



浅谈移动养殖系统和IOS安全

安天移动安全研发中心

www.antiy.com

 安天 | 智者安天下



提纲

- 一：
 - 威胁与风险
 - 解决方案
 - 系统介绍
 - 技术支持
-
- 二：
 - IOS安全防范



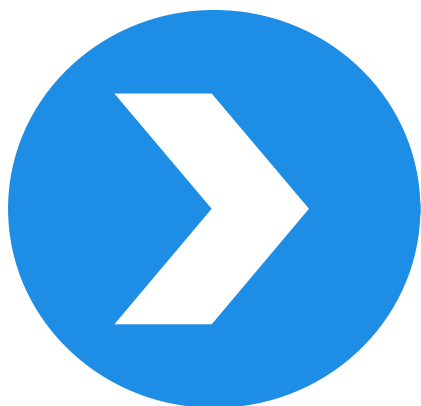
前言 Introduction

Question :

动态养殖是什么？

为什么有这个需求？

带来的效应是什么？



威胁与风险

- 恶意样本数量增速
- 恶意样本种类繁多
- 当前的检测方式
- 恶意代码的检测问题



恶意样本数量增速

5



2015-11-24

5789431

投放总数：**73893**

进入筛分的新样本数：**53460**

投放筛分丢失的样本数：**0**

2700777

休眠中入：**17155**

休眠中出：**88**

Question：增速之快怎么解决？数量之多呢？



恶意代码种类繁多

6

```
android.support.v4
├─ a
├─ app
└─ b
    ├── NilvaVOSi
    ├── WIyFuRtM
    ├── cvmhnQkE
    ├── tFqpIdzx
    ├── wXUZSeZB
    └─ c
        ├── media
        ├── view
        └─ widget
```

```
net.emrgw.rsik
├─ ㄗ
├─ 口
├─ ぼ
├─ 口
├─ ㄗ
├─ 俾
├─ 剝
├─ 噪
├─ 整
├─ 異
├─ 梗
├─ 粹
├─ 煙
└─ 光
```

```
private ㄗ(ㄗ arg1) {
    this.ㄗ = arg1;
    super();
}

public final void 予(ㄗ arg5, boolean arg6) {
    ㄗ v2 = arg5.剝();
    int v3 = v2 != arg5 ? 1 : 0;
    ㄗ v0 = this.ㄗ;
    ㄗ v1 = v3 != 0 ? v2 : arg5;
    ㄗ v5 = ㄗ.予(v0, v1);
    if(v5 != null) {
        if(v3 != 0) {
            ㄗ.予(this.ㄗ, v5.ㄗ, v5, v2);
            ㄗ.予(this.ㄗ, v5, true);
        }
    }
}
```

```
com
├─ bangcle.protect
│   └─ ACall
│       ├── ApplicationWrapper
│       ├── FirstApplication
│       ├── MyClassLoader
│       └─ Util
└─ tg.wangzi.com.zz
    ├── a
    ├── fr
    └─ pr

neo.proxy
└─ DistributeReceiver
```

```
Object v2 = paramBundle.get("ozsrsDJaPgDiPpdvHRuHYJs
.replaceAll("[vRTJBAhKDCyojXHMYPWS]", "").s
ugnsnhnb.feapeiha = "esoars izttsic ooy";
ugnsnhnb.evqhlrf = ((int)Math.round((((double)ugnsn
86))) + 91.5 - (45.25 + (((double)ugnsnhnb.a
)) - ugnsnhnb.irorsatl - 15380.9873) + (((d
))) - 15.37 + 9607.13633));
int[] v16_2 = ugnsnhnb.taiayd;
```

```
ㄗ
口
口
口
口
```

```
</activity>
<receiver := "net.emrgw.rsik.ㄗ" />
<activity := "net.emrgw.rsik.ㄗ" />
<activity := "net.emrgw.rsik.ㄗ" />
<activity := "@style/m" := "net.emrgw.rsik.ㄗ" />
```

```
mytool
APKMainAPP126A8
BeibobaDevice
BeibobaReceiver
BeibobaService
```

```
import android.app.Application;

public class APKMainAPP126A8 extends Application {
    public APKMainAPP126A8() {
        super();
        System.loadLibrary("APKProtect");
    }
}
```

