业务安全感知防御平台 V6.0.0

产品白皮书

北京顶象技术有限公司

2022 年 3 月 2 日

目录

第 1 章　概述

1.1. 行业背景

1.1.1.　黑灰产产业化规模化

云计算、大数据、互联网科技的快速发展，催生出云安全、大数据安全、业务安全等安全领域，业务的快速增长及获客渠道的多样化，防护对象也由传统的 PC、服务器拓展到移动端，如 iOS 原生态 APP、Android 原生态 APP、公众号、H5 等。

黑灰产手段专业化、分工明确，攻击手段逐渐向专业化、商业化转变。据测算黑产从业人数超过 150 万人，市场规模高达千亿级别。

1.1.2. 业务安全事件频发

2020 年 11 月，浙江省绍兴市新昌县法院审理认定，犯罪嫌疑人吴某的公司非法控制老年机达 330 余万台，获取手机验证码 500 余万条，出售获利竟有 790 余万元。

2021 年 3 月，上海检方公诉的一起涉案金额超 5 亿元的虚开发票案，牵出非法人脸识别案。犯罪嫌疑人利用特殊处理的手机‘劫持’摄像头，破解了电子营业执照 App 的人脸识别系统，下载电子营业执照后，会在 App 里添加办事员的身份信息。虚开发票团伙就以此通过办事员身份使用电子营业执照。

黑灰产深入互联网、电商、航司、金融、保险等各行业场景，网络欺诈损失占 GDP 比例多达 0.63%约 4000 多亿人民币。

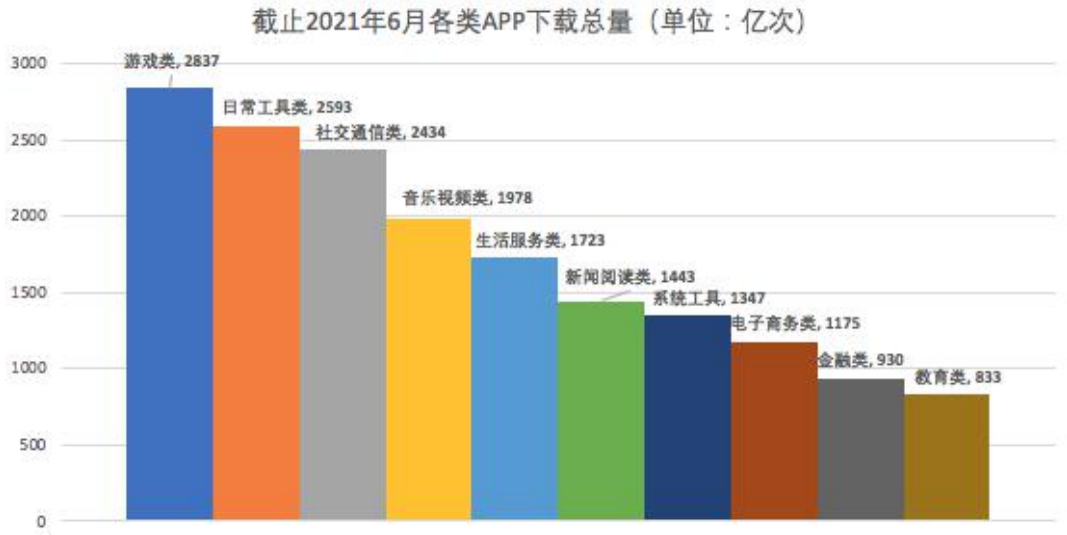
1.1.3. 传统防控手段效果有限

传统的移动安全产品旨在通过安全加固技术弥补 APP 开发中所产生的自身安全漏洞、风险，如核心代码窃取、程序逻辑破解、恶意代码注入等，但对新兴风险的发现、分析、处置等能力有限。

- 无法直接阻断风险：发现风险需要决策引擎等重量级产品参与决策，无法直接阻断；
- 防御效果有限：黑灰产针重点业务场景定点攻破，对于实时位置漂移、人脸识别绕过等风险防御效果有限；
- 数据孤岛：单点防控产品无法实现数据共享，无法联防联控赋能客户已有的上下游风控产品体系 ；
- 流程不完善：移动端安全产品只能作为风险探针使用，无风险闭环处理流程。

1.1.4. 移动端上业务安全得到广泛关注

截止 2021 年 6 月底中国手机网民规模达 10.07 亿人，较 2020 年 12 月底增加了 0.21 亿人。中国手机网民规模占总体网民规模 99.60%。智能技术发展，5G 应用下移动应用迎来更多可能性，使移动应用的场景更加广泛。 截至 2021 年 6 月底，我国国内市场上监测到的 APP 数量为 302 款。



数据显示，移动应用使用过程中，用户最关注个人隐私和支付安全保护问题占比分别为 63.8%和 63.3%。个人隐私信息、交易信息、账号信息得到关注。政府扶持加码，政策密集出台，银保监等监管部门对客户端环境安全、个人信息安全等都有明确的要求：

《个人信息保护法》（2011 年 8 月 20 日由十三届全国人大常委会表决通过，11 月 1 日起施行）

《关于防范人脸识别技术使用风险的消费提示》（银保监 2021.10.11 发布）

《中国银保监会办公厅关于银行机构网络安全漏洞引发虚假账户风险的通报》（银保监办发〔2019〕94 号）

《中国人民银行支付结算司关于加强个人 II 、III 类银行结算账户风险防范有关事项的通知》（银支付[2019]55 号）

《关于开展支付安全风险专项排查工作的通知》（银办发〔2018〕146 号）

国家对互联网空间、关键信息安全管理的要求从合规及事件管理上升到风险管理，需要企业、安全产业积极响应。在过去，关注合规要求和安全事件响应及

处置，在未来，需要合规和事件管理的基础上做到安全风险治理，发现即处置，做到防患于未然。

1.2. 痛点需求

- 需要基于设备层面的安全感知：需要能识别移动端上环境风险（root、越狱、框架软件、vpn 等)、运行攻击的检测（多开、注入、hook、调试）能力，能够识别黑灰产的各类行为。
- 需具备设备风险的即时处置能力：传统态势感知偏向流量分析，主要实现 7x24 小时对网络安全攻击方向的态势预测。需要基于设备指纹、操作行为、AI 策略模型的移动端、智能设备层面的安全感知，包括环境风险的感知、运行攻击的感知。
- 需要闭环的风险处置流程：业务人员关心当前是否存在安全问题，移动应用有没有漏洞，运行时有没有攻击，攻击来源发生在哪，能否进行有效监控及预警，能否定位到攻击位置，是否可以进行关联分析？所以对环境风险、运行时攻击、异常行为的监测、预警、具有威胁时自动触发防护策略及处置、关联关系挖掘、以及数据沉淀的闭环处置流程是切实必要的。
- 需要满足监管要求：监管加持，银保监等部门明确提出“要求利用终端威胁态势感知、客户端环境安全监测等技术及时发现并阻断恶意行为”、“应支持监控开户行为偏离多数用户的一般习惯，如在异常时间段、异常网络地址、异常地理位置等”。

