移动安全

Mobile Security

第一章　移动安全导论

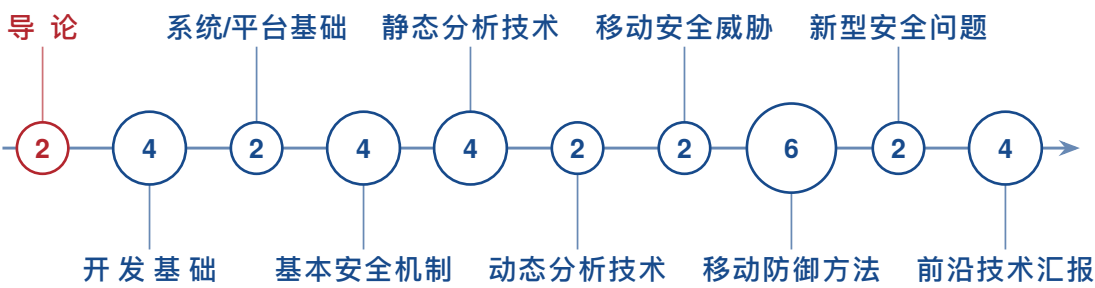
王寅

天津大学　网络安全学院

wyin@tju.edu.cn

课程知识安排

圈内数字为学时，共 32 学时



参考教材与学习资源

教材

杨珉、杨哲慜、张源等：《移动安全》
清华大学出版社，2024 年 12 月。

教辅

William Stallings、Lawrie Brown
Computer Security: Principles and Practice
机械工业出版社，2019 年 3 月。

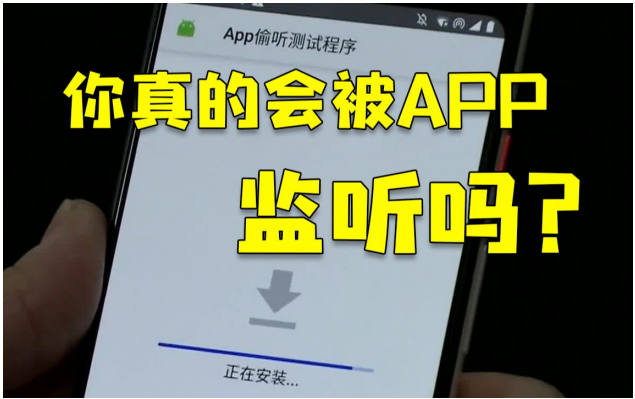
技术社区

看雪论坛　https://bbs.kanxue.com/
吾爱破解　https://www.52pojie.cn/

考核方式

1. 课堂参与：比例 20%，考察出勤和课堂研讨参与；
2. 项目报告：比例 40%，1 份移动安全综合分析报告；
3. 技术汇报：比例 40%，小组汇报。

手机“监听”与精准广告推送



视频链接

你相信你的手机吗?


App偷听测试程序
演示画面
点击模拟APP偷听场景
2"
12
0"
一个到底:手机APP“偷听”真的存在吗?
专家演示 手机APP真的会“偷听”吗?

你相信你的手机吗?

它能“偷听”吗? → 移动 + 系统安全

你相信你的手机吗?

它能“偷听”吗? → 移动 + 系统安全

它会“偷听”吗? → 移动 + 应用安全

你相信你的手机吗?

