权限引导在手机卫士中的实践

团队：手机卫士核心安全
讲师：庞洲

360
www.360.cn

团队/个人介绍

核心安全团队：

1. 负责开发维护手机卫士安全业务
2. 业务包含骚扰拦截、杀毒业务、支付保镖、反诈骗相关
3. 负责安全相关新业务的调研分析

个人负责业务：

1. 权限引导业务，提升APP活跃和权限开启率
2. 支付安全相关业务开发
3. 病毒、恶意APP逆向分析
4. 安全课题研究分析

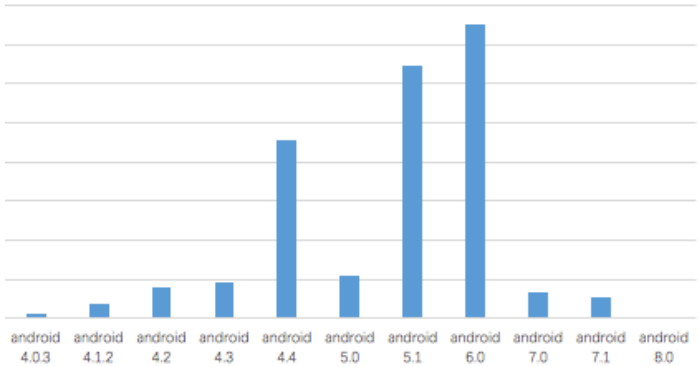
课程提纲/内容

- 为什么要做权限引导
- ROM适配技巧
- 权限管理逻辑
- 自动化监控

为什么要做权限引导

GOOGLE：

1. 漏洞越来越少，Root获取困难
2. 高版本系统有限制APP长期存活的趋势

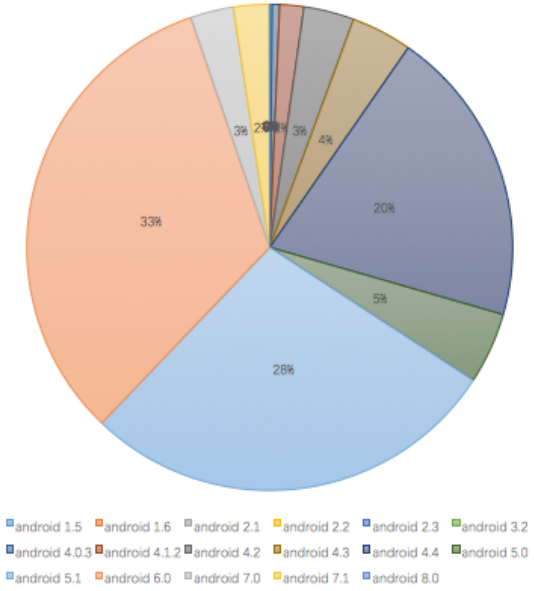


主流厂商：

1. OS更新跟进及时
2. Rom版本逐渐升高，引入权限管理

权限管理前后对比：

1. manifest声明 - 动态授权
2. 运行期间不受影响 - 用户点选
3. 存活不受影响 - 后台运行限制、自启限制
4. 拉活手段多种花样 - 关联启动限制、自启限制



Rom适配过程

什么是ROM ：

1. 内核
2. Framework
3. System service
4. UI
5. 内置APP

Rom适配过程

如何区分不同版本ROM、不同厂家ROM ：

1. 读取厂商信息
2. 读取Rom信息 - getprop
3. 版本号界定与适配范围 - 3.x 4.x
4. 主流厂商版本号判断方式 -下划线 空格
5. 为什么适配不按机型 - 刷Rom 升级Rom