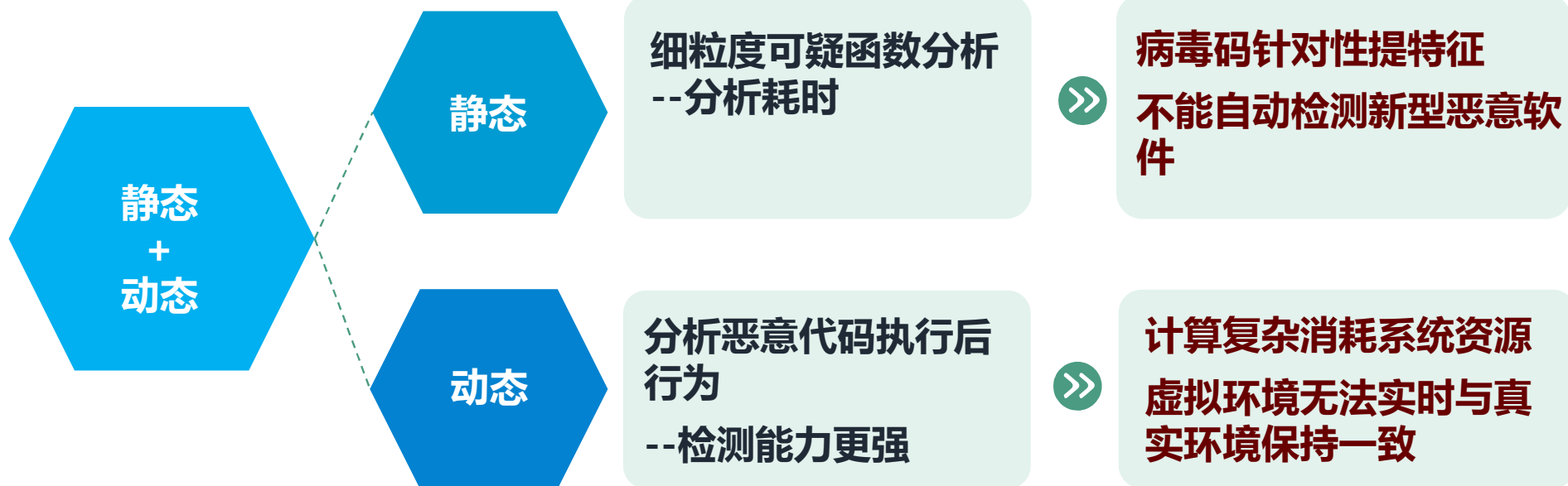
Question：恶意代码种类的增多，人工如何面对？

例：so核心功能隐藏、混淆、防反编译、自校验等加大静态分析成本。



目前检测方法的缺陷

7





恶意代码检测的问题

8



1

Dex、no Certificate、games无法检测

2

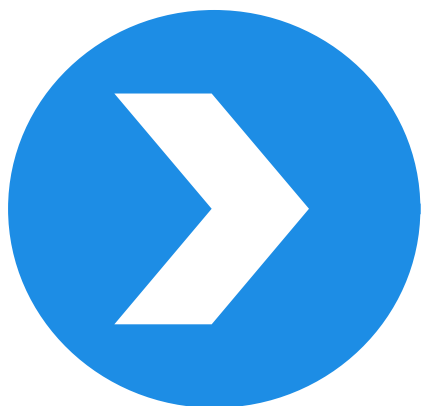
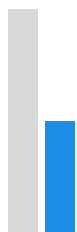
动态分析技术无法够根据恶意代码行为需要，动态模拟检测环境以提升检测能力---例：模拟填写

