GIAC | 全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

APK行为监控与分析

演讲者：王浩
奇虎360手机卫士主防组 技术经理

2017.thegiac.com

概述

1.为何要对应用进行监控

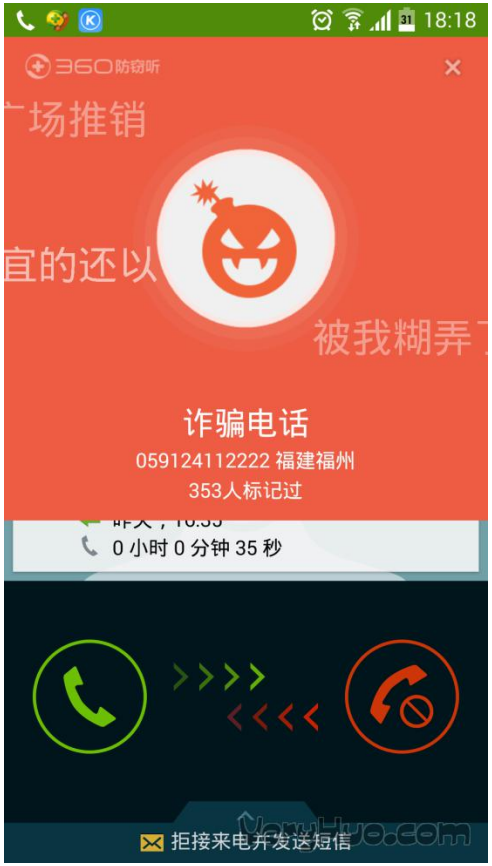
2.监控技术实现的几种技术方案

3.为何要建立自动监控体系

4.如何建立自动监控体系

你是否也有过这些困惑

自从安装了某APP之后，越来越多的骚扰电话



你是否也有过这些困惑

自从安装了某APP之后，手机上的应用越来越多了

你是否也有过这些困惑

安装了APP之后，手机越来越卡，发热严重

删了这些APP？

痛点是：这些APP我真的需要，而且不是病毒

我们其实可以解决这个问题

建立监控体系，监管手机APP权限，让APP不在非黑即白

不止于此

- 病毒行为分析
- 测试与性能监控
- 定制化需求

目录

1.为何要对应用进行监控

2.监控技术实现的几种技术方案

3.为何要建立自动监控体系

4.如何建立自动监控体系

监控技术实现的几种方案

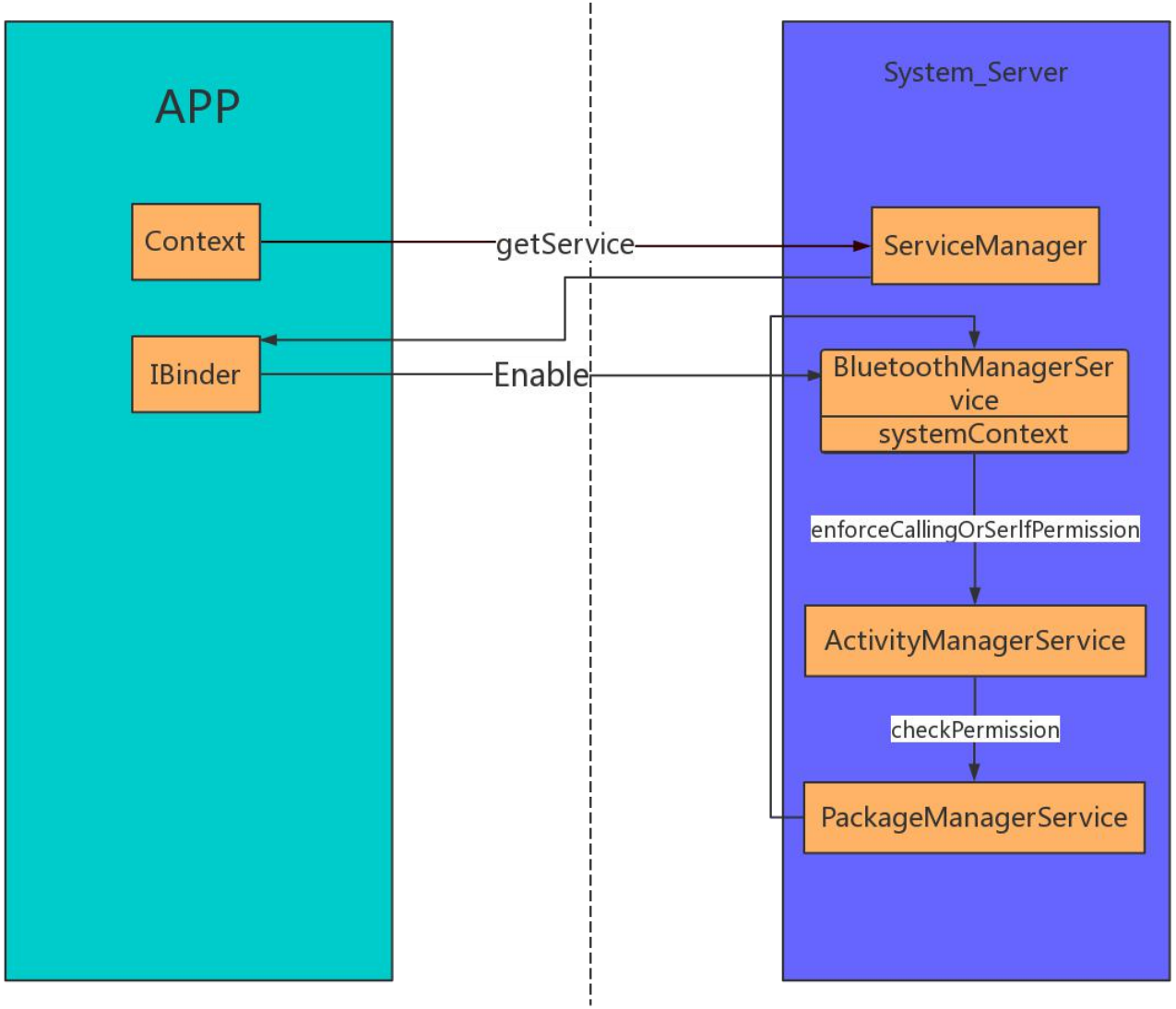
- 直接修改源码编译ROM
- 注入以及HOOK
- 重打包方式
- 分身技术

修改源码举例

- 依照打开蓝牙为例
- BluetoothAdapter.getDefaultAdapter().enable();
- android.Manifest.permission.BLUETOOTH_ADMIN