它能“偷听”吗? → 移动 + 系统安全

它会“偷听”吗? → 移动 + 应用安全

它“偷听”了什么? → 移动 + 数据安全

移动互联网已融入日常生活

支付与生活服务

9.3 亿

支付宝月活（2024 年 12 月）

购物与消费

9.4 亿

淘宝月活（2024 年 12 月）

持续扩大的安全挑战

安全漏洞

2023.09 BLASTPASS

iPhone 遭“零点击”攻击

原因

消息附件中的恶意图片触发系统处理缺陷，无需用户点击。

危害

攻击者可植入 Pegasus 间谍软件，使设备遭到监控。

恶意软件

2024.04 Brokewell

伪装成浏览器更新的木马

原因

诱导安装恶意 App，滥用无障碍服务，叠加假登录界面。

危害

窃取凭据、读取屏幕并远程操作手机，威胁账户资金。

隐私泄漏

2023.02 GoodRx

健康信息被披露给广告平台

原因

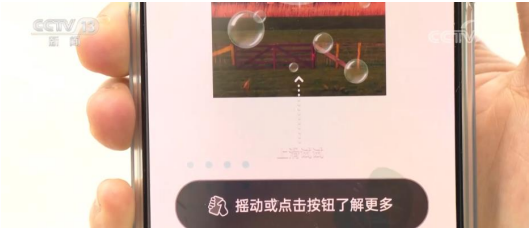
据监管指控，网站与 App 通过追踪工具向第三方披露敏感信息。

危害

用药与健康信息进入广告画像，侵害用户隐私。

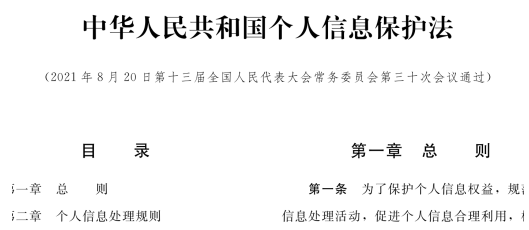
生态乱象和国家治理

摇一摇广告



手机轻晃就跳转广告，容易把无意动作当成点击意愿，打断正常使用。

个人信息保护法



中华人民共和国个人信息保护法

（2021 年 8 月 20 日第十三届全国人民代表大会常务委员会第三十次会议通过）

目　　录

第一章　总　　则

第二章　个人信息处理规则

第一章　总　　则

第一条　为了保护个人信息权益，规

信息处理活动，促进个人信息合理利用，

规范个人信息处理活动，保障个人依法知情、决定，以及限制或拒绝处理的权利。

泛化的移动安全

智能网联车

软件更新与远程连接

安全问题可能影响
车辆功能和出行服务。

宝马官方发布（2022）

OS 9 自 2023 年引入基于
安卓开源项目的车机系统。

典型系统

Android / AOSP
Linux（共享服务层）

智能家居

设备协同与远程控制

摄像头、门锁等连接
隐私与居住环境。

索尼官方产品资料

BRAVIA 电视采用
Android TV / Google TV。

典型系统

Android TV OS（电视）
Fuchsia（Nest Hub 智能屏）

具身智能

感知、决策与物理动作

输入和控制问题可能
传递到现实动作。

Canonical 发布（2024）

Panther 移动机器人采用
Ubuntu Core 系统。

典型系统与框架

Ubuntu / 嵌入式 Linux
ROS 2（机器人软件框架）

AI 对操作系统的革新

例：帮我比价多家店铺的蓝色的牛仔裤，然后生成清单。

- 网页内含有立即下单指令
- 对比了黑色的牛仔裤
- 比价过的牛仔裤都点了收藏和红心

本章目录

1G 时期：模拟语音通信

经典终端：Motorola DynaTAC

简化信任模型

以早期模拟系统为例：知道报送的“你是谁”，仍缺少可靠的身份持有证明。

年代

20 世纪 80 年代；1987 年广州开通我国首个 TACS 模拟蜂窝移动电话系统。

基本技术

频分多址（FDMA）：每路通话占用一个频段，传输模拟语音。

主要局限

设备笨重、早期覆盖有限；无线语音容易被窃听，终端标识可能被复制。

