Safe Zygote - 为移动支付安全护航

夏良钊@百度移动安全实验室

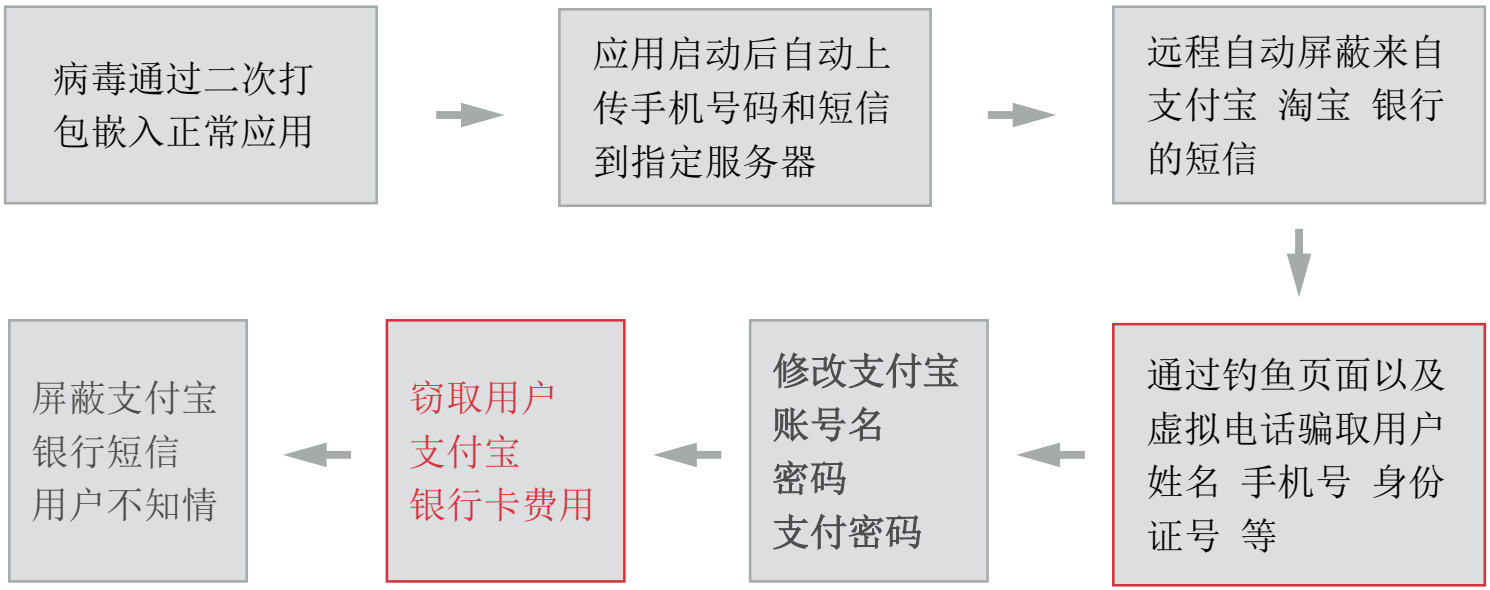
Agenda

- 当前移动支付面临的威胁
- Root or Not Root
- Safe Zygote 安全入口方案
- 4 + 1 下一代支付威胁终结者

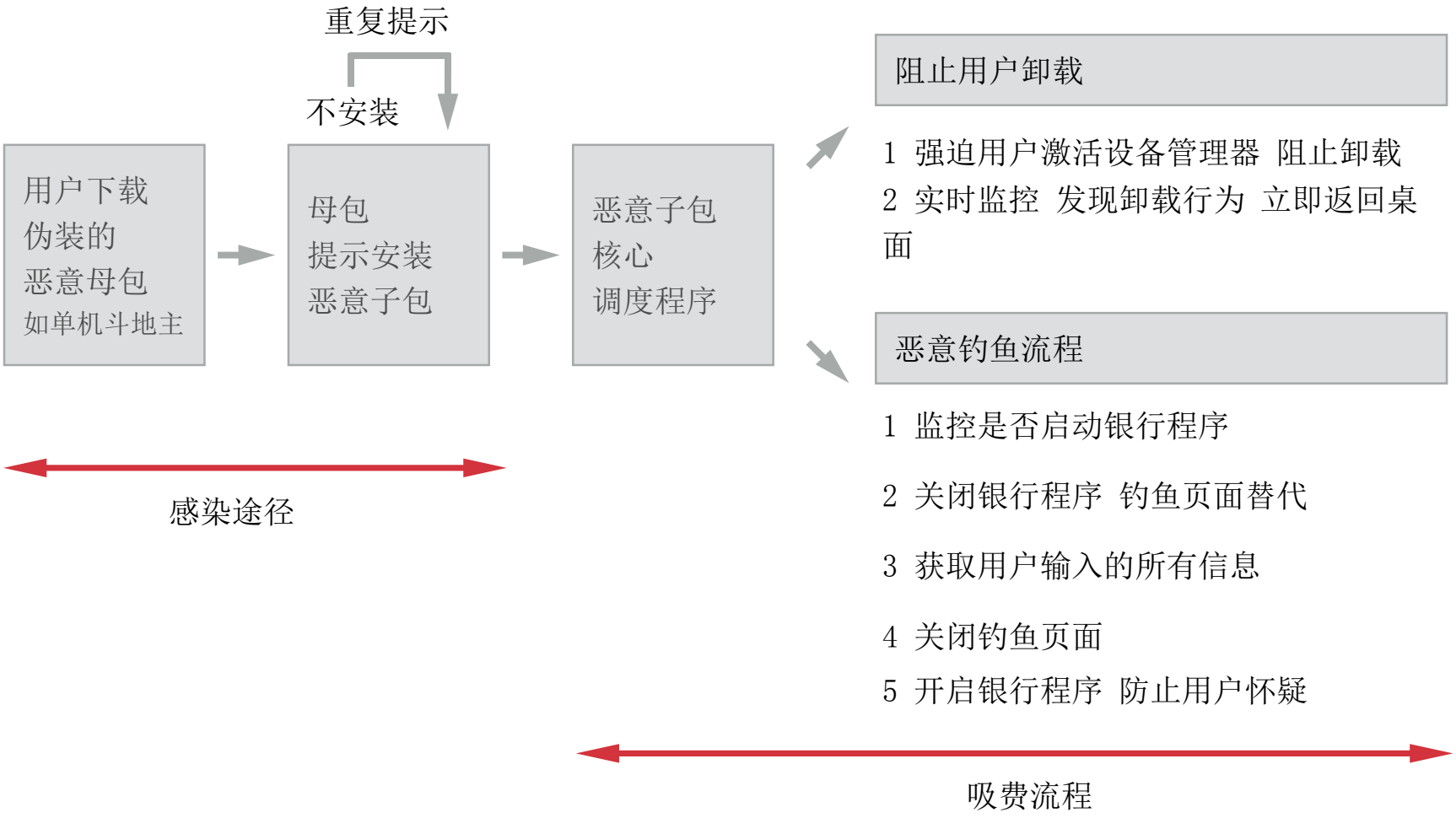
当前移动支付面临的威胁

- **“支付宝大盗”**：针对**支付宝**，骗取用户身份信息
- **“吸金幽灵”**：专门用于窃取用户**金融类账户**信息
- **“银行悍匪”**：高仿**手机银行**软件，骗取用户信息，上传到服务器，进而让用户蒙受损失
- **“微信支付大盗”**：安装**恶意扣费**软件，向好友发送欺诈短信，窃取通讯录和短信等
- **病毒伪装与山寨：**聚美优品、唯品会、淘宝特卖，10086等遭遇过