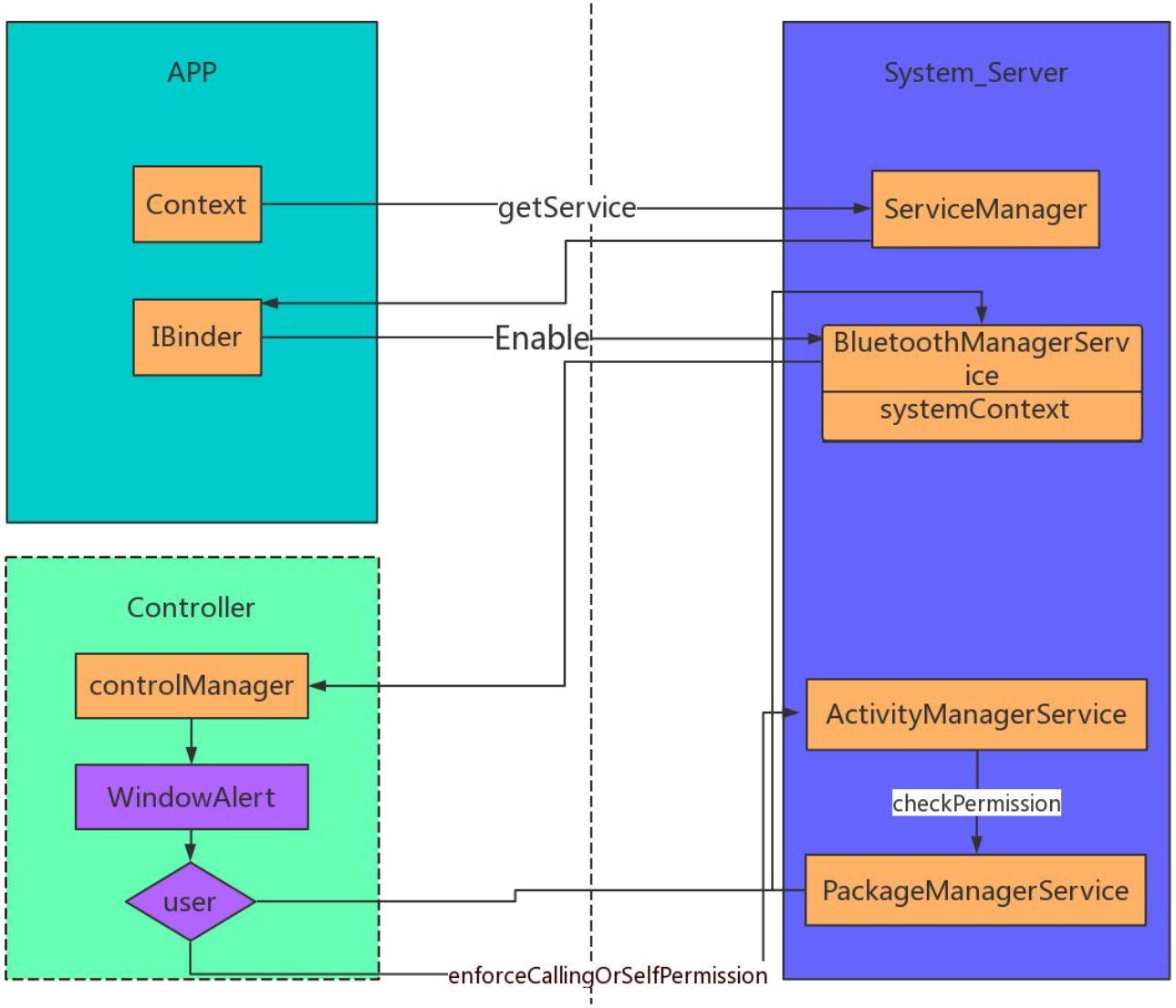
修改源码

基础权限模型


APP
Context
IBinder
getService
Enable
System_Server
ServiceManager
BluetoothManagerSer
vice
systemContext
enforceCallingOrSerlfPermission
ActivityManagerService
checkPermission
PackageManagerService

修改源码

进行修改之后


APP
Context
IBinder
getService
Enable
System_Server
ServiceManager
BluetoothManagerService
systemContext
Controller
controlManager
WindowAlert
user
ActivityManagerService
checkPermission
PackageManagerService
enforceCallingOrSelfPermission

修改源码总结

- 找寻感兴趣的函数点直接修改
- 修改函数尽量选择参数较全的
- 在SystemServer和Zygote中的修改必须try catch

监控技术实现的几种方案

- 直接修改源码编译ROM
- 注入以及HOOK
- 重打包方式
- 分身技术

注入以及hook——注入方式

ptrace注入

- PTRACE_ATTACH
- PTRACE_GETREGS
- PTRACE_POKETEXT
- PTRACE_SETREGS
- PTRACE_CONT
- PTRACE_DETACH

regs

- regs->ARM_pc = addr;
- regs->ARM_lr = 0;