1.3. 顶象观点

顶象业务安全感知防御平台通过威胁探针、实时计算、风险识别引擎、模型引擎等先进技术，集设备风险分析、运行攻击识别、异常行为检测、预警、处置为一体。对移动设备信息、安全风险信息、用户行为信息、程序运行信息进行多维度的挖掘和统计分析，并以报表、图表的方式实现多形式的数据展示和可视化管理。帮助加强应用安全防护能力，满足移动终端应用的安全保护需求。建立移动业务安全感知能力，掌控移动应用运行过程和运行状况，助力移动运营。

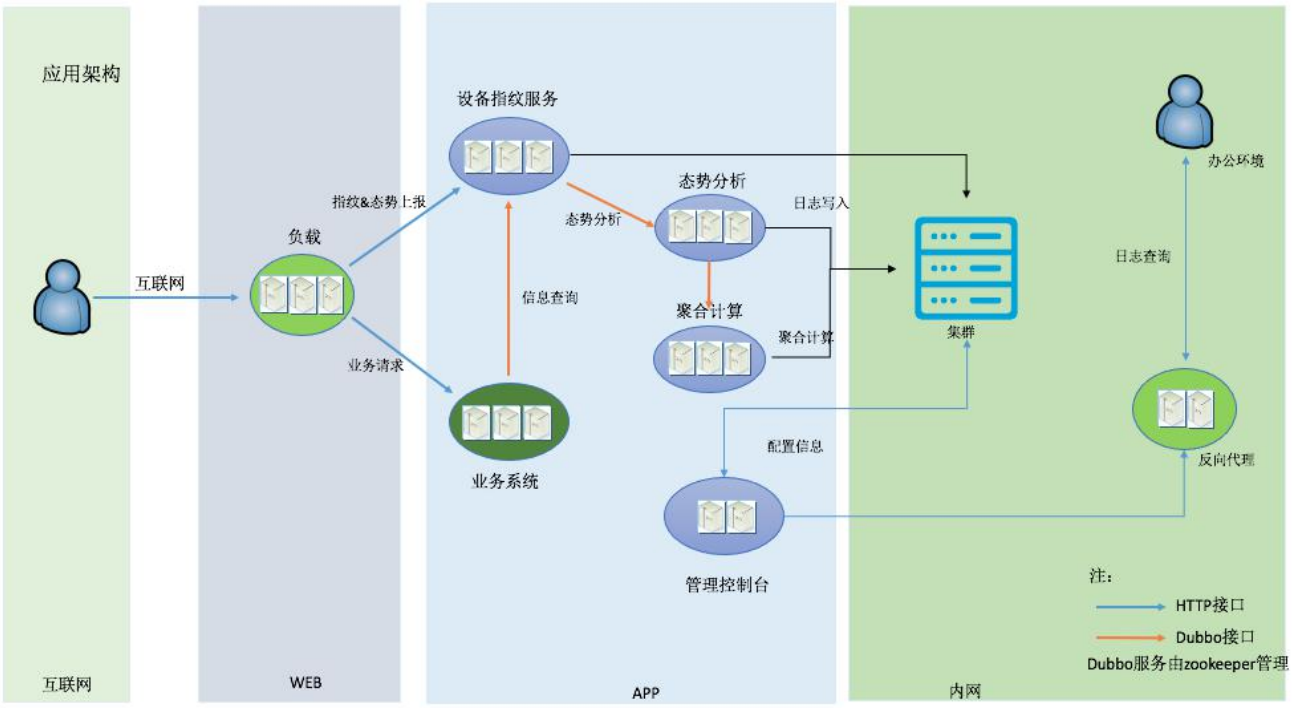
第 2 章　架构与技术

2.1. 系统架构

业务安全感知覆盖安卓和 iOS 原生态 APP、H5、微信小程序、微信公众号，原生态 APP 以 SDK 方式接入，H5、小程序、公众号通过 JS 文件集成。在后台管理界面可以查询环境相关、运行时相关的风险项。在指标特征模块可以配置行

为特征指标，例如高频更换设备、高频更换地域等指征。在防御策略管理模块，可以配置防御策略以及发现风险时的处置动作，如静默监测、程序退出等。在可视化模块，具有感知大盘、数据统计等对手机型号、风险趋势、设备画像等维度，以折线图、饼状图、雷达图等形式进行图形化展示。同时设备的基本信息、风险数据、设备评分等支持以接口形式，供客户自有的业务系统或者风控产品调用。

2.2. 部署架构



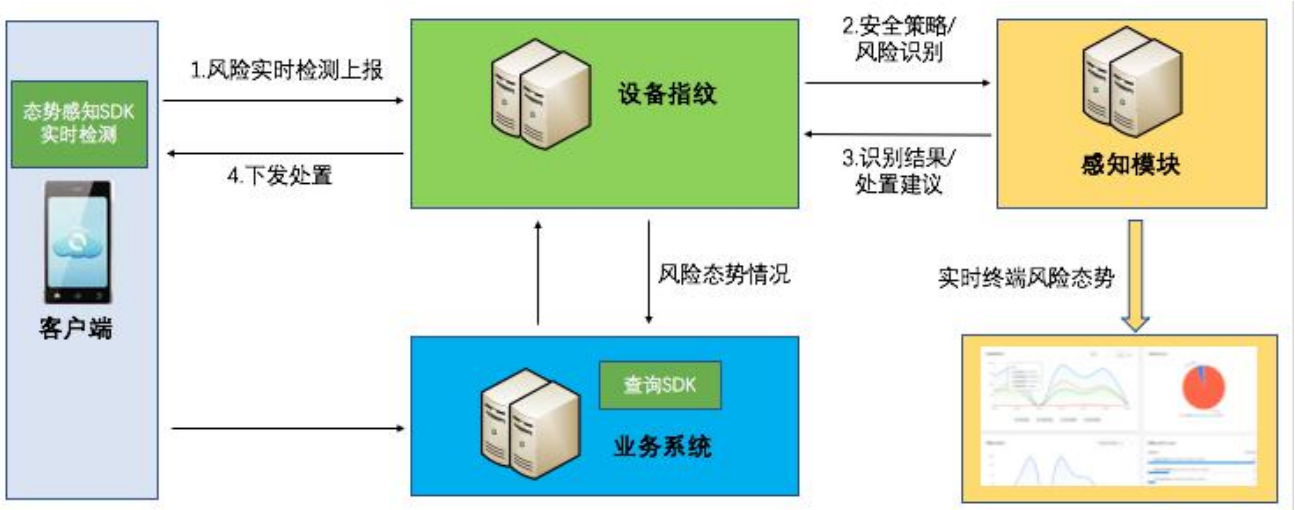
业务安全感知分为 SAAS 版和私有化本地部署版本，私有化版本部署在客户服务器。客户可根据业务情况及对数据安全的要求选择不同的版本。顶象业务安全感知私有化版本部署简单，客户端集成 SDK、JS，客户端通过 Http 接口与服务端进行交互。以 TPS 2000 为例服务器配置要求如下：