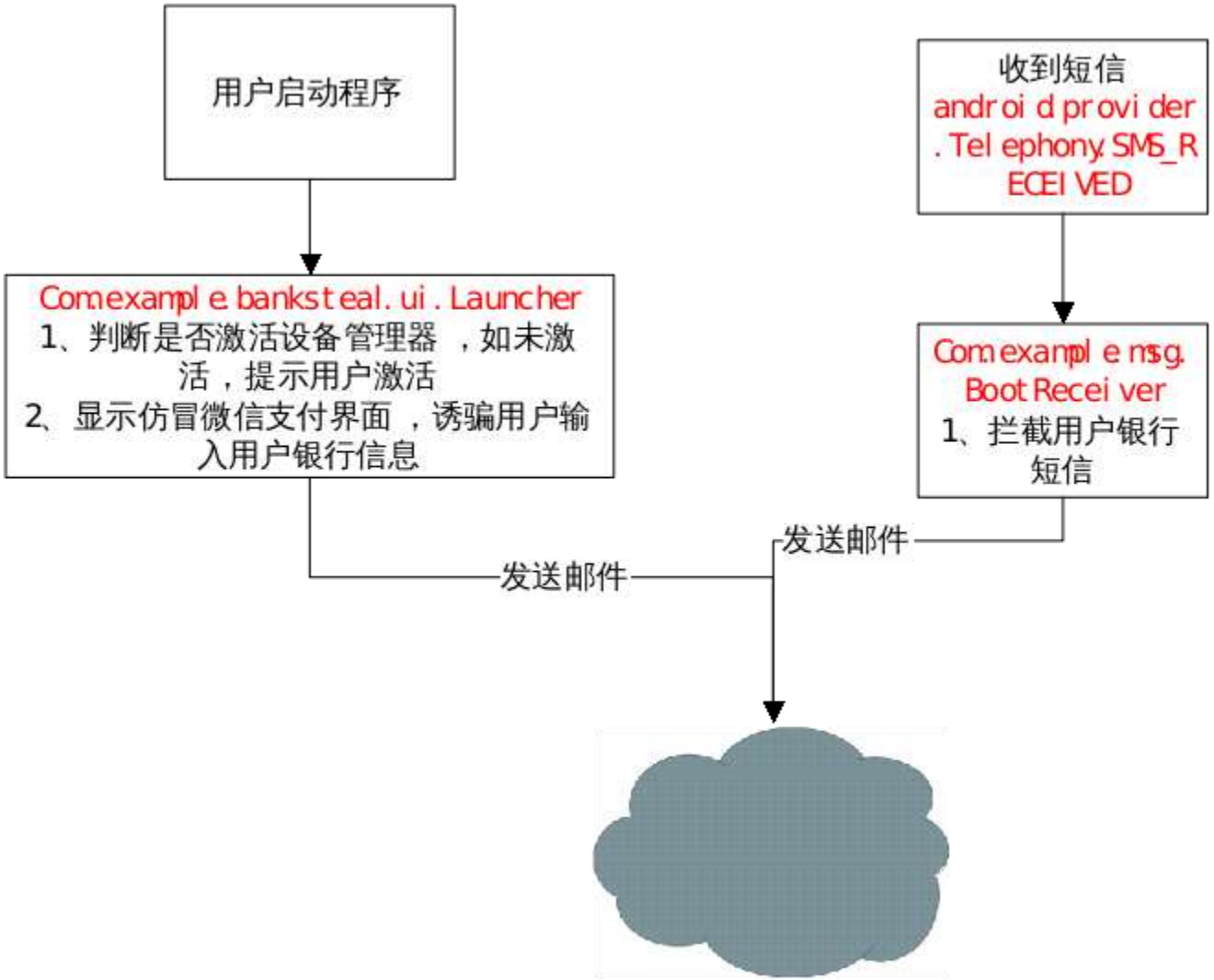
揭秘 支付宝大盗


病毒通过二次打包嵌入正常应用
应用启动后自动上传手机号码和短信到指定服务器
远程自动屏蔽来自支付宝 淘宝 银行的短信
通过钓鱼页面以及虚拟电话骗取用户姓名 手机号 身份证号 等
修改支付宝账号名 密码 支付密码
窃取用户支付宝银行卡费用
屏蔽支付宝银行短信用户不知情

揭秘 银行悍匪


重复提示
不安装
用户下载
伪装的
恶意母包
如单机斗地主
母包
提示安装
恶意子包
恶意子包
核心
调度程序
阻止用户卸载
1 强迫用户激活设备管理器 阻止卸载
2 实时监控 发现卸载行为 立即返回桌面
恶意钓鱼流程
1 监控是否启动银行程序
2 关闭银行程序 钓鱼页面替代
3 获取用户输入的所有信息
4 关闭钓鱼页面
5 开启银行程序 防止用户怀疑
感染途径
吸费流程

揭秘 微信支付大盗

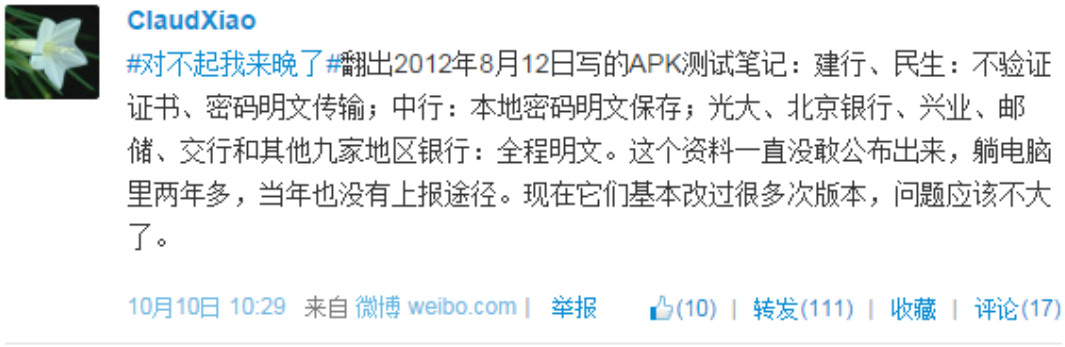

用户启动程序
收到短信
android.provider.Telephony.SMS_RECEIVED
Com.example.banksteal.ui.Launcher
1、判断是否激活设备管理器 ，如未激活，提示用户激活
2、显示仿冒微信支付界面 ，诱骗用户输入用户银行信息
Com.example.msg.BootReceiver
1、拦截用户银行短信
发送邮件
发送邮件