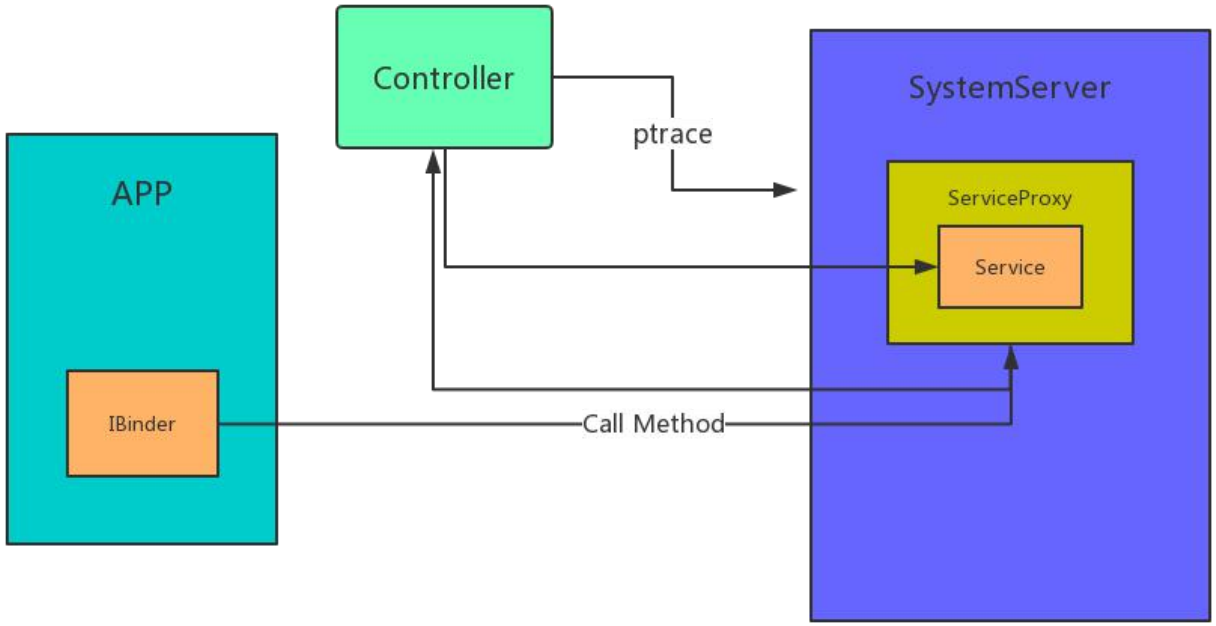
waitpid

注入以及hook——hook方式分类

- JAVA HOOK
- Native Hook
- 纯C函数的hook

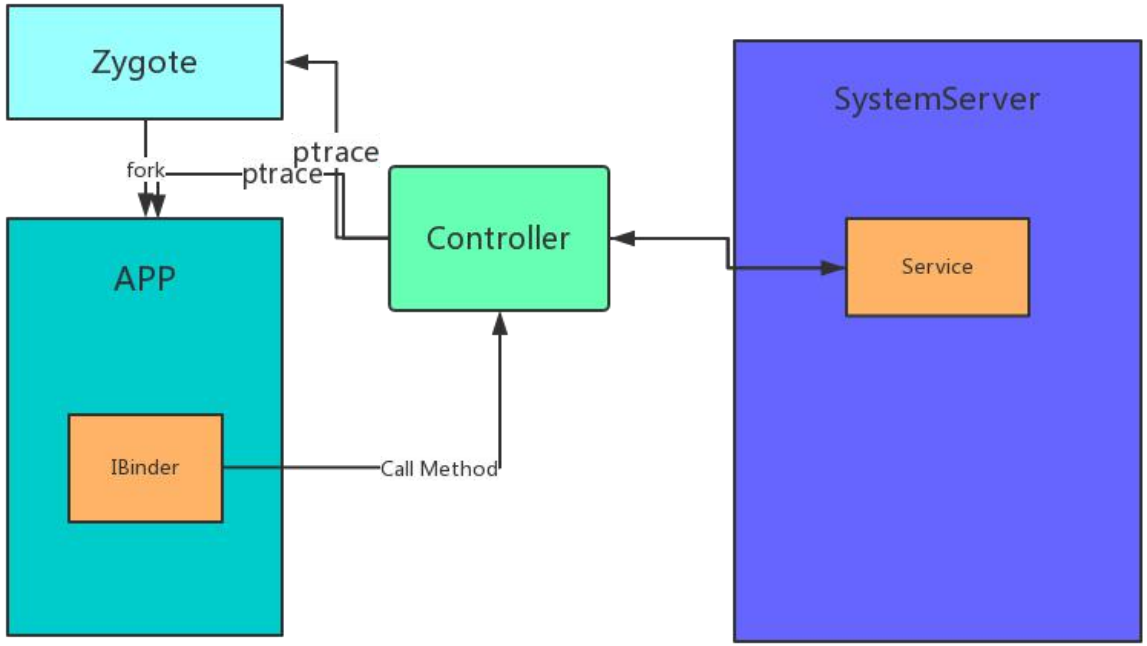
注入以及hook

- 随时注入随时生效
- 如果需要限制应用行为，那么针对hook函数返回值要谨慎处理
- 某些service需要native层做hook
- Controller的效率要做到极致



