一、前言

有没用办法在算法分析环节使绊子，即不阻止其模拟执行出结果，但阻止其使用*Unidbg*辅助算法分析?

一、执行时间检测

在关键执行流前后获取时间戳，即检测其执行耗时，即使手机性能再差，也不太可能执行十秒八秒，一般这意味着执行流中发生了断点调试或者单步调试，除此之外我们注意到，traceCode也十分耗时间，千万行汇编的执行流，可能需要数小时才能跑完，Unidbg的traceCode速度比IDA快很多，但依然和非trace情况差距极大。因此执行时间检测可以检测目标代码段是否被调试或者被trace指令。

二、Hook 框架检测

Unidbg在Android架构下，支持如下Hook

- hookZz
- dobby
- whale
- xhook
- unicorn 原生hook

其中hookzz是dobby前身，但在使用体验上，hookzz在arm32上更稳定，dobby在64位上效果更好，所以在Unidbg中将其作为两个框架。hookzz、dobby、whale 都是 基于inline hook 技术的hook 框架，xhook 是爱奇艺出品的plt hook 框架，unicorn 原生 hook 用的也很广，比如Unidbg的 codetrace、console debugger、traceWrite、traceRead、系统调用拦截、JNI实现 等等，都是基于原生Hook去封装和实现。

inline hook 需要篡改目标函数指令，因此校验目标代码是否被修改的一切思路都可以用于检测inline hook，简单一些的做法是检测函数开头的几个字节，更好一些的方法是对整个函数做crc32完整性校验，或者md5等函数也是可以的，但开销太大。plt hook 的检测也很成熟，下面是一些开源的方案。

SliverBullet5563/CheckGotHook

acbocai/vergil: android native anti hook

分析者如果在Unidbg中使用上述Hook框架，可以采用上述检测进行检测。只有unicorn 原生 hook 相对难以检测，因为它毫无特征。

三、尾声

其他方式想到了再补充。