问题总结

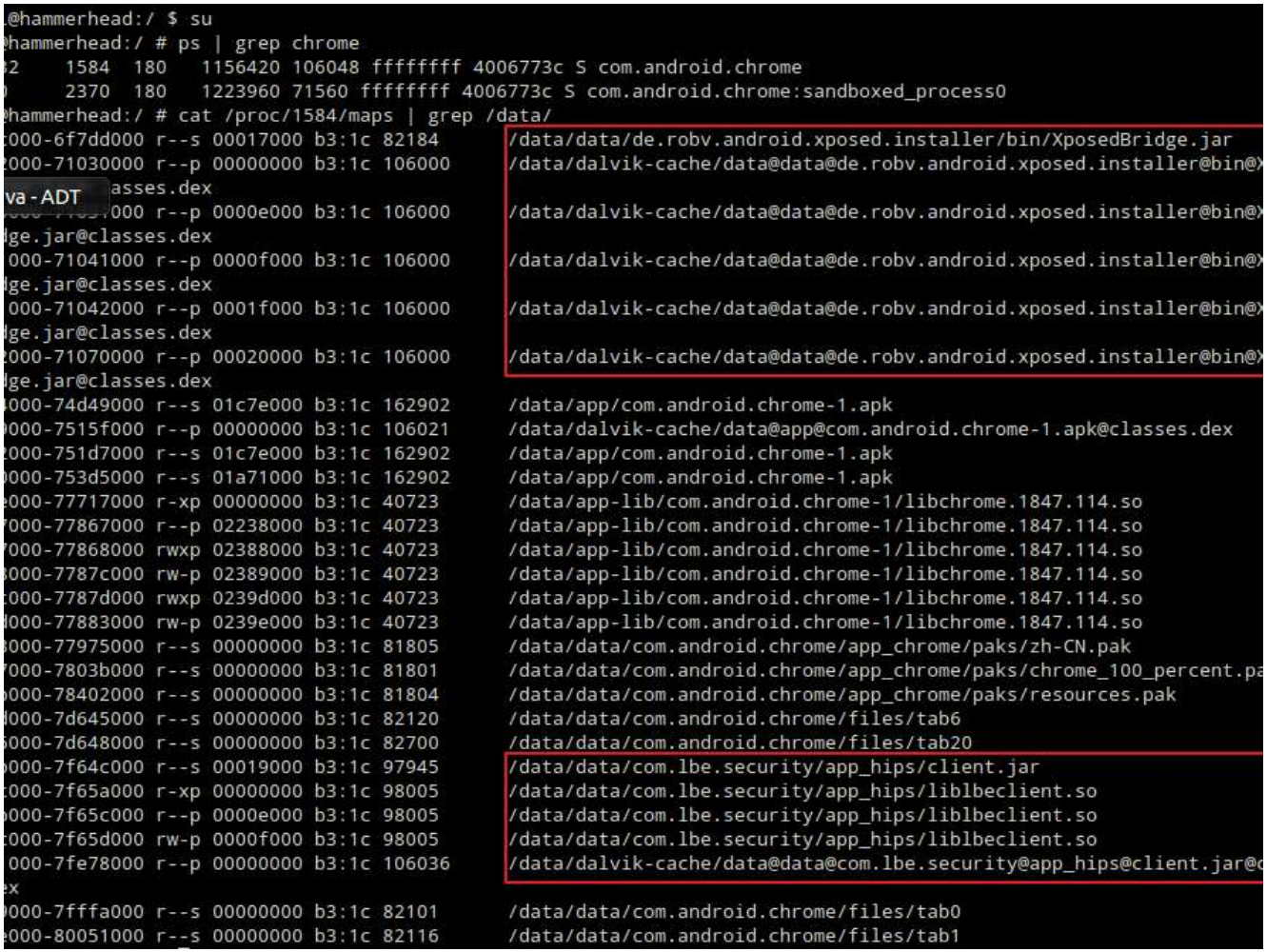
- 伪装过的手机病毒
- 正常应用重新打包
- 代码注入
- 动态跟踪

Not Root

- 系统漏洞
  - MasterKey/FakeID漏洞
  - Activity劫持钓鱼
- 应用程序漏洞
  - 组件暴露
  - WooYun-2014-53037 敏感信息明文存储



Root


@hammerhead:/ $ su
hammerhead:/ # ps | grep chrome
2 1584 180 1156420 106048 ffffffff 4006773c S com.android.chrome
2370 180 1223960 71560 ffffffff 4006773c S com.android.chrome:sandboxed_process0
hammerhead:/ # cat /proc/1584/maps | grep /data/
000-6f7dd000 r--s 00017000 b3:1c 82184 /data/data/de.robv.android.xposed.installer/bin/XposedBridge.jar
000-71030000 r--p 00000000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
va - ADT
asses.dex
000 r--p 0000e000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
dge.jar@classes.dex
000-71041000 r--p 0000f000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
dge.jar@classes.dex
000-71042000 r--p 0001f000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
dge.jar@classes.dex
000-71070000 r--p 00020000 b3:1c 106000 /data/dalvik-cache/data@data@de.robv.android.xposed.installer@bin@
dge.jar@classes.dex
000-74d49000 r--s 01c7e000 b3:1c 162902 /data/app/com.android.chrome-1.apk
000-7515f000 r--p 00000000 b3:1c 106021 /data/dalvik-cache/data@app@com.android.chrome-1.apk@classes.dex
000-751d7000 r--s 01c7e000 b3:1c 162902 /data/app/com.android.chrome-1.apk
000-753d5000 r--s 01a71000 b3:1c 162902 /data/app/com.android.chrome-1.apk
000-77717000 r-xp 00000000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
000-77867000 r--p 02238000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
000-77868000 rwxp 02388000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
000-7787c000 rw-p 02389000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
000-7787d000 rwxp 0239d000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
000-77883000 rw-p 0239e000 b3:1c 40723 /data/app-lib/com.android.chrome-1/libchrome.1847.114.so
000-77975000 r--s 00000000 b3:1c 81805 /data/data/com.android.chrome/app_chrome/paks/zh-CN.pak
000-7803b000 r--s 00000000 b3:1c 81801 /data/data/com.android.chrome/app_chrome/paks/chrome_100_percent.pa
000-78402000 r--s 00000000 b3:1c 81804 /data/data/com.android.chrome/app_chrome/paks/resources.pak
000-7d645000 r--s 00000000 b3:1c 82120 /data/data/com.android.chrome/files/tab6
000-7d648000 r--s 00000000 b3:1c 82700 /data/data/com.android.chrome/files/tab20
000-7f64c000 r--s 00019000 b3:1c 97945 /data/data/com.lbe.security/app_hips/client.jar
000-7f65a000 r-xp 00000000 b3:1c 98005 /data/data/com.lbe.security/app_hips/liblbeclient.so
000-7f65c000 r--p 0000e000 b3:1c 98005 /data/data/com.lbe.security/app_hips/liblbeclient.so
000-7f65d000 rw-p 0000f000 b3:1c 98005 /data/data/com.lbe.security/app_hips/liblbeclient.so
000-7fe78000 r--p 00000000 b3:1c 106036 /data/dalvik-cache/data@data@com.lbe.security@app_hips@client.jar@
ex
000-7fffa000 r--s 00000000 b3:1c 82101 /data/data/com.android.chrome/files/tab0
000-80051000 r--s 00000000 b3:1c 82116 /data/data/com.android.chrome/files/tab1