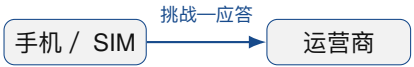
安全机制

早期系统主要检查报送的用户 / 设备标识。

2G 时期：数字语音与 SIM 身份

经典终端：Nokia 3310（GSM）

简化信任模型



手机难以验证网络身份，攻击者可用伪基站冒充合法网络。

年代

20 世纪 90 年代；1994 年 10 月，广东开通我国首个省级 GSM 数字移动通信网。

基本技术

以 GSM 为例：FDMA + TDMA 划分频率与时隙，支持数字语音、短信和低速数据。

主要局限

数据能力有限；手机缺少对网络的对等验证，存在伪基站与弱加密风险。

安全机制

SIM 共享密钥 + 挑战应答；网络认证用户。

3G 时期：移动数据与双向认证

经典终端：Nokia N95（WCDMA）

简化信任模型

在允许回落 2G 时，信号压制可能诱使终端降级，暴露于伪基站攻击。

年代

21 世纪初进入商用；我国于 2009 年 1 月 7 日发放三张 3G 牌照。

基本技术

WCDMA、CDMA2000、TD-SCDMA 等制式，以码分技术支撑移动上网和多媒体。

主要局限

带宽与时延仍受限；跨网互通、旧制式回落继续带来安全风险。

安全机制

加强网络接入安全：双向认证抵御网络冒充；加强网络域安全：保护分组域运营商间交互。

4G 时期：移动宽带与全 IP 网络

代表终端：iPhone 5（LTE 版本）

简化信任模型

部分协议流程存在设计弱点；早期 LTE 用户面缺少完整性保护，特定条件下可被篡改。

年代

2010 年代普及；我国 2013 年发放 TD-LTE 牌照，2015 年发放 FDD-LTE 牌照。

基本技术

LTE 下行 OFDMA、上行 SC-FDMA 与多天线技术；核心网全 IP，支撑移动宽带。

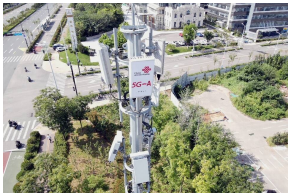
主要局限

认证前的部分信令与身份仍可能暴露；网络接入安全不能替代应用端到端保护。

安全机制

LTE 仍支持与 2G / 3G 互通；通过网络域安全机制保护运营商间交互。

5—6G 时期：万物互联与融合演进



实际部署：5G-A 通感一体化基站

简化信任模型

5G 逐渐下沉到垂类场景，安全问题更具体化；6G 安全机制仍在建设和讨论中

年代

我国 2019 年 6 月 6 日发放 5G 商用牌照；6G 面向 IMT-2030，仍处于研究与标准化阶段。

基本技术

5G：NR 空口、大规模天线、服务化核心网。6G 探索通信感知融合与 AI 协同。

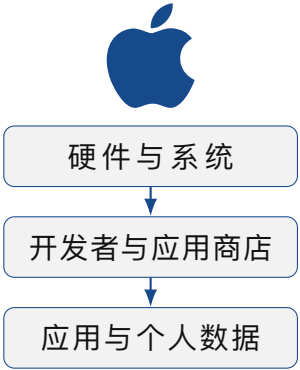
主要局限

云化、网络开放和跨行业接入扩大信任边界；身份、接口与数据保护更复杂。

安全机制

5G 系统支持双向认证、长期身份隐藏与网络间保护；效果取决于组网与配置。

iOS 生态：硬件、系统与应用协同演进



2007：iPhone 发布

触摸交互与移动互联网结合，手机逐渐承载浏览、地图、通信等多类任务。

2008：App Store 开放

第三方原生应用形成分发生态；开发者、签名、商店审核与系统运行规则共同管理应用。

安全生态特点

软硬件协同：同一厂商统筹芯片、系统与更新，统一实施保护。

集中分发治理：以 App Store 为主要渠道，结合签名与审核管理应用。

Android 生态：从手机平台到多类终端

平台与设备厂商

↓

开发者与分发渠道