Rom适配过程

适配需要做哪些事情：

1. 看，找到所有权限所在的界面

   - dumpsys命令

2. 扒，提取权限管理相关app

```
sagit:/system/framework $ ll
total 20760
-rw-r--r-- 1 root root    6482 2009-01-01 00:00 ConnectivityExt.jar
-rw-r--r-- 1 root root     318 2009-01-01 00:00 QPerformance.jar
-rw-r--r-- 1 root root     318 2009-01-01 00:00 QtiTelephonyServicelibrary.jar
-rw-r--r-- 1 root root     318 2009-01-01 00:00 WfdCommon.jar
-rw-r--r-- 1 root root    2014 2009-01-01 00:00 activation.jar
-rw-r--r-- 1 root root     318 2009-01-01 00:00 am.jar
-rw-r--r-- 1 root root     126 2009-01-01 00:00 android-support-v13.jar
-rw-r--r-- 1 root root     126 2009-01-01 00:00 android-support-v7-recyclerview.jar
-rw-r--r-- 1 root root     318 2009-01-01 00:00 android.test.runner.jar
-rw-r--r-- 1 root root   17632 2009-01-01 00:00 apache-xml.jar
-rw-r--r-- 1 root root     318 2009-01-01 00:00 appwidget.jar
drwxr-xr-x 2 root root    4096 2009-01-01 00:00 arm
drwxr-xr-x 2 root root    4096 2009-01-01 00:00 arm64
```

3. 分析，逆向分析app代码逻辑，定位权限状态存储

Rom代码分析技巧

没有Root：

1. /data/data/等路径不能访问
2. 不能随便修改系统属性
3. 不能部署xposed module测试

odex处理：

1. oat2dex
2. 提取/system/framework/路径文件
3. 转换得到dex

常用路径及文件抓取方式

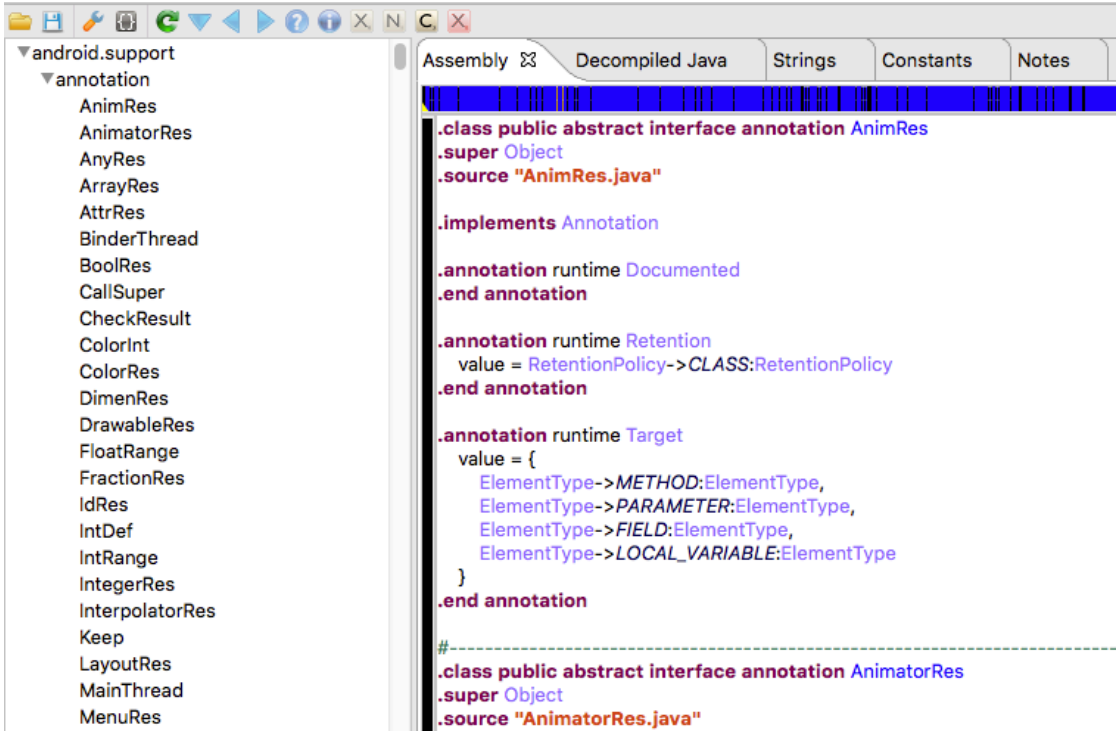
1. /system/app/
2. /system/priv-app/
3. adb pull(skipped)