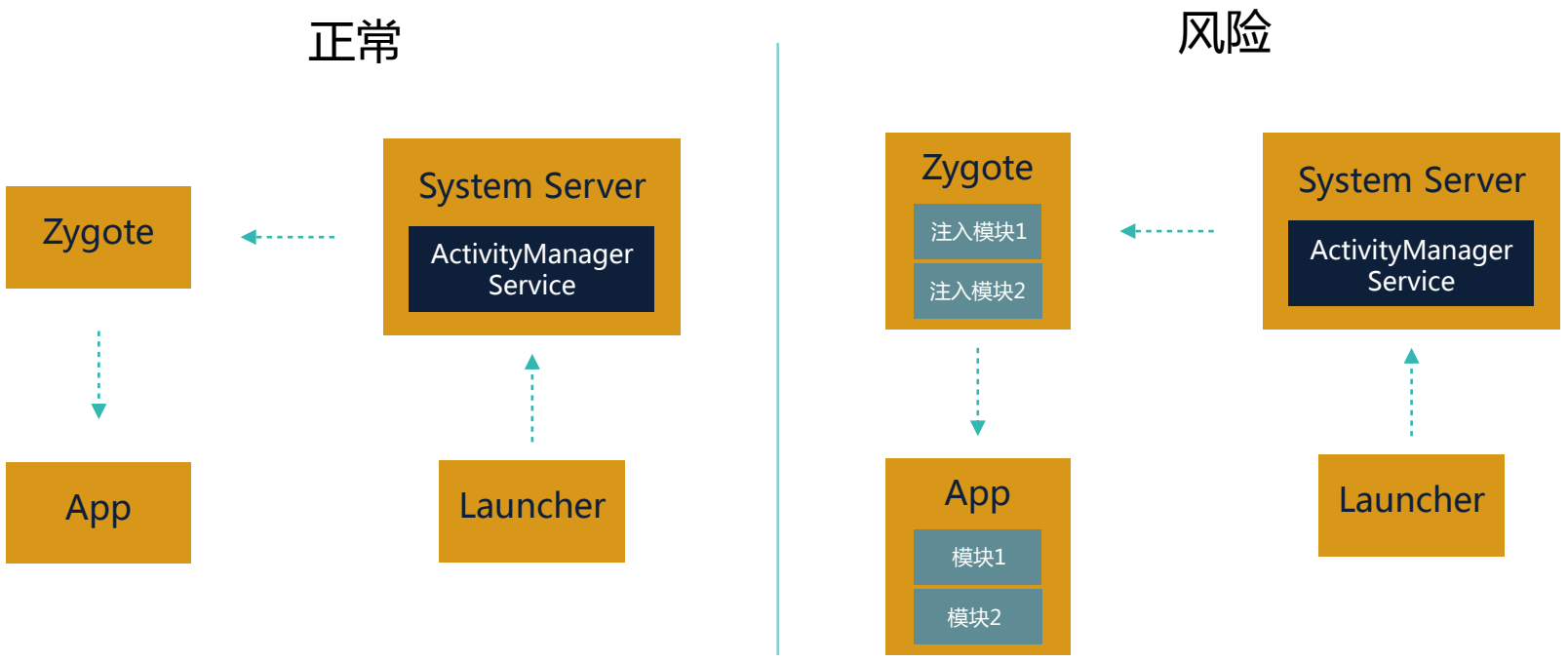
Hook

- Xposed
  - http://repo.xposed.info/
- Cydia Substrate
  - iOS/Android
  - http://www.cydiasubstrate.com/
- ADBI/DDI
  - https://github.com/crmulliner/adbi
  - https://github.com/crmulliner/ddi

Xposed – Hook

- Android平台最流行的开源hook框架
- 替换app_process
- Java函数转为Native函数
- 在被hook的函数执行时，调用BeforeHook/AfterHook接口
- 添加/删除hook需要重启zygote

运行时风险


正常
Zygote
System Server
ActivityManager Service
App
Launcher
风险
Zygote
注入模块1
注入模块2
System Server
ActivityManager Service
App
模块1
模块2
Launcher