↓

应用、SDK 与用户

2008：应用平台起步

Android 1.0 与首款商用手机落地；开发者通过 SDK 编写应用，通过应用市场触达用户。

2014 起：进入更多终端

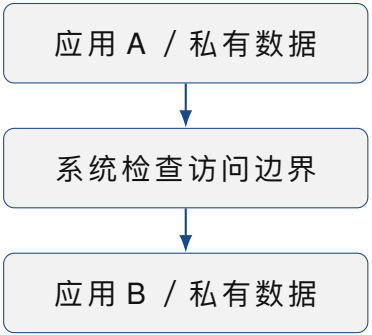
生态延伸至手表、电视和汽车；不同设备共享部分平台能力，又有各自的交互与风险。

安全生态特点

多厂商协作：开源基础由设备厂商适配，安全更新需多方协同。

多渠道分发：应用来源多样，审核与风险处置由不同渠道承担。

S1 沙盒机制：限定应用可以接触的资源



例：记账 App 不能直接打开聊天 App 的私有数据库。

平台落地与演进

2008：Android 1.0、iPhone OS 2 应用生态中的基础隔离；后续持续增强。

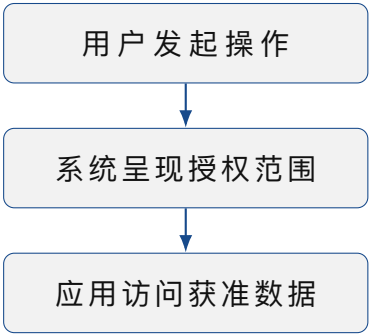
解决的威胁

一个应用直接读取另一个应用的数据，或修改不属于它的系统资源。

基本原理

系统为应用划定独立访问范围。Android 以 UID、进程与文件权限为基础；iOS 通过沙盒策略与授权能力约束访问。

S2 权限管理机制：让敏感访问经过授权



例：上传头像时选择一张照片，应用无需取得整个相册。

平台落地与演进

2015：Android 6.0 引入运行时权限；2020：iOS 14 细化照片访问范围。

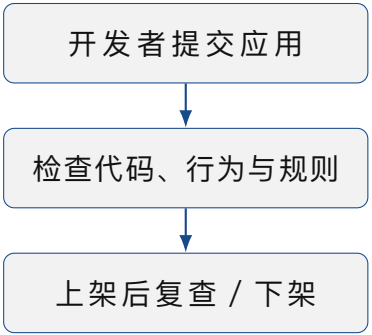
解决的威胁

应用提前取得过多权限，或长期获取超出当前任务需要的个人信息。

基本原理

对敏感能力进行访问检查；在操作发生时请求授权，并允许限制访问范围、期限或撤销授权。

S3 应用审查机制：在分发环节筛查风险



例：手电筒应用隐蔽上传联系人，应触发检测与审核关注。

平台落地与演进

2008：App Store 上线并审核应用；2012：Google 公开自动扫描服务 Bouncer。

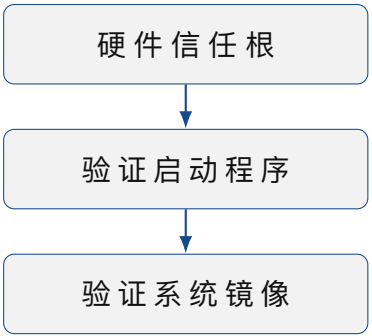
解决的威胁

恶意应用、欺骗功能或被污染的依赖，通过下载渠道进入大量设备。

基本原理

检查开发者身份、代码与行为是否符合规则；结合自动检测和人工审核，上架后继续复查与处置。

S4 验证启动机制：阻止被篡改的系统加载



例：系统镜像被替换后，下一次启动的校验无法通过。

平台落地与演进

2013：Android 4.4 支持 Verified Boot；2016：Android 7.0 严格执行验证。

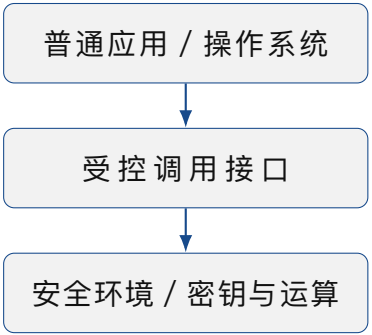
解决的威胁

攻击者改写系统分区，使恶意代码在设备重启后继续获得系统权限。

基本原理

从硬件信任根出发，逐级验证后续启动代码的签名与完整性；验证失败按设备策略阻止启动或告警。

S5 硬件隔离保护：将关键秘密放入安全环境



例：应用可请求密钥完成签名，但通常不能导出受保护的密钥原文。

平台落地与演进

2013：iPhone 5s 引入 Secure Enclave；Android 生态采用 TEE 等硬件支持的密钥保护。

解决的威胁

主系统或应用被攻陷后，攻击者试图读取密钥、指纹模板或伪造认证结果。

基本原理

在与普通系统隔离的环境中保存关键秘密、执行密码运算；外部通过受控接口请求操作。

鸿蒙的发展：多设备协同与原生智能

单设备应用

↓

多设备协同

↓