注入以及hook

- 可以选择是否注入Zygote
- 会有一定冲突概率，hook点争抢

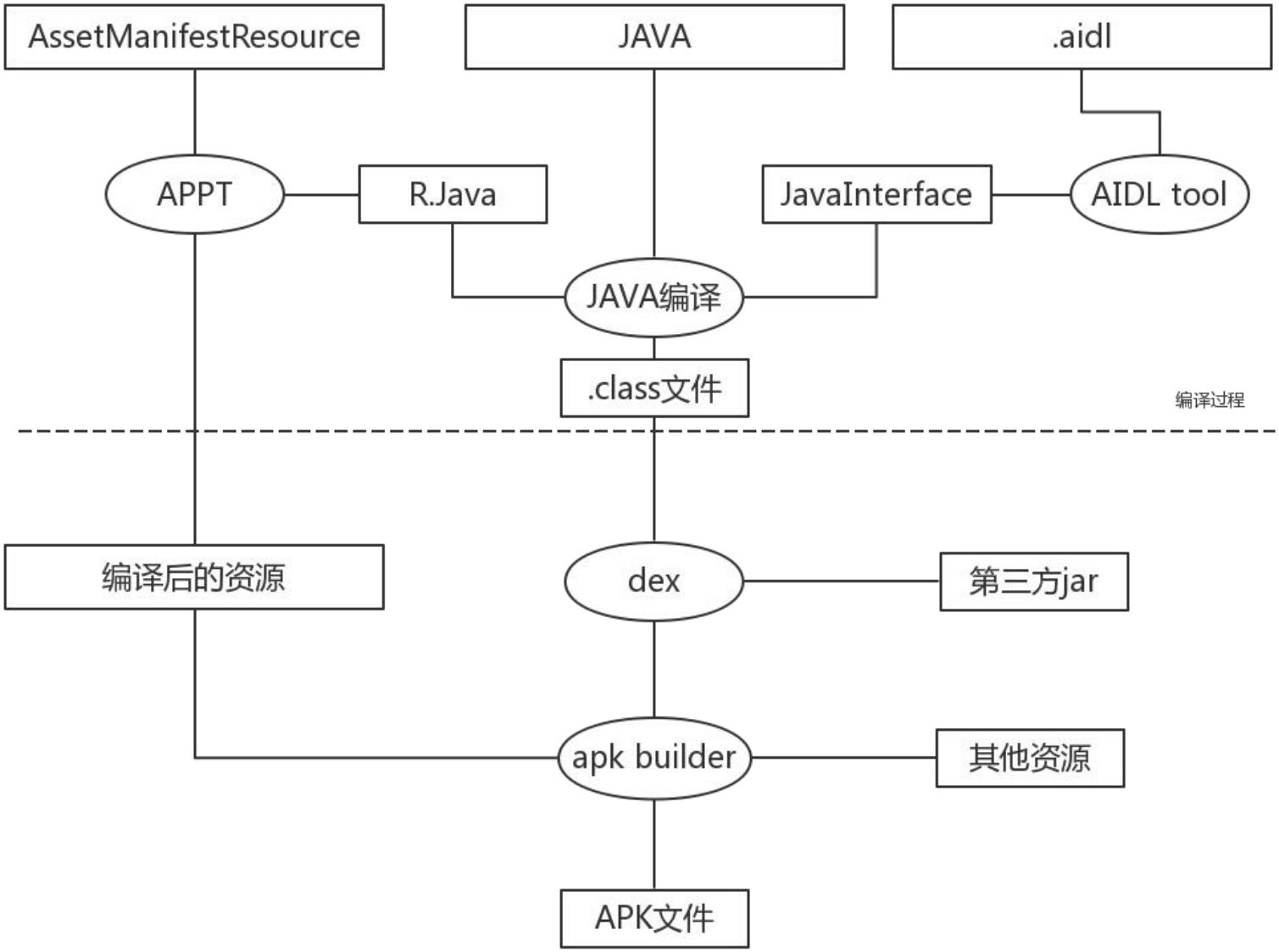


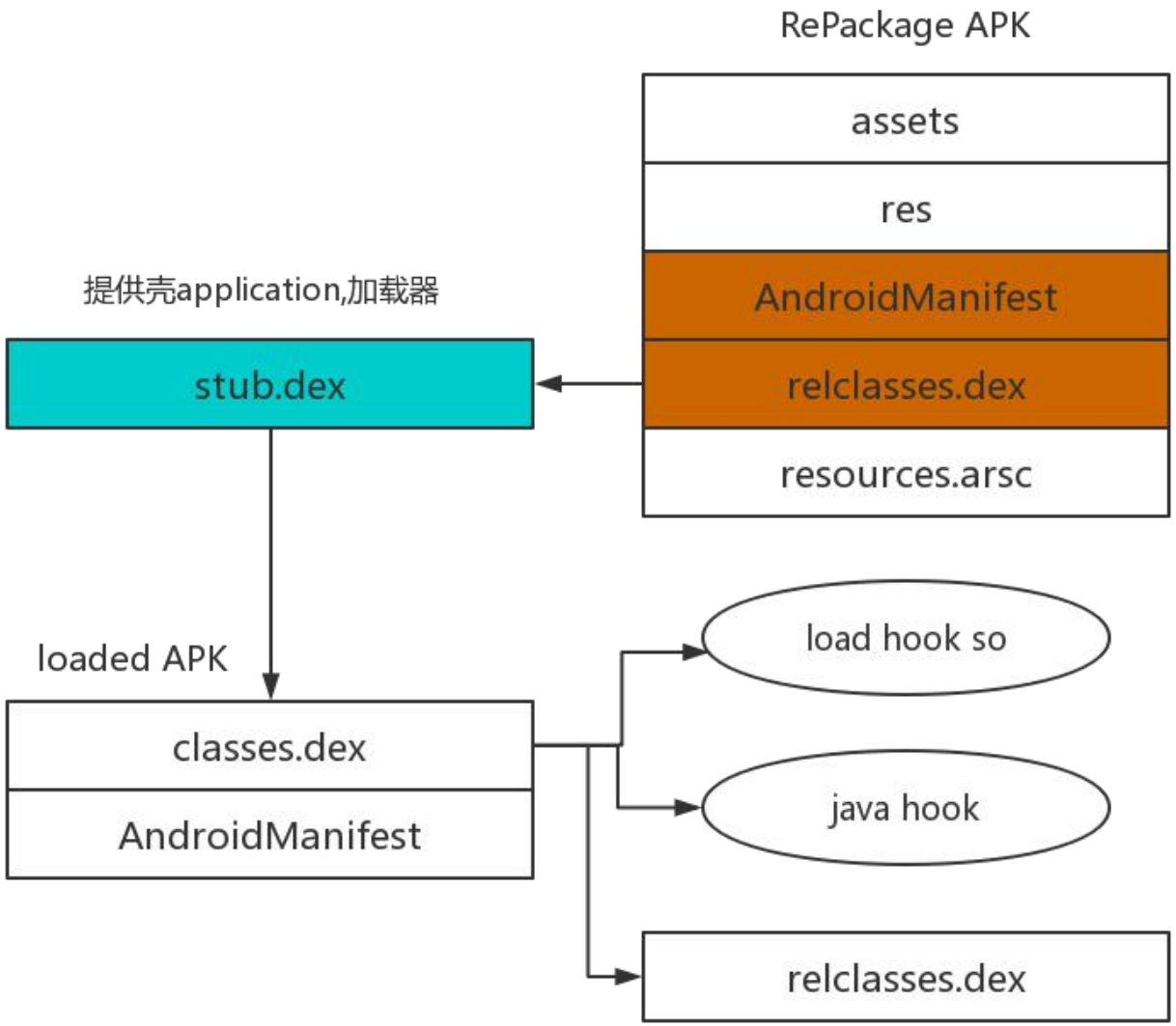
注入以及HOOK总结

- 对HOOK技术的掌控要求较高
- 寻找HOOK点要考虑时机
- 尽量做到动态hook，在真实使用的时候在去hook，避免出现在调用堆栈

监控技术实现的几种方案

- 直接修改源码编译ROM
- 注入以及HOOK
- 重打包方式
- 分身技术


AssetManifestResource
JAVA
.aidl
APPT