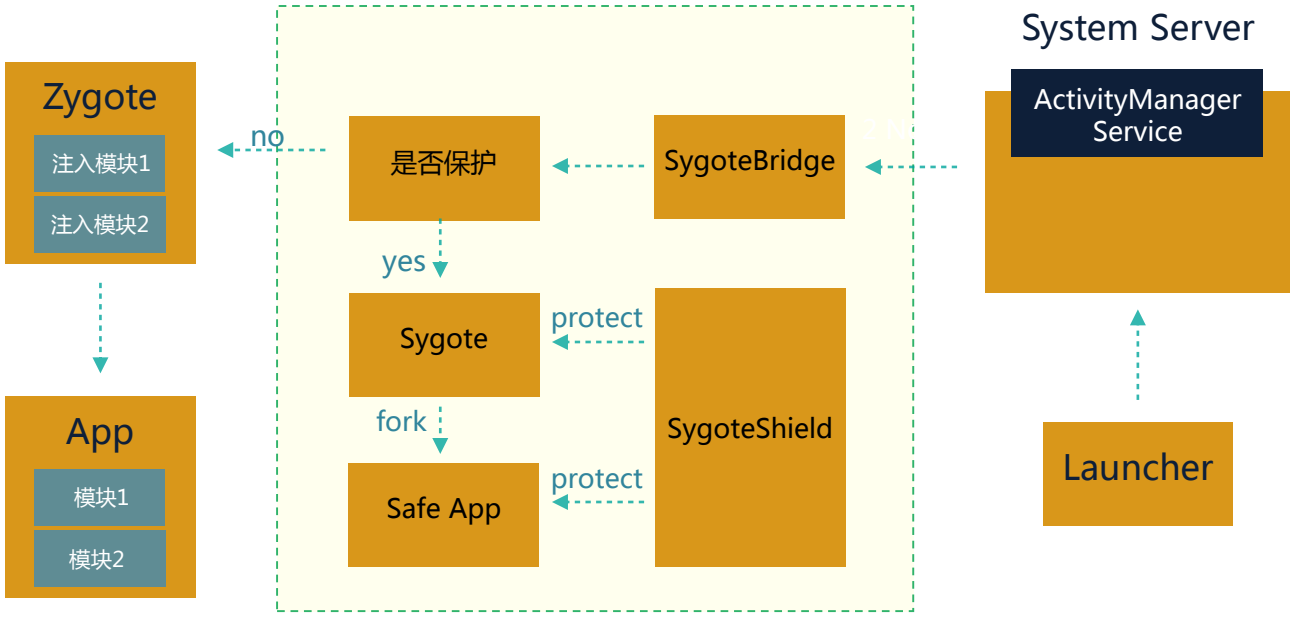
运行时风险实例


99% 上午 10:55
facebook
test@gmail.com
登录
忘记密码？
facebook帐号:
帐号: test@gmail.com
密码: 123456
注册 Facebook

