## 背景

如何定位到唯一设备，是方刷单中最重要的一个点

## 技术难点

## 实现方案

多维度匹配

IMEI、IMSI、MAC、序列号、Android ID，这几个是Android设备识别比较关键的指标，但是由于Android系统自身的限制，或者说系统上无法避免的漏洞，系统API都是JAVA，被HOOK篡改的成本几乎为零，如果采用系统API获取数据，很容易被绕过去。

另外，模拟器甄别也是个很关键的点，模拟器很容易获取不同的设备特征值，而且模拟器作假成本为零，如果不能禁止模拟器下单，风控几乎无效，我们采用了多种方式识别模拟器，而且经过多个版本的迭代，不断优化，保证不误杀真机的情况下，尽可能的识别模拟器，我们从搓个维度识别模拟器，

基于特征值：不是很靠谱，后面不断

模拟器甄别优化：基于ARM及Simple X86架构（模拟器采用）的区别，识别模拟器，主要是ARM的指令缓存跟数据缓存不同步（并非100%，可以多次检测，取概率：优化点），

原理：self-modify-code，这里比较需要注意的是ARM的3级流水，在C跟ARM混合编程的时候，很容易导致崩溃，当然，模拟器并不存在这个问题，不存在积极流水，可以看做解释执行。

## 效果

就模拟器识别来看：

就风控结果来看：

## 竞品分析

破解《数字联盟》的设备指纹 （采用的是MAC地址）但是MAC地址被篡改后，指纹就变了（已验证），

破解《网易易盾》的设备指纹 采用MAC地址（篡改后改变）

我们项目：多个维度，采用积分制，MAC、序列号、IMEI、AndroidID，有一个或者两个被篡改了，也能定位到唯一设备，而且，特征值的获取方法也并非是系统的API，而是采用了C native获取或者说通过异常API，或者直接获取Binder服务来获取特征值。

模拟器甄别：单独进程识别，避免用户感知崩溃，

## 总结