服务	CPU 核数（个）	内存（GB）	数据盘（GB）	建议主机类型	建议系统类型	数量（台）	网卡	备注
感知平台 web 应用	8	32	300	物理机/	CentOS 7.6	3	千兆	生产环境

				虚拟机				
感知平台缓存	8	32（建议64）	1000	物理机/虚拟机	CentOS 7.6	3	千兆	生产环境
感知平台数据存储	8	32（建议64）	1000	物理机/虚拟机	CentOS 7.6	2	千兆	生产环境
设备指纹存储	8	32（建议64）	1000	物理机/虚拟机	CentOS 7.6	3	千兆	生产环境
部署控制台应用	8	16	300	物理机/虚拟机	CentOS 7.6	1	千兆	生产环境

无论是 SAAS 形态还是私有化形态，前端 SDK 及 JS 的接入都比较简单，根据开发水平不同，通常在 1～2 天可以集成结束，之后可以进入调试阶段。部署时间在服务器准备完成后，2 天左右时间可以完成部署（具体依赖于客户对镜像的上传到服务器的速度）。接入和部署工作会有专门的工程师进行支持。

2.3. 交互流程



安全感知具备主动防御机制，在端上发现风险时，会自动执行防御策略并下发处置，具体交互流程如下：

a) 移动端集成 SDK、JS ，当发现存在风险时，自动将风险信息上传给安全感知的服务端；

b) 数据进入到安全感知服务端后，自动执行防御策略；

c) 命中防御策略，生成处置动作，如静默监测、弹窗、App 退出等；

d) 安全感知服务端下发处置动作到端上，端执行具体的处置动作；

e) 系统会在管理控制台生成报表数据，所有的风险趋势等情况可图形化查询；

f) 系统提供 SDK 查询设备指纹对应的设备基本信息及风险情况。

2.4. 关键技术

虚机保护：SDK 采用虚机自我保护专利技术，基于白盒的加密帮助客户端低成本接入高标准的安全保护机制，免受恶意攻击。

设备指纹：通过采集设备（浏览器）、网络、设备等维度要素，为每一台设备生成一个全球唯一的设备指纹 ID，不可被篡改。保持追踪设备，且设备指纹不会随模拟器、虚拟机、刷机改机而改变。

威胁探针：对设备环境异常检测、攻击检测，识别 Root、越狱、代理、Hook、注入、框架攻击等风险。

指标特征：自主研发的基于 Redis 高并发流式计算，可以按时间序列进行复杂计算，输出行为特征。

风险识别引擎：基于实时计算的风险识别引擎，可集成变量、指标特征、公有云数据、模型等，复杂逻辑的平均处理速度仅需 20 毫秒。

2.5. 技术参数

1) SDK 体积：iOS、Android SDK 小于 700K，JS 约 100K，小程序约 8K ；
2) 加载性能：Android SDK 加载时间为 10ms < 50ms ，iOS SDK 加载时间为 2ms < 50ms ；
3) 闪退率：Android SDK 运行 10000 次崩溃率为 0，iOS SDK 运行 10000 次崩溃率为 0，满足 1/10000 崩溃率要求 ；
4) 唯一性：通过百余项的采集要素，结合多维度权重综合决策算法，能够为每台移动设备生成唯一的设备指纹，唯一性可达到 100%；
5) 稳定性：通过多重计算、相似计算等算法，结合模型抽取高权重特征维度，可有效对抗因为部分维度篡改或获取异常对设备指纹精度的影响，可靠保证设备指纹在刷机、升级、甚至通过黑产软件修改后，设备指纹保持不变，稳定性>99.9999%；
6) 可扩展性：为了适应客户已有业务的增长需求，系统满足高可扩展性的要求，包括性能的可扩展性和功能的可扩展性。支持容器化，具备横向扩展能力；
7) 可靠性：系统在实际运行中可应对高并发的情况，以确保系统不间断提供服务支撑，保障 7*24 小时运行；
8) 安全性：采用国密算法实现安全加密需求。具备各个层次间系统的安全性，用户身份认证、权限管理控制、日志记录等安全控制功能。以确保系统与其他系统信息交换过程的安全性；
9) 高并发：系统通过自研的高速流处理技术，可保证实时交易响应时间在 100 毫秒以内，平均的响应时间在 50 毫秒以内。并发处理能力根据服务器配置支持不低于 2000TPS，目前线上客户日访问量最大可达亿级。

