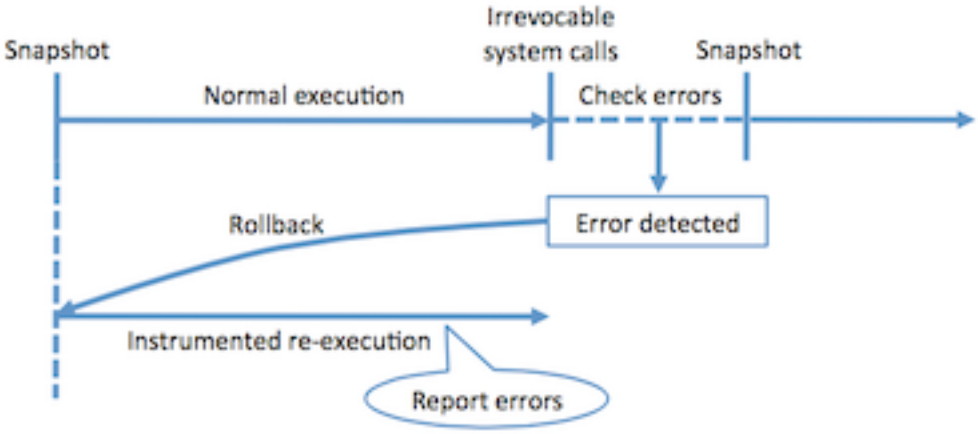
MAR 11TH, 2016

论文下载

Abstract & Introdutcion

- 缓冲区溢出、UAF和内存泄露一直存在于C和C++程序
- 但当前如Valgrind工具的动态监测overhead很高
- 这里提出的方法只有5%的overhead

Overview



- 运行前对内存做快照，然后运行

- 遇到irrevocable的系统调用时，如果遇到问题，rollback，找到造成问题的指令

Analyses

- 修改分配、释放内存的库函数

Heap Overflow

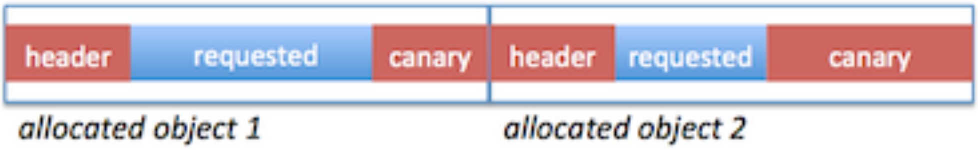


Figure 2: Heap organization used to provide evidence of buffer overflow errors. Object headers and unrequested space within allocated objects are filled with canaries; a corrupted canary indicates an overflow occurred.

- 分配内存的时候放个cookie
- 如果cookie被改写就回滚找是那条指令修改的

UAF

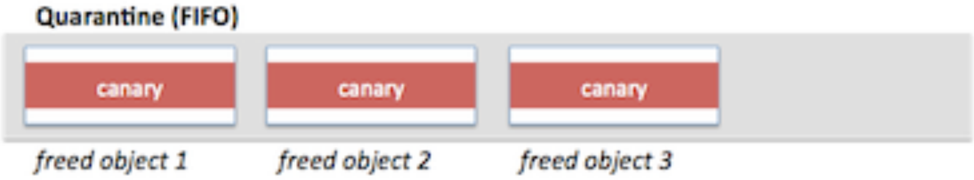


Figure 3: Evidence-based detection of dangling pointer (use-after-free) errors. Freed objects are deferred in a quarantine in FIFO order and filled with canaries. A corrupted canary indicates that a write was performed after an object was freed.

- 没释放一段内存就在这个内存上写cookie
- 如果被改写和上一步一样

Memory Leakage

- 类似conservative garbage collection

- 在执行的时候并不做处理，只在回滚的时候看一下
- 说每个对象头上有个marked bit和allocated bit
- 如果前者是1，说明被访问过
- 如果是0，把这段内存标记成可访问，递归搜索内存上的其他指针
- 指针识别是把内存上数值在堆区间的都认为是指针