R.Java
JavaInterface
AIDL tool
JAVA编译
.class文件
编译过程
编译后的资源
dex
第三方jar
apk builder
其他资源
APK文件

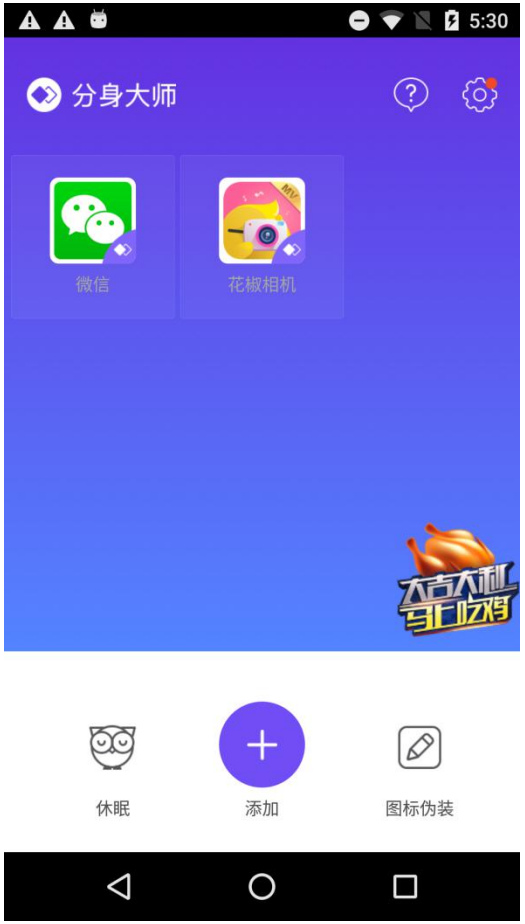

RePackage APK
assets
res
AndroidManifest
relclasses.dex
resources.arsc
提供壳application,加载器
stub.dex
loaded APK
classes.dex
AndroidManifest
load hook so
java hook
relclasses.dex