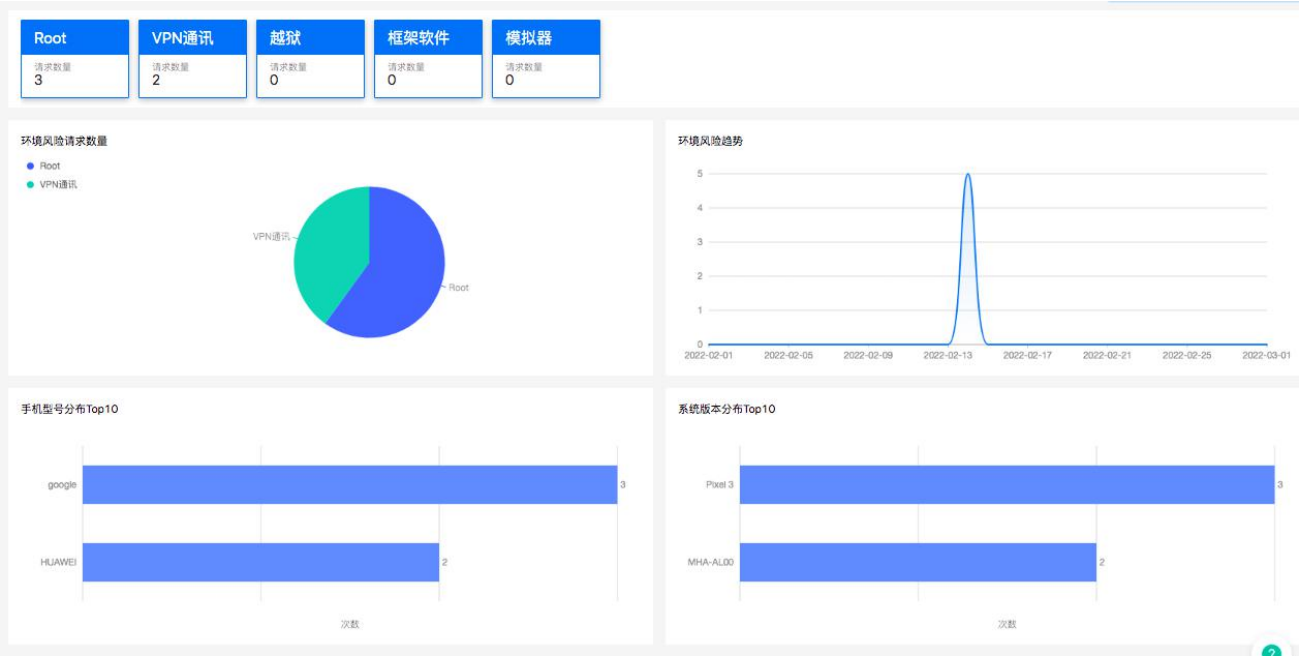
第 3 章　功能介绍

3.1. 设备信息查询

设备信息查询展示设备基础信息如 Ip、MAC、Wi-Fi 等，环境风险信息如 Root/越狱、开发者模式等，运行时攻击手段如注入、调试等，为分析人员展示设备的信息全貌。

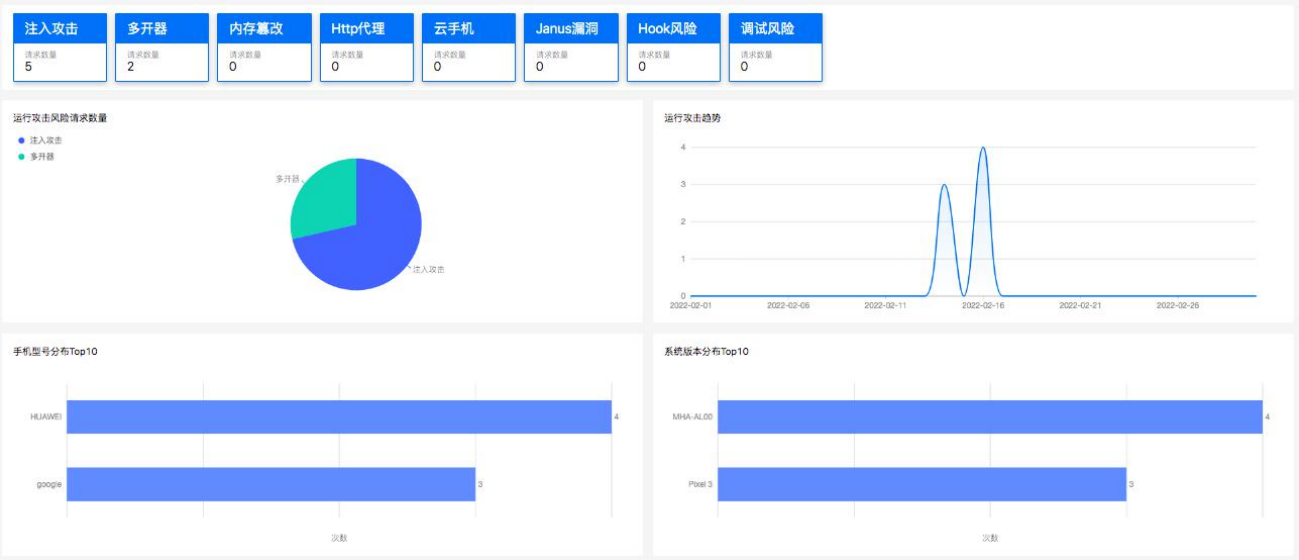
3.2. 环境风险分析

通常正常用户的设备不会存在 Root/越狱或者开启了敏感配置的操作，在风控领域，策略专家也会根据环境风险来判断用户是否为黑灰产，所以在环境风险分析模块，展示了通过威胁探针获取的环境中框架软件、风险应用、定制 ROM、界面劫持、伪造浏览器、篡改 UA 等信息。



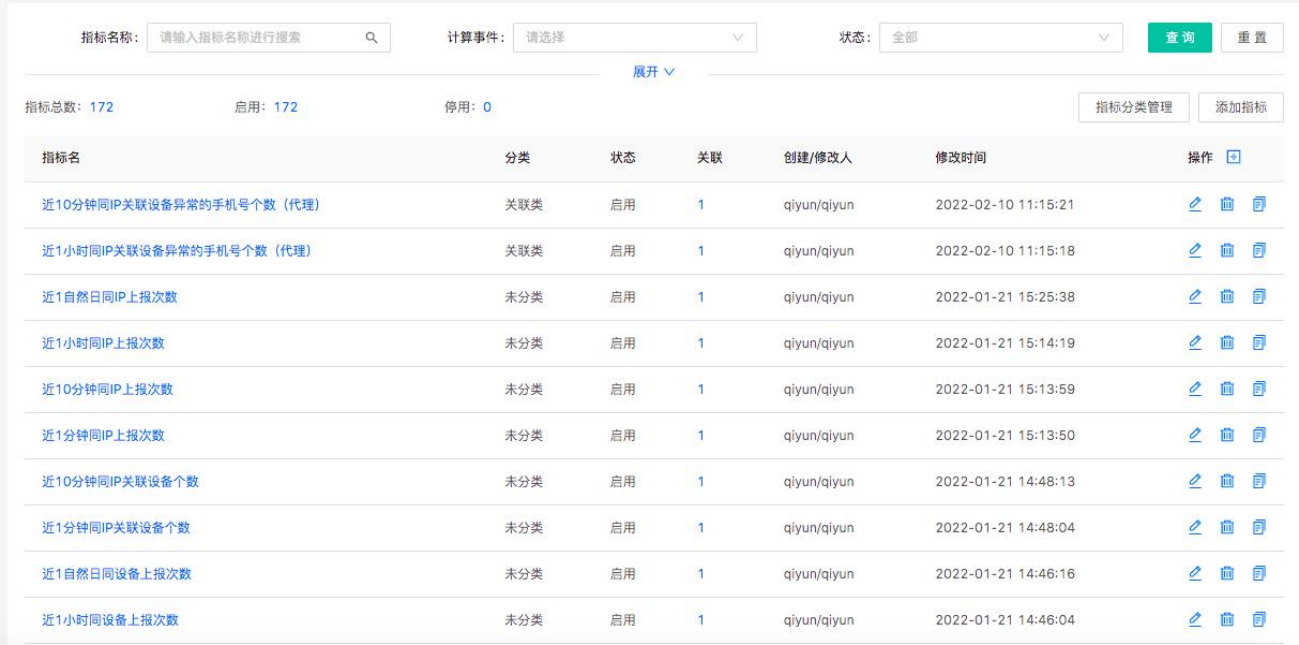
3.3. 运行攻击识别

设备环境具有风险不意味着可能存在攻击行为，但是如果在设备上检测出注入、内存篡改、调试、外挂、摄像头关键函数被篡改则基本可以判断为威胁设备且可能会执行攻击行为，设备也大概率为黑灰产所用，所以需要重点防御。在运行攻击识别模块，通过展示设备上识别到的攻击行为，为分析人员研判、防御提供重要依据。