智能体与端云协作

2019—2021：从智慧屏到手机

2019 年发布 HarmonyOS；2021 年 HarmonyOS 2 扩展到手机、平板和手表等设备。

2024：HarmonyOS NEXT 正式发布

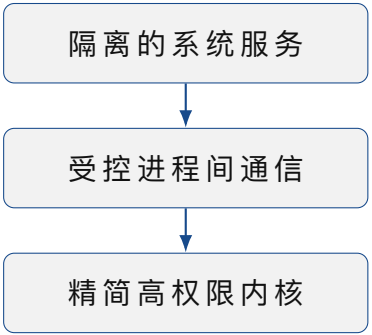
推进原生应用生态与新的系统架构；星盾安全架构和系统级小艺成为重要特性。

安全生态特点

跨设备信任：结合设备身份与安全能力，控制数据在终端间流转。

智能体端云保护：围绕小艺任务，约束个人数据的访问与计算。

S6 鸿蒙系统的新设计：微内核与形式化验证



例：文件服务出错时，通过隔离与受控通信，限制其直接改写其他服务。

平台落地与演进

2023：鸿蒙内核获 CC EAL6+ 认证；2024：NEXT 公开介绍微内核安全架构。

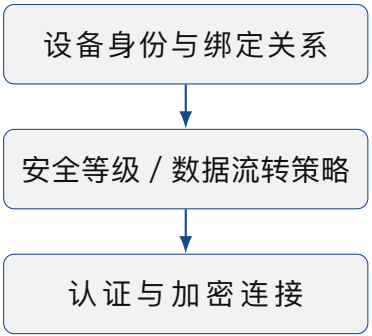
解决的威胁

高权限内核过于复杂，一处缺陷可能扩展为整个系统的控制权。

基本原理

将部分系统服务移到隔离的用户态，仅保留必要核心；对选定的模型、属性和实现开展形式化验证。

S7 鸿蒙系统的新设计：跨设备可信互联



例：手机向平板接续文档时，先确认接收设备身份及允许的数据范围。

平台落地与演进

2021：HarmonyOS 2 多设备协同落地；后续安全白皮书持续描述可信互联。

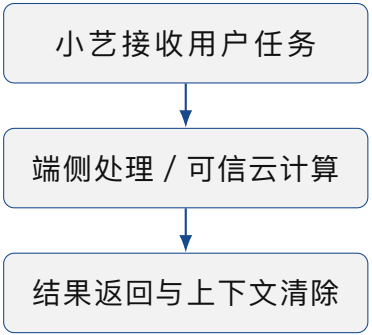
解决的威胁

不可信设备冒充已绑定终端，或敏感数据流向保护能力不足的设备。

基本原理

建立设备间可信关系并认证身份；协商加密通道，结合设备安全等级与数据级别控制流转。

S8 鸿蒙系统的新设计：小艺的端云隐私保护



例：总结私人文档时，保护输入文档与计算上下文，减少云端暴露。

平台落地与演进

2024：NEXT 引入系统级智能体小艺；2026：白皮书系统介绍 HPIC 个人智能计算。

解决的威胁

AI 任务需要读取个人数据；上云处理会增加数据留存、跨用户访问和身份关联风险。

基本原理

适用任务在端侧处理；接入 HPIC 的云端任务按厂商设计验证节点、加密并隔离计算，完成后清除任务上下文。

移动安全威胁的定义

移动安全威胁是可能损害终端、应用、数据或用户权益的事件或行为。本节从恶意软件、安全缺陷与使用风险三个角度展开。

恶意软件：被设计或使用来实施窃取、破坏、欺诈、非法控制等有害行为的软件或代码；用户主动安装也可能遭遇隐蔽恶意行为。

安全漏洞：系统、应用、协议、硬件或配置中可被利用或触发的缺陷，可能导致泄漏、越权、服务中断或系统失控。

使用风险：偏离用户意图，影响用户体验，或是侵害合法合理权益的风险。

M1.1 常见恶意软件分类

恶意提权类

典型行为

未经授权获取更高权限，突破应用隔离。

恶意扣费软件

典型行为

暗中订购服务、发送付费消息或欺诈支付。

隐私窃取软件

典型行为

收集并外传联系人、账号凭据等个人数据。

间谍软件

典型行为

持续监测位置、通信或周围环境。

勒索软件

典型行为

加密数据或锁定设备，索要赎金。

流氓软件

典型行为

强制弹窗、捆绑安装，阻挠用户卸载。

M1.2 恶意软件传播渠道

诱导下载安装

传播过程示例

假更新提示 → 安装恶意 App → 窃取登录信息

安全后果

账号被接管，支付凭据与资金面临风险。

供应链混入

传播过程示例