监控技术实现的几种方案

- 直接修改源码编译ROM
- 注入以及HOOK
- 重打包方式
- 分身技术

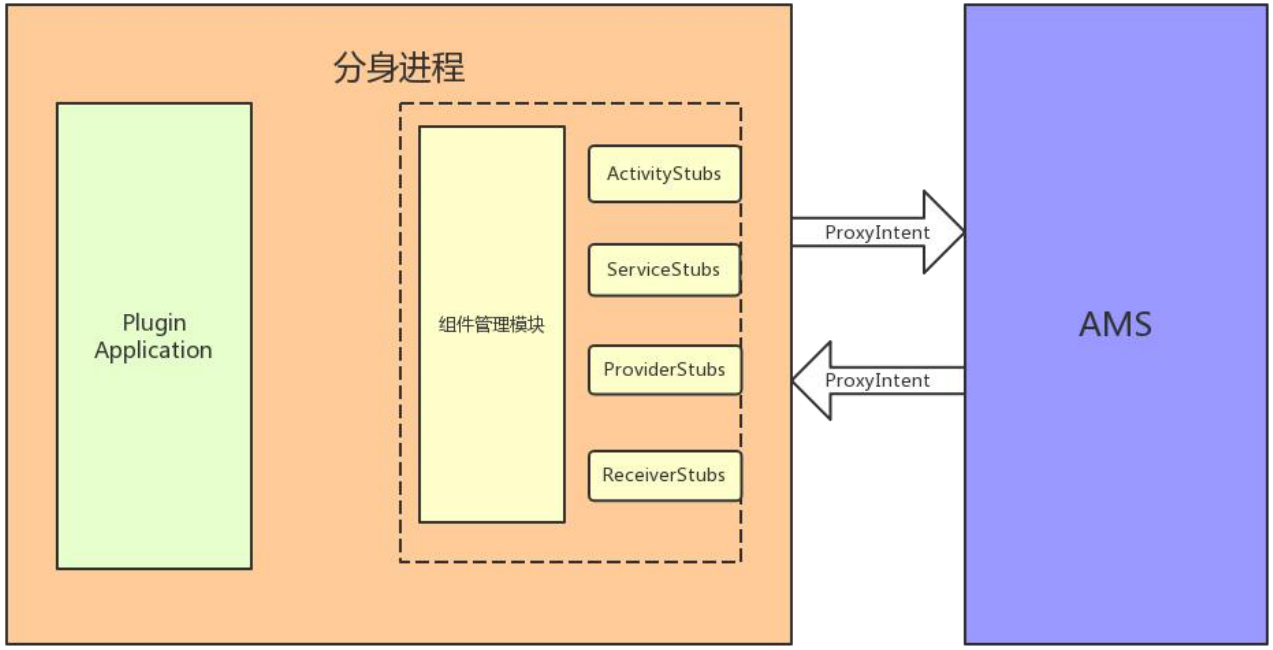
分身技术

- 基于沙箱机制打造
- 内部运行原生的Android应用
- 依赖Android的Hook机制
- 轻量级的Android虚拟机



分身技术

- Android四大组件代理
- Application初始化



分身技术

- 每一种组件都要制定方案代理
- HOOK量巨大
- APK千差万别

监控技术实现的几种方案

修改源码

- 需要编译rom，无法搞定机型问题

注入以及hook

- 需要适配不同的安卓版本
- 需要适配不同的品牌ROM
- 需要手机有root权限

重打包

- 需要适配加固APP
- 需要更换签名

分身方式

- 需要适配加固的app
- 开发以及维护难度较大

目录

1.为何要对应用进行监控

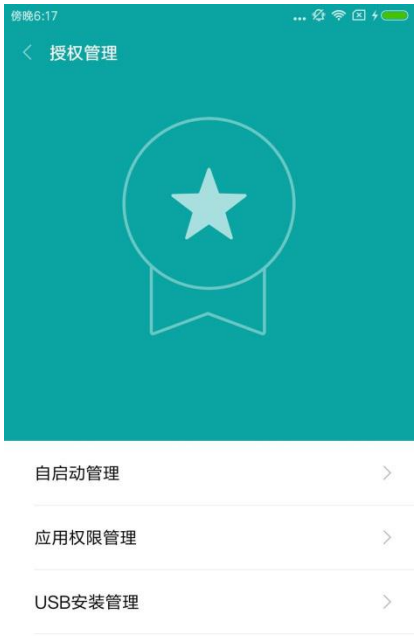
2.监控技术实现的几种技术方案