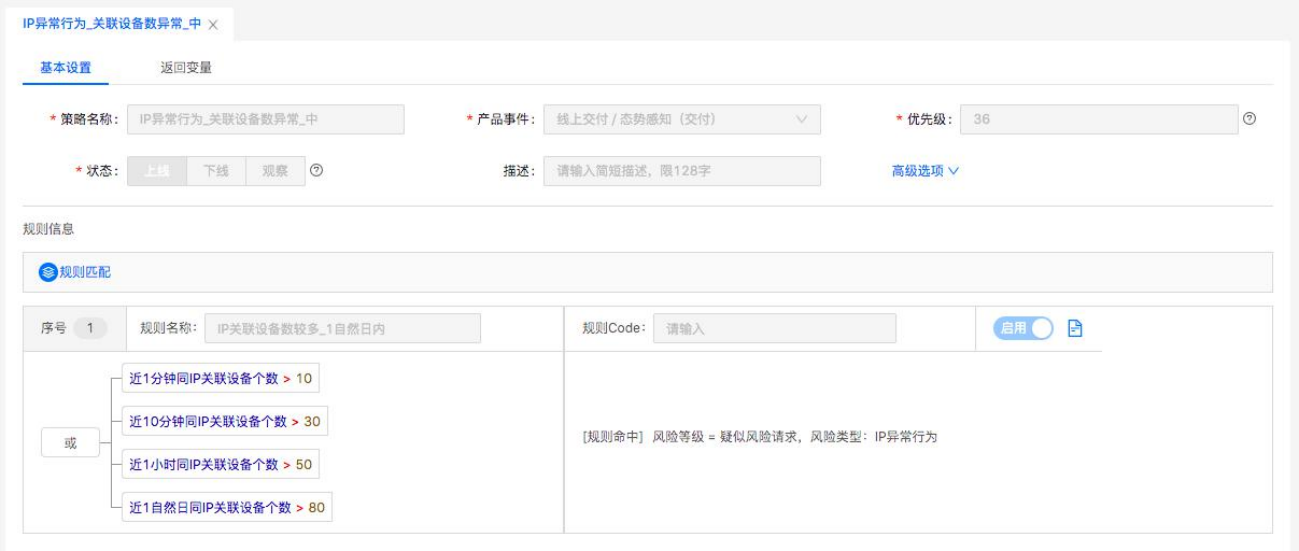
3.4. 异常行为检测

通过实时计算，系统可以实时输出当前设备某个时间段的行为信息，例如单个设备在短时间内关联多个 IP，单个设备在短时间内多次更换地点，单个设备是否出现在敏感时间等。设备的环境信息结合历史数据，可以更精准的判断设备存在的威胁，预测可能发生的攻击。



3.5. 防御策略管理

在确认威胁之后，系统支持自动执行防御策略，设备执行命中防御策略之后对应的防御处置，可以有效阻断风险操作。在防御策略管理模块，业务人员可以配置设备维度、异常行为等维度的策略，并支持上线审核等操作。

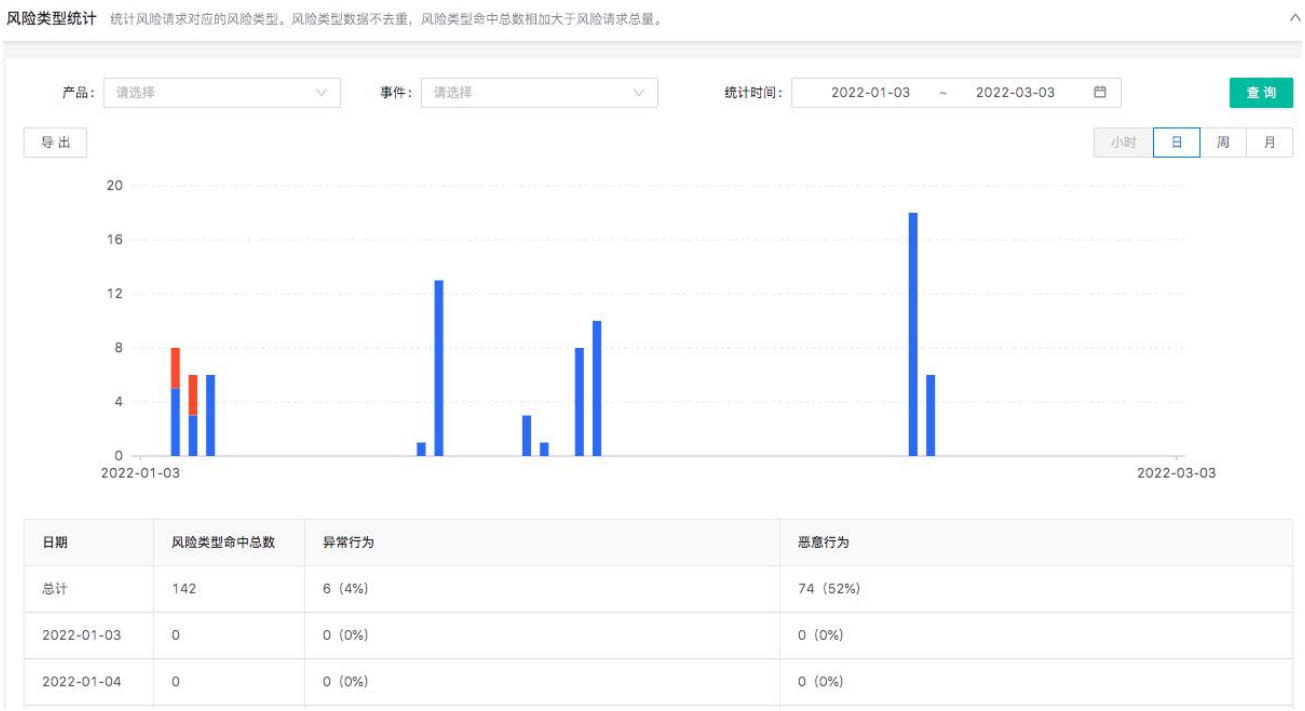


3.6. 设备画像

对历史数据的挖掘，关联分析之后，形成了设备画像。设备画像支持按设备指纹、设备指纹 token 来查询设备的画像，设备画像包括设备的基础属性、近期行为属性、某一时间段的常用地，某一时间段 IP、指纹的关联信息。