创建可信的运行环境

- 创建可信的Zygote进程
- 保护可信的Zygote
- 让App从可信的Zygote进程孵化

Safe Zygote - 安全入口方案



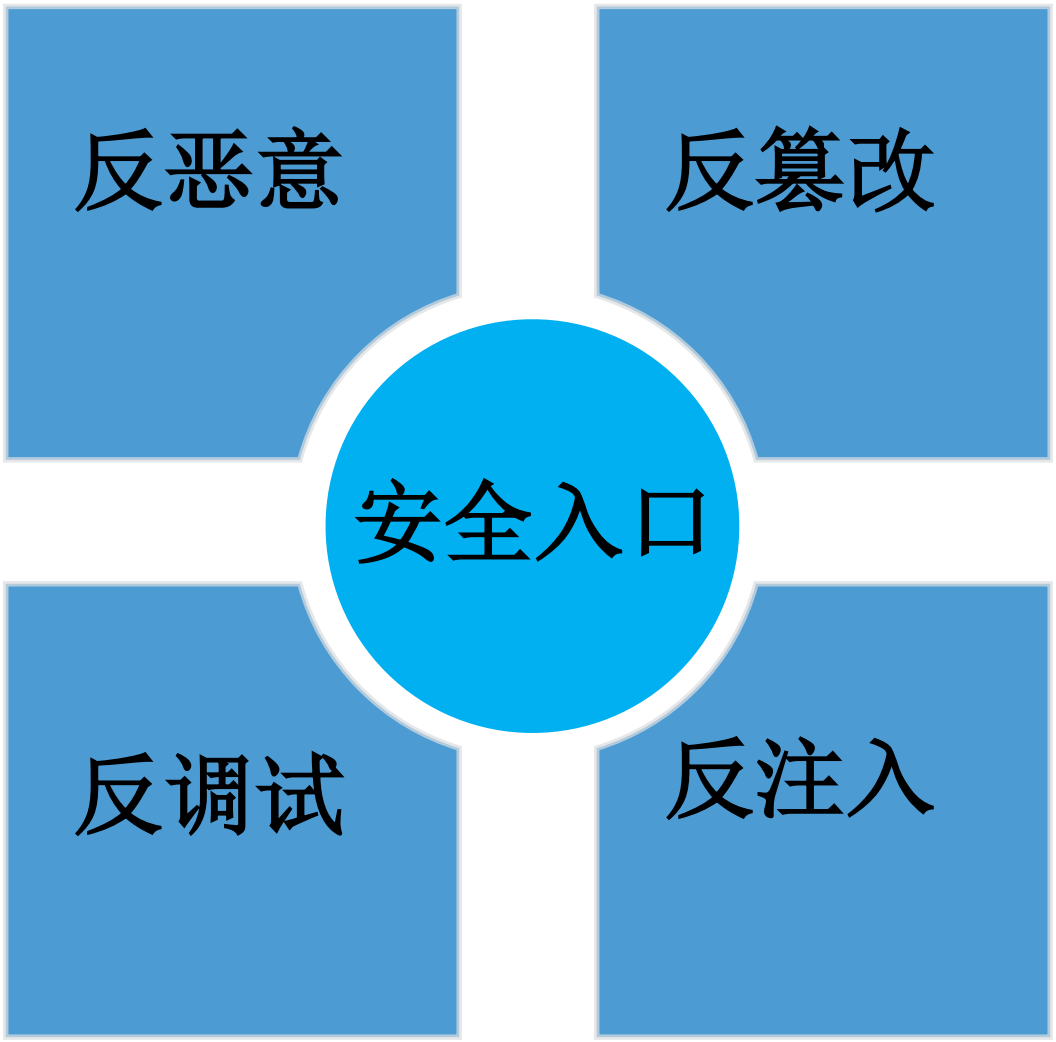
安全入口方案演示


风险列表
XPosed框架54
忽略
退出


11:34
SafeAppBox
Start Safe Box Server
Google Play 声音搜索
com.google.android.ears
com.qualcomm.timeservice
com.qualcomm.timeservice
应用包访问权限帮助程序
com.android.defcontainer
Xposed 安装器
de.robv.android.xposed.installer
安全先锋
kvpioneer.cmcc
Exchange 服务
com.google.android.exchange
com.android.providers.partnerbookmarks
com.android.providers.partnerbookmarks
Set Safe App List
Exit

4 + 1 下一代支付威胁终结者

4 + 1 安全支付方案



反恶意 让恶意软件无处藏身

恶意软件识别系统

云端扫描引擎

支付安全
SDK

病毒扫描
漏洞扫描
应用解固
高识别率

- 隐私窃取
- 恶意扣费
- 远程控制
- 恶意传播
- 资费消耗
- 系统破坏
- 诱骗欺诈
- 流氓行为

百度移动杀毒引擎以 100% 查杀率、零误报、性能满分，在 AV-Test 评测中蝉联四连冠

反篡改 盗版软件现原形

- 篡改

  是指将应用反编译后，替换原有应用中的部分文件，然后重新打包，分发。是制作盗版应用常用方法。

- 高风险

  任意位置，增加恶意代码

- 解决方法

  正版验证，检测盗版

  应用加固，阻止盗版

反篡改 方案

手机之家	1.0	盗用品牌	包名为 zhu.baidumusic	http://apk.imobile.com.cn/detail-apk_id-45353.html
安智市场	1.0	盗用品牌	包名为 zhu.baidumusic	http://www.anzhi.com/soft_714404.html 注 1
赚网安卓吧	1.0.1	盗用品牌	None	http://apk.zhuannet.com/soft/324.html
安酷市场	1.0	盗用品牌	None	http://www.apkchina.net/apps/3941 注 2