Rom代码分析技巧

分析工具：

1. jeb
2. apktool

   加强版



代码定位：

1. dump界面
2. 按钮点击/checkbox/switch，framework-res
3. provider/db
4. System service

Rom代码分析技巧

代码定位：

静态分析

1. 文案 + 图片

```
<public type="string" name="overdueDetail" id="0x7f080221" />
<public type="string" name="statistics" id="0x7f080222" />
<public type="string" name="newInvest" id="0x7f080223" />
<public type="string" name="myInvestList" id="0x7f080224" />
```

out/res/public.xml

2. 方法

```
.method public final synthetic a(a)Object
    .registers 6
```

3. 接口

```
.class final ad
.super u
```

```
.class final an
.super Object
.source ""

.implements v
```

4. aidl

跨dex用grep搜索

5. 混淆的处理

自下而上rename

代码定位：

动态分析

1. xposed module

   -> func_orig() ->

   -> func_before() -> func_orig() -> func_after() ->

2. jdb断点

   常用系统关键API

   stop in android.content.ContextWrapper.startActivity(android.content.Intent)

   stop in android.os.Handler.sendEmptyMessage(int)

   获取调用栈

3. 重打包

   修改log开关

   执行流程

Rom代码分析技巧

查询权限：

1. content query
    - 程序实现了provider
    - 没有特殊权限限制
    - 位运算规则
2. xml/pref/txt
3. 反射系统服务

特殊情况处理：

1. 文件夹名字，注意大小写排列组合
2. 代码模糊搜索 grep -nr

权限管理业务

接口设计：

1. queryAuthStatus() - 权限状态
2. queryLastGuideTimePair() - 引导频次
3. setAuthStatus() - 权限状态自治
4. isRomAdapted() - 是否适配
5. startAuthGuide() - 跳转

权限状态维护：

1. 准确判断
2. 不准确判断
   - 业务自检
   - 业务上报

数值	重启次数	运行时长	状态设置	
0	0	0	重启少 时长短	后台关 自启关
1	0	1	重启少 时长长	后台开 自启开
2	1	0	重启多 时长短	后台关 自启开
3	1	1	重启多 时长长	后台开 自启关

辅助功能

辅助功能特性：

1. 优点
    - 模拟点击
    - 自动化完成所有流程
2. 缺点
    - 存活不稳定
    - 部分Rom屏蔽或针对单个APP屏蔽

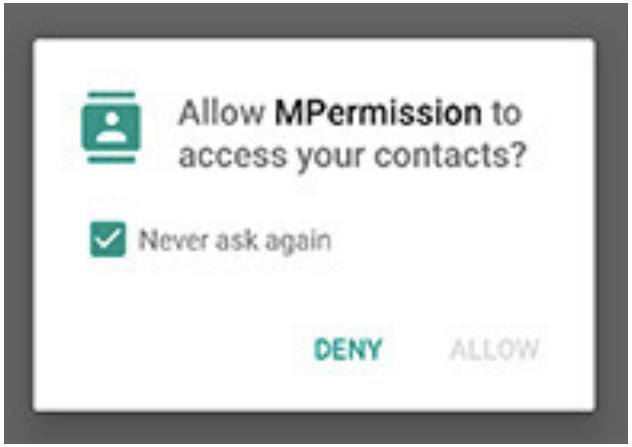
适配方式：

1. Rom
2. 模拟点击
    - 界面元素查找 - 个别ROM元素比较难找
    - 开启流程控制 - 维护状态 STEP1 STEP2 STEP3

TARGET23机制

OS与APK：

1. 系统适用范围 - android 6.0及以上
2. APP需要做的改动
   - target
   - 涉及权限的代码 try/catch、权限申请调用
   - 其他升级target之后的兼容性改动
3. OS对APP版本的处理策略