3.7. 数据统计

数据统计为分析人员以折线图、列表等形式展示态势下的各类报表数据，包括威胁趋势统计、防御策略命中统计、威胁详情等数据，通过丰富的报表展示态势发展趋势，方便分析人员进行交叉多维分析，以应对更多的未知威胁。



3.8. 监控预警

系统具备风险通报及态势预警机制，包括防御策略的命中情况，威胁数量环比增加、降低等维度预警，支持可以自定义监控周期、告警接收人。同时系统也支持对本身的服务器、中间件、应用的预警，以保障系统的稳定运行。

3.9. 私有化与 SAAS 功能区别

私有化版本具有部门概念，支持多法人的数据隔离。在防御策略层面，私有化支持自定义防御策略，SAAS 版本则由顶象安全专家统一维护。私有化版本具有系统监控模块，供客户运维人员监控预警，SAAS 版本的机器相关性能，则由顶象运维人员负责维护。在报表层面，私有化版本具有风险大盘功能，SAAS 暂无，但 SAAS 版本在设备层面的统计数据有更多的报表，如环境风险分析、云心攻击检测。所以如果对数据要严格要求，需要存储在本地，对法人需求，需要自主配置防御策略，建议选择私有化版本；如果需要快速接入，减少运营成本则建议选择 SAAS 版本。

类型	功能	菜单/接口名称	SAAS	私有化部署	说明
前端接入接口	设备指纹采集	Web环境数据采集	✓	✓	Web JS
		微信小程序环境数据采集	✓	✓	微信小程序插件
		APP设备环境数据采集	✓	✓	iOS SDK、Android SDK
后端服务接口	设备指纹服务	设备指纹生成	✓	✓	接受前端采集的信息，计算生成设备指纹，存储前端采集信息和设备指纹
		设备信息查询	✓	✓	服务端接口，根据设备指纹关联信息，查询对应设备信息
控制台	风险大盘	风险大盘	✗	✓	展示设备维度的风险标签，设备分布，上报情况
	态势概览	态势概览	✗	✓	根据时间展示态势趋势，风险事件分布
	终端风险感知	环境风险分析	✓	✗	环境层面的风险检测，如Root、越狱、VPN等
		运行攻击检测	✓	✗	运行时的攻击检测，如注入、Hook、多开灯
	数据赋能管理	字段、模型、名单等	✗	✓	防御策略里用到的变量管理
	防御策略管理	场景管理	✓	✓	态势接口里可选择是否上传场景，可按照场景来配置执行的防御策略
		策略配置	✗	✓	防御策略配置，策略里可以使用字段、名单、模型等变量
		策略实验室	✗	✓	对防御策略进行试验，试验后可上线使用
	统计报表	态势波动统计	✓	✓	使用折线图、列表的方式展示态势的波动趋势和态势的统计数据
		防御策略统计	✓	✓	使用饼状图、排序列表的方式展示防御策略命中排行及风险处置的分布情况
	设备画像	数据统计	✓	✓	按Android、iOS、Web的维度展示昨日、近7日、近30日设备的访问趋势
		设备分析	✓	✓	以设备的维度展示设备的风险命中和分布情况，如模拟器、调试、越狱等风险
		设备画像	✓	✓	可以对某一个设备指纹或指纹凭证进行设备画像分析，设备画像有上一次访问属性、所有访问历史、近7日常用地、近七日关联信息
	业务监控	态势请求监控	✓	✓	态势防御策略的命中情况查询，进入到系统的所有数据皆可查
		设备采集监控	✓	✓	可以对风险进行溯源，包括设备的详情、风险标签、IP关联、账号关联等数据分析
		态势监控报警	✓	✓	根据设定的态势情况，给予邮件告警通知
		态势报告管理	✓	✓	根据设定的时间点，以邮件形式定时发送数据报告
	系统监控	应用节点监控	✗	✓	监控应用节点状态
		服务器性能监控	✗	✓	监控服务器的CPU、内存使用率、负载、网络等硬件状态
		接口调用监控	✗	✓	监控设备指纹接口的调用时长及调用次数
		中间件监控	✗	✓	监控中间件慢查询、连接数等情况
		监控报警管理	✗	✓	可以对应用节点、服务器性能、接口调用、中间件进行报警阈值设置，触发报警后，可以发报警邮件给相关人
	系统管理	产品管理	✓	✓	对设备接入时应用的AppId、AppSecret进行管理
		部门管理	✗	✓	系统支持多部门
		账号管理	✓	✓	部门管理员、超级管理员可进行账号的增删改查
		权限管理	✗	✓	系统内置超级管理员、安全管理员、部门管理员，并支持添加角色

第 4 章　顶象优势及适用场景

4.1. 优势特点

4.1.1.　覆盖原生态 APP、H5、小程序、公众号

4.1.2.　针对威胁的主动防御机制

顶象业务安全感知在识别风险，多维分析的基础上，提供了与终端、业务紧密结合的响应处置能力:

- **终端响应处置**:在客户端进行响应处置,第一时间在终端处置对应的风险,对于高等级风险或者核心操作可以采取此种方式。
- **业务响应处置：与业务、风控系统结合，把终端发现的风险，以及对应的策略分析结果输出给业务/风控体系，再结合业务流程进行人工处置、加黑等操作。**

风险检测	终端可选响应手段	业务端可选响应手段
重打包监测	APP退出、中断流程	中断业务、黑名单
内存监测	APP退出、终端提示	打标监控、灰名单
调试器监测	APP退出、终端提示	打标监控、灰名单
设备root监测	APP退出、终端提示	打标监控、灰名单
代码注入监测	APP退出、中断流程	中断业务、黑名单
代码Hook监测	APP退出、中断流程	中断业务、黑名单
APP多开	终端提示、中断业务	打标监控、灰名单
模拟器监测	终端提示、中断业务	打标监控、灰名单
摄像头劫持	APP退出、终端提示、中断业务	中断业务、黑名单、人工审核
……	……	……

4.1.3. 完善的风险审查机制

顶象业务安全感知具备行业内领先的风险审查机制，覆盖自动、手动生成风险事件，认领、分配机制及案件沉淀的整个流程。

4.1.4. 重点场景的实时风险监测能力

针对人脸识别等重点场景及关键操作的实时风险监测，根据黑灰产的每一步步骤，针对性的监测如摄像头遭劫持、设备伪造等行为，并触发主动防御机制进行处置，满足监管对某些新兴技术及业务流程的监管要求。