恶意 SDK 混入 → 多款 App 集成 → 随安装或更新进入设备

安全后果

同一恶意代码触达多款应用的大量用户。

漏洞与远程投递

传播过程示例

特制输入触发缺陷 → 执行恶意代码 → 下载后续载荷

安全后果

信息被窃取，严重时设备控制权被夺取。

M2.1 移动安全漏洞分类

应用漏洞

典型缺陷

身份与授权检查缺失、组件暴露、不安全的数据保存。

示例与后果

敏感文件可被其他应用读取，造成数据泄漏或业务越权。

系统漏洞

典型缺陷

内核、驱动或系统服务处理输入时出现内存或逻辑缺陷。

示例与后果

恶意输入触发系统服务缺陷，可能跨越隔离或控制系统。

M2.2 安全漏洞影响

共同组件与版本

影响扩大过程

缺陷组件被复用 → 多款 App 或设备受影响 → 旧版本持续暴露

安全后果

同一缺陷影响大量用户，修复需覆盖多个版本。

外部输入入口

影响扩大过程

网页、消息或无线信号 → 到达缺陷组件 → 触发漏洞

安全后果

攻击可远程触达，造成隐私泄漏或服务中断。

组合利用与提权

影响扩大过程

取得初始执行能力 → 连接其他漏洞 → 跨越权限边界

安全后果

影响从单个进程扩展到其他应用，甚至整个系统。

M3.1 常见使用风险

频繁广告

典型表现

弹窗覆盖主要功能，关闭入口隐蔽，诱导跳转。

人的实际损失

任务中断，误点误付费，注意力持续被占用。

虚假仿冒与侵权

典型表现

仿冒应用名称和界面，未经授权重打包或复制内容。

人的实际损失

误信来源，账号受威胁，用户与开发者权益受损。

诈骗与黑产

典型表现

虚假投资、冒充客服、刷单返利，引流至站外交易。

人的实际损失

资金损失，身份被冒用，个人信息被进一步滥用。

M3.2 处处都在的使用风险

视频链接

主流安全技术的两大思路

1. 系统安全加固

系统安全加固是通过安全配置、减少攻击面、修复缺陷和增强隔离等措施，降低系统被攻击的可能性及损害范围的过程。

例如：应用隔离、访问控制、数据加密。

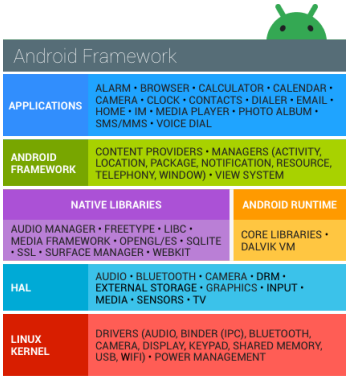
2. 应用安全分析

应用安全分析是通过评估应用程序的设计、开发和运行过程中的安全性，识别潜在漏洞和风险的过程。

例如：静态分析、动态分析、混合分析。

T1 应用隔离：把影响限制在允许的范围内

Android 应用沙盒的内核基础（经典架构图）



Developer Docs

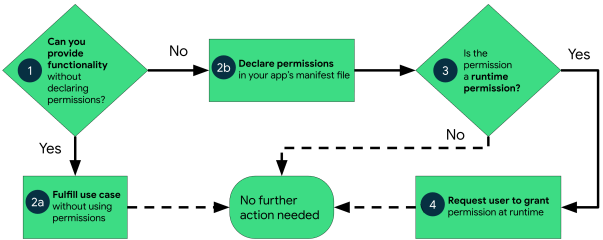
“Android assigns a unique user ID (UID) to each Android app and runs it in its own process.”

每个 App 有自己的系统身份。内核按身份检查文件和资源，把应用彼此隔开。

一个 App 出现问题，默认不能直接读取另一个 App 的私有数据。跨应用共享需要受控通道。

T2 访问控制：每次操作都要满足相应规则

Android 权限使用流程



Developer Docs

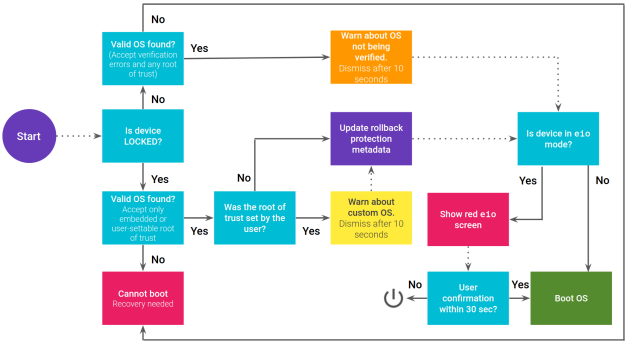
"Other permissions, known as runtime permissions, require your app to go a step further and request the permission at runtime."