OS	APP	STATUS
22	22	默认允许
22	23	默认允许
23	22	默认允许
23	23	动态申请

TARGET23机制

权限分组：

1. Normal权限
2. Dangers权限

Permission Group	Permissions
android.permission-group.CALENDAR	• android.permission.READ_CALENDAR • android.permission.WRITE_CALENDAR
android.permission-group.CAMERA	• android.permission.CAMERA
android.permission-group.CONTACTS	• android.permission.READ_CONTACTS • android.permission.WRITE_CONTACTS • android.permission.GET_ACCOUNTS
android.permission-group.LOCATION	• android.permission.ACCESS_FINE_LOCATION • android.permission.ACCESS_COARSE_LOCATION

3. OS分组管理机制
4. 特殊权限
    - WRITE_SETTINGS
    - SYSTEM_ALERT_WINDOW

TARGET23机制

权限机制与厂商的处理：

1. 申请流程
    - Context.checkSelfPermission(String permission)
    - Activity.requestPermissions(String[] permissions, int requestCode)
    - Activity.shouldShowRequestPermissionRationale(String permission)
    - Activity.onRequestPermissionsResult(int code,String[] perms,int[] ret)
    - Settings.System.canWrite(Context context)
    - Settings.canDrawOverlays(Context context)
2. 厂商支持情况
    - 厂商定制过的ROM是否支持target23机制，需要具体分析

TARGET23机制

引导方案：

1. 权限请求接口
    - requestAuth(int authCode)
    - isRequestAuthSupported(int authCode)
    - 透明Activity
2. 队列处理
    - 入队缓存
    - 分组去重、合并
    - 队列排序

自动化适配工具

自动化适配工具

1. 适配步骤抽象
   - 定位界面 -> 提取dex -> manifest查看export属性 -> onCreate intent extra
2. 工具设计
   - odex提取与转换
   - smali分析
   - 结果excel

```
"Landroid/content/Intent;->getStringExtra",
"Landroid/content/Intent;->getBooleanExtra",
"Landroid/content/Intent;->getByteExtra",
"Landroid/content/Intent;->getShortExtra",
"Landroid/content/Intent;->getCharExtra",
"Landroid/content/Intent;->getIntExtra",
"Landroid/content/Intent;->getLongExtra",
"Landroid/content/Intent;->getFloatExtra",
"Landroid/content/Intent;->getDoubleExtra",
```

自动化监控

设计目标：

1. 回归测试
    - 发版之前多个机型大规模回归测试
2. 适配方案复用
    - 发现“缝隙”版本号的兼容程度
3. 适配预警
    - 及时发现新版本Rom的不适配情况

自动化监控

客户端实现：

1. 输出校验标准case
2. debug命令接收测试指令并输出调用结果
3. 线上回归(插件覆盖至最新release包)

PC端实现：

1. 提取测试机信息
2. 部署测试包、部署调度工具
3. 校验权限状态、校验界面跳转
4. 收集测试结果，生成报告，发送预警邮件
5. 自动化调度，每日自动回归

自动化监控

报表统计与预警规则：

1. 跳转结果
2. 权限状态
3. 预警规则

   适配状态

   跳转是否正确

测试场景	权限 Code	action	状态	extra_info	Activity 是否匹配成功
查询该 rom 是否适配	null	action_check_rom_adapted	1	rom adapted	
发短信权限	1	action_start_guide_auth	1	guide success	success
发短信权限	1	action_check_auth_status	1		

后续计划

方案化

sdk云控升级数据文件，最大程度抽象ROM适配的通用逻辑

1. 引导页跳转
2. 权限状态查询方法
    - 系统标准，sdk23
    - ROM中特定provider
    - 反射调用特定api
3. 无需反复发版集成jar包

后续计划

360
www.360.cn

- 技术交流（干货）：
- 奇卓社（360移动技术微信公众号）

谢 谢!

安全第一