4.1.5. 行业领先的系统开放能力

具备行业内领先的系统开放能力，包括处置自定义、名单数据管理、公有云数据对接、指标特征输出，可以为客户自有的决策引擎、机器学习平台赋能，提升整体风控产品的防控能力。

4.2. 适用场景

4.2.1. 互联网类客户

- **客户分布**：电商、航旅、出行、社交、直播、游戏等。
- **共性需求：**
  - ✧ 在注册、登录、薅羊毛、订单查询、积分兑换等场景有较强安全需求；
  - ✧ 对 Root、多开、越狱等典型设备风险，需要发现即处置；
  - ✧ 配合决策引擎使用，增加设备维度防控策略，打造完善的全链路风控体系。
- **匹配场景：**账号注册、登录、营销活动、刷榜刷单、机器秒杀、恶意占座、盗刷积分、虚假用户套利等。

4.2.2. 金融保险类客户

- **客户分布：**政策性银行、国有大型商业银行、股份制银行、城商行、民营银行等、保险公司等。
- **共性需求：**
  - ✧ 严控合规风险，个人信息保护法的出台，银行业监管会越来越严，违规成本也将越来越高；
  - ✧ 针对人脸识别绕过、视频流劫持、通过技术手段劫持用户侧视频流欺骗行方等风险较为突出；
  - ✧ 反欺诈体系的持续完善，AI 新型技术的应用，需要设备层面数据为基础，为风控、模型建设赋能。
- **匹配场景：**账号注册、登录、营销活动、盗卡盗刷、人脸绕过、团伙骗贷、中介代办、打卡作弊等。

第 5 章　典型案例

5.1. 出行 APP 营销反欺诈

- **业务现状：**

- ✧ 国内头部出行 APP，目前 APP 终端+微信端用户超 1 亿，月活跃用户 1000 万+，整体业务重度依赖移动互联网；
- ✧ 目前保持营销费用的持续投入，成为黑灰产的关注目标，存在营销费用的薅羊毛风险，黑灰产参与养号、使用虚假设备等手段进行攻击；
- ✧ 客户已使用加固，手机号反欺诈数据等手段，防范部分风险，但仍有持续攻击。

➢ 客户需求：

- ✧ 有效识别异常用户，在营销活动与费用投放中对异常用户进行有效的规避；
- ✧ 增强设备维度风险的识别，提升黑灰产攻击的成本；
- ✧ 高效的识别能力，降低人为参与判断的成本。

➢ 解决方案：

- ✧ 建设设备层面业务安全感知，识别客户端运行环境、操作行为等异常信息，并提供给风控系统作为决策依据；
- ✧ 通过决策引擎配置应对策略，自动化解决流程中的问题，减少人工介入，进一步降低了人为操作的风险点。

➢ 落地成果：

- ✓ 运行期间平均每日更新设备终端信息超 300 万，峰值设备数近 1000 万；
- ✓ 每日识别近 10 万风险设备；
- ✓ 有效提升营销投放效率超 20%。

5.2. 保险代理人反欺诈

➢ 业务现状：

- ✧ 国有特大型金融保险公司，世界 500 强，个人业务销售人员超过 100 万，日常通过 APP 远程打卡实现销售人员的高效管理；
- ✧ 保险代理人通过黑灰产恶意破解工具对险司内部的打卡 App 进行破解，通过摄像头劫持注入预制人脸图像，VPN 篡改手机 IP、WIFI 地址等手段，实现无需到场即可自动化虚拟打卡，完成日常考勤，非法获取奖励。

➢ 客户需求：

- ✧ 识别异常打卡代理人和团队，增强代理人管理手段，避免销售费用的无效浪费；
- ✧ 尽可能自动化完成识别与处置，在低误杀低投诉的基础上，降低人力的工作量。

➢ 解决方案：

- ✧ 在业务安全感知方案的基础上，结合实时风险决策引擎与无感验证等产品，构建打卡反欺诈系统；
- ✧ 通过感知代理人操作中的设备环境异常，以及摄像头劫持等欺诈手段，识别虚假设备，打卡异常等各类风险；
- ✧ 通过决策引擎配置应对策略，自动化解决流程中的问题，减少人工介入，进一步降低了人为操作的风险点。

- **落地成果：**
  - ✓ **第一阶段运行期间，某省自动化识别超过 1 万名虚假代理人；**
  - ✓ **每月阻止超过 10 万次违规打卡操作；**
  - ✓ **挽回该客户每月 500 万+的代理费用损失。**

5.3. 监管合规风控体系升级

- **业务现状：**

某商业银行因拓展小程序、公众号业务渠道，原指纹能力有限无法覆盖新渠道的安全需求，同时行内需要对风控、反洗钱等系统升级，需要获取更多的设备黑白样本数据。

- **客户需求：**
  - ✧ **对个人网银、企业网银、手机银行、微信银行、网贷 APP 等非柜面渠道接入，能够获取设备唯一标识；**
  - ✧ **能够对设备进行安全检测，防范设备上作弊行为。**
- **解决方案：**

设备端通过集成 SDK，小程序、公众号集成 JS 的方式，进行安全感知，为决策引擎提供了设备维度的策略，包括设备指纹异常、模拟器检测、多开检测、注入、Hook 检测等维度，可有效分辨正常用户及黑灰产的设备特征。

- **落地成果：**
  - ✓ **2020 年 6 月完成生产环境上线，对接了风控、反洗钱、机器学习等平台，并开始积累设备风险数据；**
  - ✓ **满足监管对跟人信息保护及客户端环境安全检测等要求。**

附接入说明

私有化版本接入请联系销售经理获取，SAAS 接入概要如下：

前端接入

安卓接入

态势感知接入要求：在设备指纹接入完成，getToken 获取到 token 之后开始

1.初始化或主动检测风险

Map<String,Object> updateSituation(String url,String appId, HashMap<String, String> paramsMap)

其中指纹中如果已经指定了 paramsMap，则可不传：

Map<String, Object> result = DXRisk.updateSituation("https://constid.dingxiang-inc.com/udid/sa-m", appId,null);

2.通知回调接收：

通过广播 BroadcastReceiver 接收，通过 action 判断： a.RESULT_SITUATION

```
 private void registerSituationReceiver(){

        mReceiver = new BroadcastReceiver() {

            @Override

            public void onReceive(Context context, Intent intent) {

                if (intent.getAction() != null && intent.getAction().equals(a.RESULT_SITUATION)) {

                    String action = intent.getStringExtra("action");

                    Log.e("SituationNotify",action);

                    String msg = intent.getStringExtra("msg");

                    if (!TextUtils.isEmpty(action)) {

                        updateAction(action, msg);
```

```
                    }

                }

            }

        };



        LocalBroadcastManager manage = LocalBroadcastManager.getInstance(this);

        manage.registerReceiver(mReceiver,new IntentFilter(a.RESULT_SITUATION));

    }
```

3.回调说明

action：回调根据后台设置的 action 来处理移动端的逻辑。

如服务端配置这些值'toast' | 'death' | 'relogin' | 'limit'|'no action ' | '...'