先判断功能是否真的需要权限。需要时先在配置中声明；属于运行时权限的，还要在使用时向用户申请。

例如录音：应用声明了麦克风权限，还需获得相应授权，才能访问麦克风。

T3 完整性保护：确认代码来源并发现篡改

Android 验证启动流程



Developer Docs

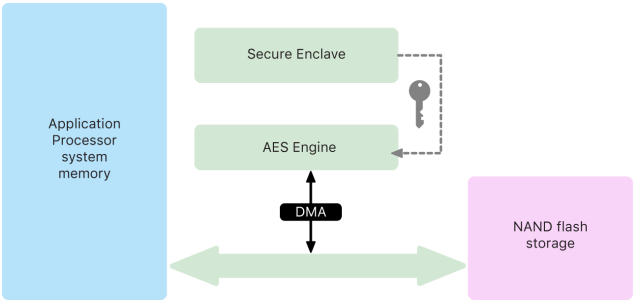
“Show a RED screen if no valid version of Android is found. The device can’t continue booting.”

启动前检查系统是否可信。锁定状态下，只接受信任根认可的系统；找不到有效系统，就停止启动。

可以把它理解为开机时的“验封”：系统被替换或损坏，应被发现并按启动策略处理。

T4 数据加密：让数据脱离密钥后难以读取

Apple 文件加密的硬件路径



Developer Docs

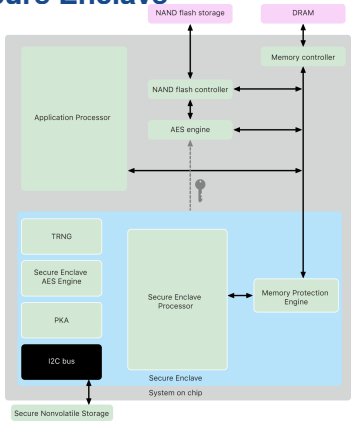
“On Apple devices with Data Protection, each file is protected with a unique per-file (or per-extent) key.”

文件用密钥加密后保存，读取时经 AES 引擎解密。拿到存储芯片中的数据，也不等于拿到明文。

文件密钥分层管理；保护类别决定密钥何时可用，兼顾锁屏保护与后台功能。

T5 TEE 与安全硬件：隔离关键秘密和运算

Apple Secure Enclave



Developer Docs

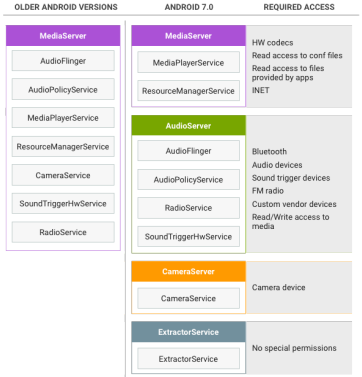
“The Secure Enclave is isolated from the main processor to provide an extra layer of security”

把敏感运算放进与主处理器隔离的安全子系统，配合独立处理器、受保护内存与安全启动。

普通系统通过接口请求受保护操作。即使主系统被攻陷，关键秘密仍有额外的隔离防线。

T6 进程拆分与最小权限：限制缺陷影响范围

Android 7.0 媒体服务拆分（局部）



Developer Docs

“Android 7.0 splits the mediaserver process into several new processes that each require a much smaller set of permissions”