反调试 为应用的运行保驾护航

- 调试

动态调试是指黑客用特定调试器控制应用，通过“下断点+单步”的方式来跟踪应用运行的流程。

动态调试的目的一般为当静态无法分析出应用的代码逻辑时，需要动态运行应用以查看程序实际运行时的流程和变量的值。

- 风险

高：程序运行被控制，当前状态被窥探。

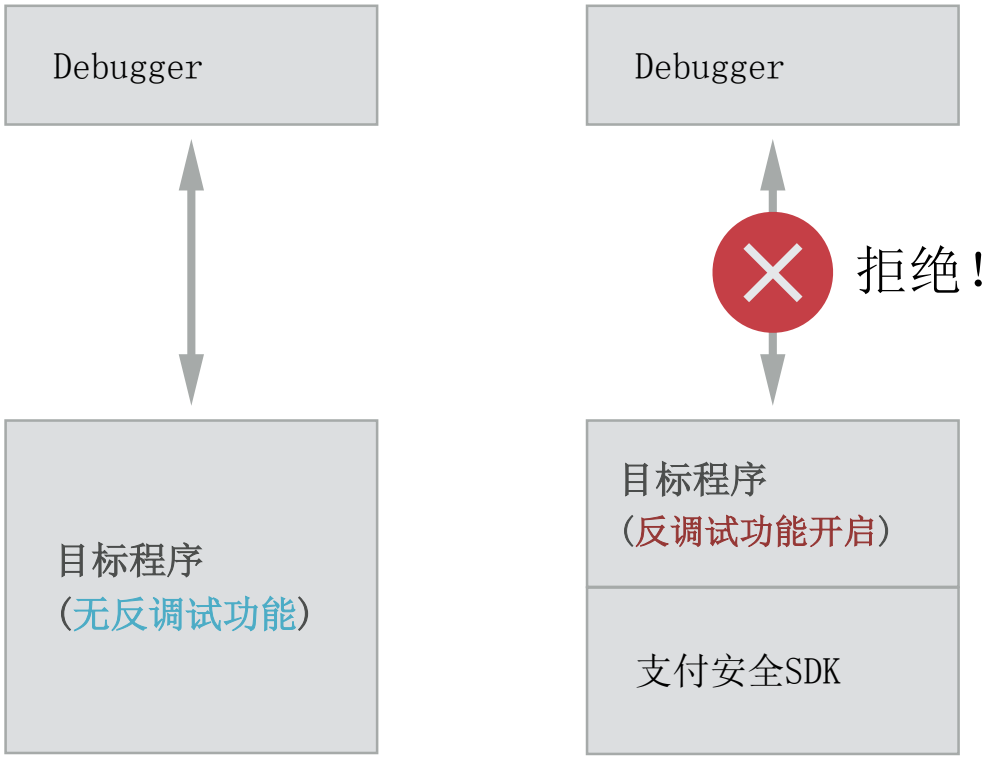
- 解决方法

调试检测及应对

程序启动时，开启反调试防护措施

反调试 为应用的运行保驾护航

反调试功能开启前后对比



反注入 确保应用本身环境安全

- 注入

在应用运行时,其内存空间里被其他应用插入一段代码并且执行，这段被插入的代码通常会做一些风险较高的行为，比如偷取密码等。主流安全软件也会使用代码注入方法，用来监听其他应用里的特殊行为

- 高风险

当前程序的数据和运行被劫持.

- 解决方法

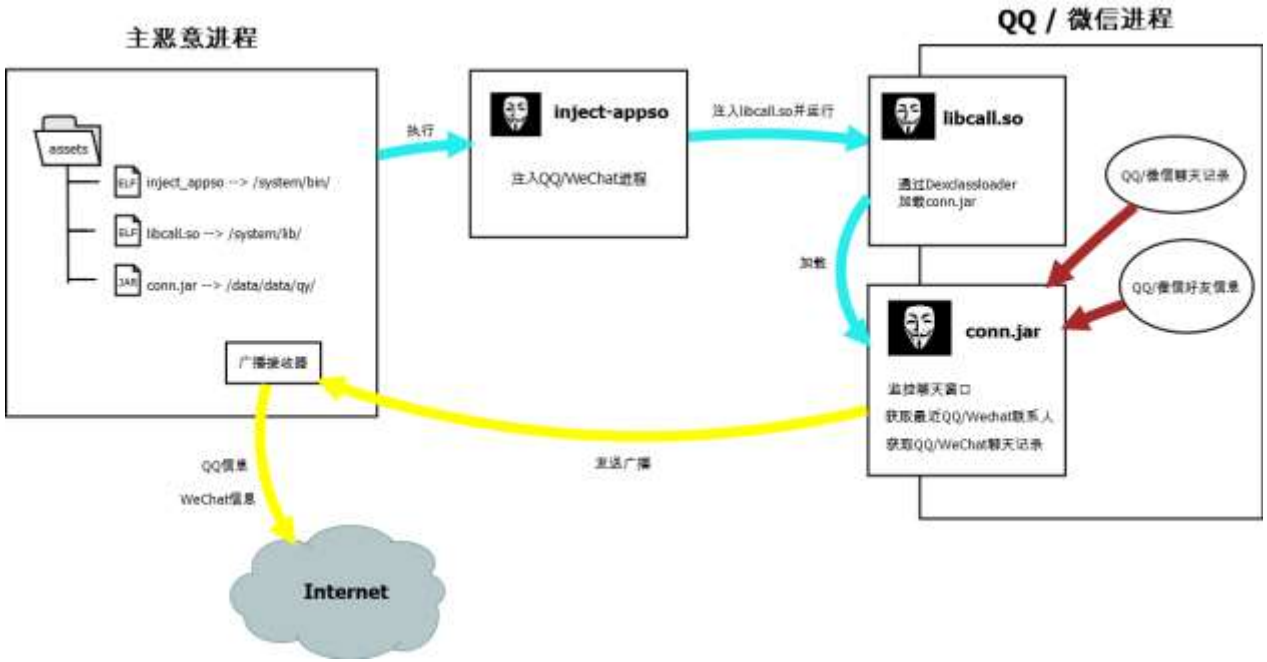
扫描应用进程中被注入的模块

清理应用中被注入的模块，保证应用进程干净安全

反注入 确保应用本身环境安全

- 微信大盗

病毒注入到微信，窃取微信聊天记录



安全支付总结

4 + 1

- 反恶意：让恶意软件无处藏身
- 反篡改：盗版软件现原形
- 反调试：保证应用不被跟踪
- 反注入：确保应用不被劫持
- 安全入口：提供安全、可信的运行环境

从应用启动，到应用运行，提供全面防护！

合作应用 百度理财 V1.4.5

- 百度理财

精选投资、保险、众筹、贷款等各类金融产品，
全面满足各类家庭投资理财、资金借贷、资金消费等金融需求。

- 风险分析

1 篡改风险

2 注入风险

3 调试风险

谢谢