如示例中 toast 表示 toast 提示；death 表示客户端强制退出；relogin 表示强制重登；limit 表示访问 受限，即一些关键页面无权访问；no action 表示不做操作。 客户也可以根据自己的需要在服务端 进行配置，在客户端进行 action 的实现

msg：后台设置的操作理由，风控理由等，客户端可以直接 toast 输出给客户

4.业务场景上报

在易被攻击的节点上报当前业务场景

beginScene(string)

业务场景完成时调用 endScene()表示该节点已完成

endScene()

iOS 接入

态势感知接入需先接入指纹并调用

使用示例:

```
// 整个过程由于是耗时操作，必须要在非主线程上执行，否则会阻塞 UI。如果本身已经在非 UI 线程上执行，则不需要另开线程
        dispatch_queue_t dxrisk_queue = dispatch_queue_create("com.xx.xx", DISPATCH_QUEUE_CONCURRENT);
        dispatch_async(dxrisk_queue, ^{
            // 根据业务逻辑，填充自定义字段(非必填)
                NSMutableDictionary *params = [NSMutableDictionary new];
             //        params [DXRiskManagerKeyUserId] = @”123456”;
            //开启线上数据备份
            //        params [DXRiskManagerKeyBackup] = DXRiskManagerKeyBackupEnable;
            /* 私有化配置(必填) */
        NSString *appId = @"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
        //设备指纹 URL
        params [DXRiskManagerKeyURL] = @"https://constid.dingxiang-inc.com/udid/m1";
```

```
//态势感知 URL
params [DXRiskManagerSituationUrl] = @"https://constid.dingxiang-inc.com/udid/sa-m";


    /*
    获取 token
    注意：token 最好不要保存在某个局部变量或者字段，每次使用时，都通过 API 获取。
    环境初始化
    */
    BOOL isSuccess = [DXRiskManager setup];
    NSLog(@"setup success: %@" , isSuccess ? @"YES":@"NO");
   // 获取设备指纹 token
    // 获取设备指纹 token
    NSString *token= [DXRiskManager getToken:appId extendsParams:params WithSid:""];
    NSLog(@"token: %@", token);
    // TODO 把 token 通过 Post 请求，传到业务后台。
    // 下面是模拟频繁调用的过程
    while(TRUE) {
```

```
                NSLog(@"token: %@",  [DXRiskManager getToken:appId extendsParams:params
WithSid:""];



                [NSThread sleepForTimeInterval:.5];

            }

        });
```

在适当的位置添加通知:

```
NSNotificationCenter *no = [NSNotificationCenter defaultCenter];

    [no addObserver:self selector:@selector(test:) name:@"DXRISKSITUATION" object:nil];
```

通知回调根据后台设置的 action 来处理移动端的逻辑。

如服务端配置这些值'toast' | 'death' | 'relogin' | 'limit'|'no action ' | '...'

如示例中 toast 表示 toast 提示；death 表示客户端强制退出；relogin 表示强制重登；limit 表示访问 受限，即一些关键页面无权访问；no action 表示不做操作。 客户也可以根据自己的需要在服务端 进行配置，在客户端进行 action 的实现

业务场景上报

在易被攻击的节点上报当前业务场景

beginScene(Nsstring)

业务场景完成时调用 endScene()表示该节点已完成

后端接入-设备信息查询

SDK 接入方法说明

Java SDK 接入（PHP 、HTTP 等接入方式请联系销售经理获取）

包的引入

```
<!-- 将 constid-client-sdk.jar deploy 到自己的 maven 私有仓库或 insall 本地，然后添加依赖：
-->
```

Java 使用示例

```
public class Demo {
 private static String appId = "你的 AppID";
 private static String appSecret = "你的 AppSecret";
 private static String token = "SDK 里面获取到的 token";
 public static void main(String[] args) {
     // 填写设备指纹域名或者 url 如：http://127.0.0.1:8080
     String result =
DeviceFingerprintHandle.getDeviceInfo("https://constid.dingxiang-inc.com",
appId,appSecret, token);
        }
}
```

业务场景风险查询

```
public class Demo {

 private static String appId = "你的 AppID";

 private static String appSecret = "你的 AppSecret";

 private static String token = "SDK 里面获取到的 token";

  //可以在这里扩展传入业务数据，可选

  Map<String, Object> data = new HashMap<>();

  data.put("card", "6534443xxxxxxxxx");

  data.put("user_id", 456799324); //  用户 ID

  data.put("phone_number", "13800138000"); //  手机号



 public static void main(String[] args) {

      //  填写设备指纹域名或者 url 如：http://127.0.0.1:8080

      RiskResponse result =
DeviceFingerprintHandle.getRiskInfo("https://constid.dingxiang-inc.com", appId,appSecret,
sid, bussinessSign,data);

        }

}
```

HTTP 接口入参

URL: udid/api/deviceRisk
Method:POST

- 接口参数

字段	类型	描述
appId	String	应用公钥，需要同客户端保持一致
sign	String	签名，计算方法 MD5(appSecret+businessSign+appSecret)
sid	String	业务会话 id 或者其他业务标识
businessSign	String	客户端 getToken(sid)返回的数据
data	String	其他参数，以 map 形式存放，以 json 传输

返回数据

字段	类型	描述

code	String	错误码，0-成功
success	String	接口是否成功
riskLevel	String	风险等级，默认 ACCEPT/REVIEW/REJECT
hitPolicy	String	命中策略名称
hardId	String	设备指纹
deviceRisk	Map	设备风险特征
action	String	建议处置动作
extraInfo	Map	其他信息

deviceRisk 说明

字段名	字段描述	类型
token	设备 token	String
deviceType	设备类型	String
constId	设备指纹	String
producter	生产厂商	String
macAddress	mac 地址	String

emulator	模拟器运行	Boolean
isRoot	是否 root，仅对安卓设备	Boolean
isMultirun	是否多开，仅对安卓设备	Boolean
isInject	是否存在注入风险	Boolean
isMemdump	是否存在内存 dump 风险	Boolean
isDebug	是否存在调试风险	Boolean
isHook	是否存在 hook 风险	Boolean
isJailBreak	是否越狱，仅对 iOS 设备	Boolean
vpn	是否使用 vpn	Boolean
isProxy	是否使用代理	Boolean

saData 说明

字段名	字段描述	类型
last_report_time	该设备上一次态感知成功上报时间	date
last_fail_times	最近态势感知上报失败次数	int