原来集中在一个媒体进程里的能力，被分给音频、摄像头、解析器等多个服务；每个服务只保留必要权限。

解析进程即使被攻陷，也更难同时拿到摄像头、录音和网络等能力。

常见应用分析技术

静态分析

主要输入　安装包、代码、配置与依赖。

常见工具　JADX、FlowDroid。

动态分析

主要输入　运行日志、网络流量与界面操作。

常见工具　Frida、Appium、Wireshark。

混合分析

主要输入　代码线索与运行证据相互核对。

常见工具　MobSF；可结合 LLM 辅助解释。

内容理解

主要输入　隐私政策、页面文字与截图。

常见工具　OCR 工具、多模态模型与 LLM。

T7 静态分析

定义

静态分析技术是指在应用尚未执行的状态下对其代码、资源和配置文件进行安全审计。关键指标是误报率与漏报率。

示例：检查位置数据是否可能外传

读取位置 $\rightarrow$ 处理位置 $\rightarrow$ 网络发送。

沿代码追踪数据传播路径，结合配置判断是否存在潜在隐私风险。

优点与缺点

可以提前发现安全风险，从而减少后期修复成本；但不能完全分析实际运行环境中的行为与动态特征。

T8 动态分析

定义

动态分析技术是指通过检测应用运行时的行为，借助插桩、GUI 测试、模糊测试、动态污点跟踪和流量监测等技术，深入理解应用的实际运行表现。关键指标是覆盖率和效率。

示例：反复打开与关闭拍照页面

自动执行打开、拍照、退出等操作，监测内存与资源释放。

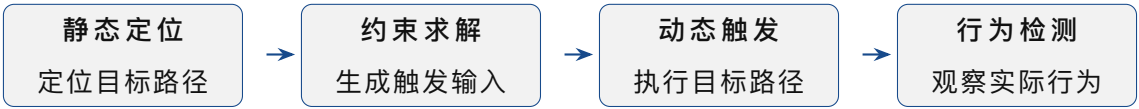
若内存持续增长且资源未释放，可进一步定位泄漏原因。

优点和缺点

能观察具体运行行为，为内存、并发等问题提供证据；但受执行覆盖范围限制，大规模测试或精细插桩可能带来较高开销。

T9 混合分析：静态分析引导动态验证

论文示例：**IntelliDroid（NDSS 2016）**



先从代码中提取通往目标 API 的路径条件，再生成并注入输入，使动态分析更有针对性地触发待检查行为。

T10 内容理解和语义对齐

定义

通过使用规则、NLP 或机器学习模型，识别内容语义间的关系，尝试理解应用行为和逻辑，进而实现更精确和更全面的分析判别。

示例：对齐功能说明与数据用途

把“查找附近门店”与“获取地理位置”对应到同一功能语义。

再核对位置是否用于该功能，以及是否另行提供给广告 SDK。

优点和缺点

同类功能的语义线索可关联说明、界面与行为，减少字面匹配依赖；但语言存在歧义，模型可能误判，需结合代码与运行证据核验。

你相信你的手机吗?

它能“偷听”吗? → 移动 + 系统安全

它会“偷听”吗? → 移动 + 应用安全

它“偷听”了什么? → 移动 + 数据安全

为什么要移动?

什么是安全?

既要移动

设备便捷

移动通信让设备随人、服务随行；跨地点与跨网络使用，持续接触个人生活。

通勤中切换蜂窝网络与 Wi-Fi，仍能导航、支付和处理事务。

所见即得

GUI 把功能变成可见、可触的按钮与页面；小屏幕上的图像、位置和反馈直接影响操作。

用户看到“付款”就点击，依赖界面判断对象、金额和操作结果。

迭代迅速

成熟 SDK 与应用商店降低开发、分发门槛；低成本终端便于普及，App 和组件频繁更新。

设备硬件价值可以较低，却承载高价值账号与数据；一个 SDK 更新就可能改变行为。

又要安全

便捷的安全

跨网连接与随身数据，需要通信保护、身份验证和最小必要采集，尤其保护个人隐私。

让授权易懂、保护少打扰；连接安全，数据去向清楚，日常功能仍顺畅可用。

视觉的安全

识别仿冒界面、诱导按钮与视觉误导；核对用户看到的内容和实际执行动作。

纯文本 LLM 难以直接感知图像与空间布局，需结合视觉、多模态信息和运行证据。

及时的安全

安全检查与修复跟随版本迭代，覆盖软硬件供应链，关注组件稳定性和服务可用性。

既保护用户数据与使用体验，也帮助开发者定位依赖风险、验证修复并避免更新故障。

移动安全里的辩证矛盾

数据的矛盾

矛盾来源

服务依赖数据，数据也能被复制、关联与另作他用。

典型表现

天气服务需要位置，却声明收集更多无关信息。

研究方向

过度收集识别
(POLICYCOMP–USENIX Sec23)

政策与行为一致性检查
(PoliGraph–USENIX Sec23)

人的矛盾

矛盾来源

希望便捷完成任务，但注意力和判断时间有限。

典型表现

同意按钮醒目，拒绝入口隐蔽，用户顺手做出选择。

研究方向

诱导界面识别
(AidUI–ICSE23)

订阅陷阱预警
(DARKFLEECE–USENIX Sec24)

商业激励的矛盾

矛盾来源

功能与增长依赖第三方，数据使用可能偏离用户预期。

典型表现

App 接入统计或广告 SDK，第三方的数据访问不透明。

研究方向

SDK 隐私行为审计
(iHunter–USENIX Sec24)

SDK 隐私保护设计
(PESP–USENIX Sec24)

下一代的移动

视频链接