3.为何要建立自动监控体系

4.如何建立自动监控体系

为何要建立自动监控体系

厂商提供的监控能力越来越完善，安全厂商需要差异化



为何要建立自动监控体系

- 给出的运营建议值多是靠收集手机用户数据
- TOP覆盖较好，其他应用基数较少，容易有误差
- APP数量太大，人工无法完全运营
- APP的权限也会有变化

目录

1.为何要对应用进行监控

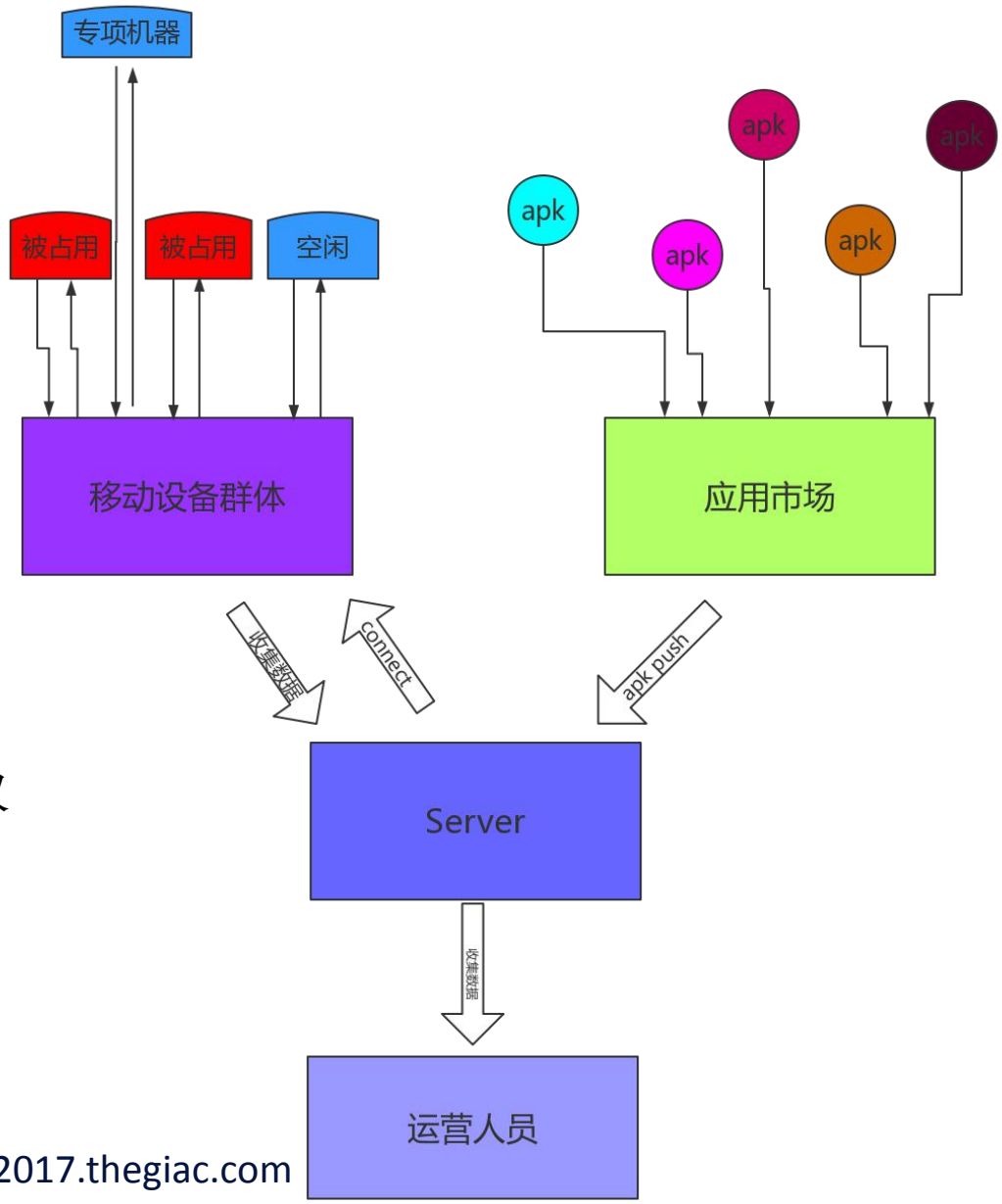
2.监控技术实现的几种技术方案

3.为何要建立自动监控体系

4.如何建立自动监控体系

如何建立自动监控体系

- 架设设备群
- 结合应用市场
- 机群部署监控方案
- 导出运营数据，生成合理化建议



如何建立自动监控体系

构建设备群

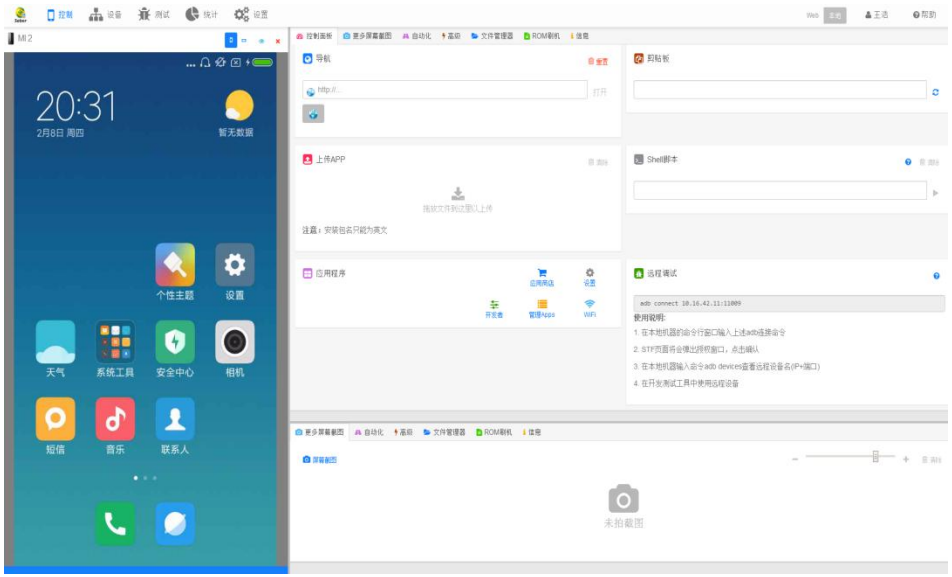
Stf	AirDroid	Mobizen
开源	非开源	非开源

尽量采用真机环境

合理利用现有机器资源

利用机器的空闲时间

可以随时批量添加撤换设备



通过应用市场获取目标apk

- 通过应用市场获取最近版本更新的TOP　APP包
- 通过应用市场获取增长率最快的包
- 通过应用市场获取标签分类下TOP包
- 声明权限有较多增加的包

选取监控技术方案

- 混合之前讲过的监控方案，确保数据获取成功
- 利用自定义ROM作为保底方案

如何自动运行app

- 利用app monkey
- 人工设计app点击，运行点击脚本，配合点击截图
- AI使用app

自动监控生成的数据

- xxx app enable the bluetooth, allow, process as usual
- 配合申请权限时的截图

鉴定应用合理行为

- 针对应用类别进行标签标注
- 只保证应用的主要功能
- 不涉及到页面交互的拉起行为全部记录，禁止

写在最后——给应用开发者的一点建议

- 正视被注入
- 尽量采用开源SDK
- 不过多的申请权限
- 本地不保存过多信息，数据加密

GIAC | 全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

扫码关注GIAC公众号

2017.thegiac.com