3

沙箱技术容易被恶意代码检测并逃逸check deviceId

4

原生ROM镜像分析周期长



解决方案

- 需求
- 构思图
- 关键技术



解决方案----需求

10



=





解决方案----构思图

11

•定制化监控参数设置

- 自动化流程操作
- 定制化配置调节



•丰富样本行为触发分析能力

- 动态模拟满足系统核心资源
- 全面行为触发&反逃逸能力升级



动态养殖分析系统



- 基于Java堆栈原理，实现敏感代码调用追踪

•恶意行为代码级函数调用追踪

- 以时间为序进行行为触发展示
- 以危险性对行为分级
- 四种标准格式输出



•详细结果展示和标准格式输出

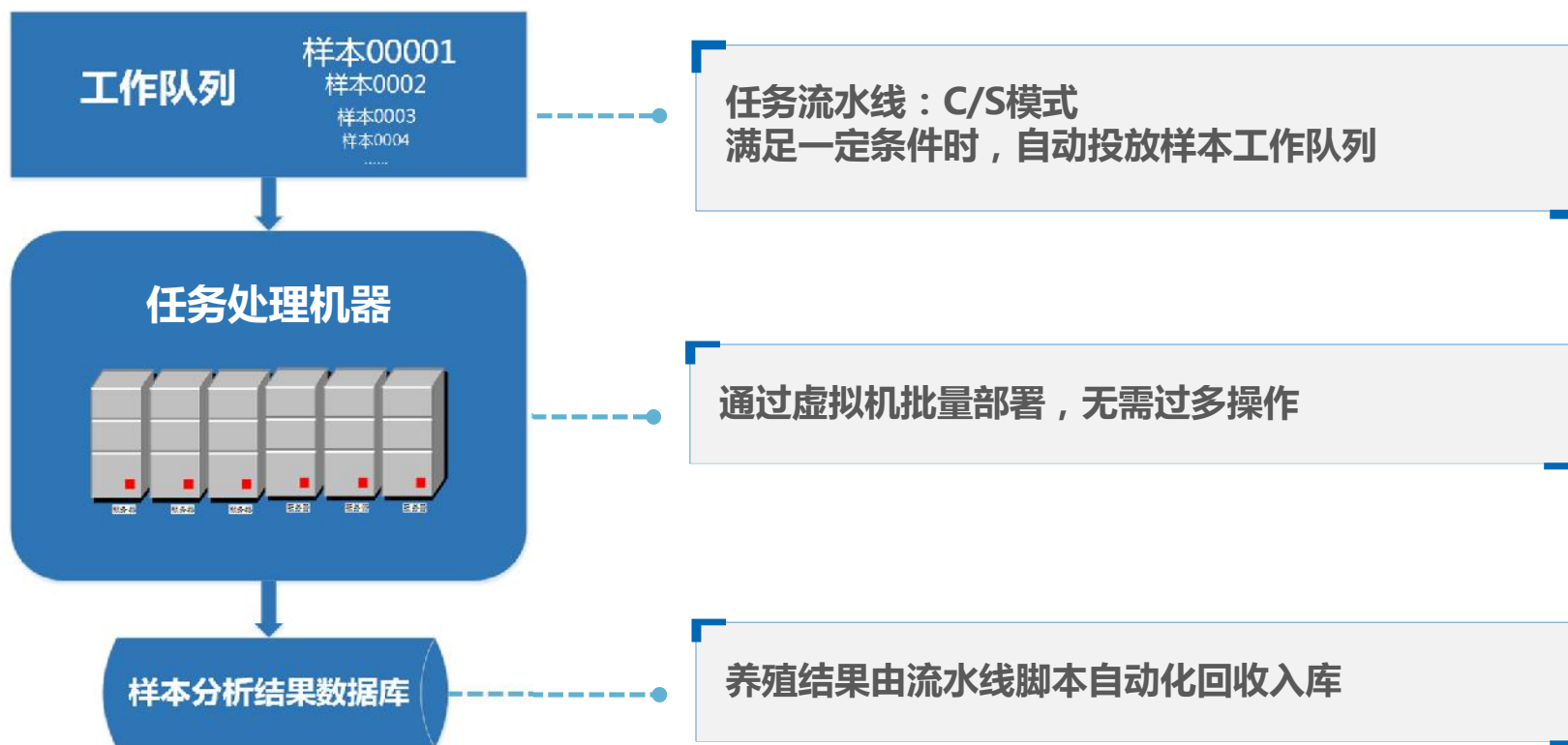


关键技术----自动化流程

12

1

自动化流程操作



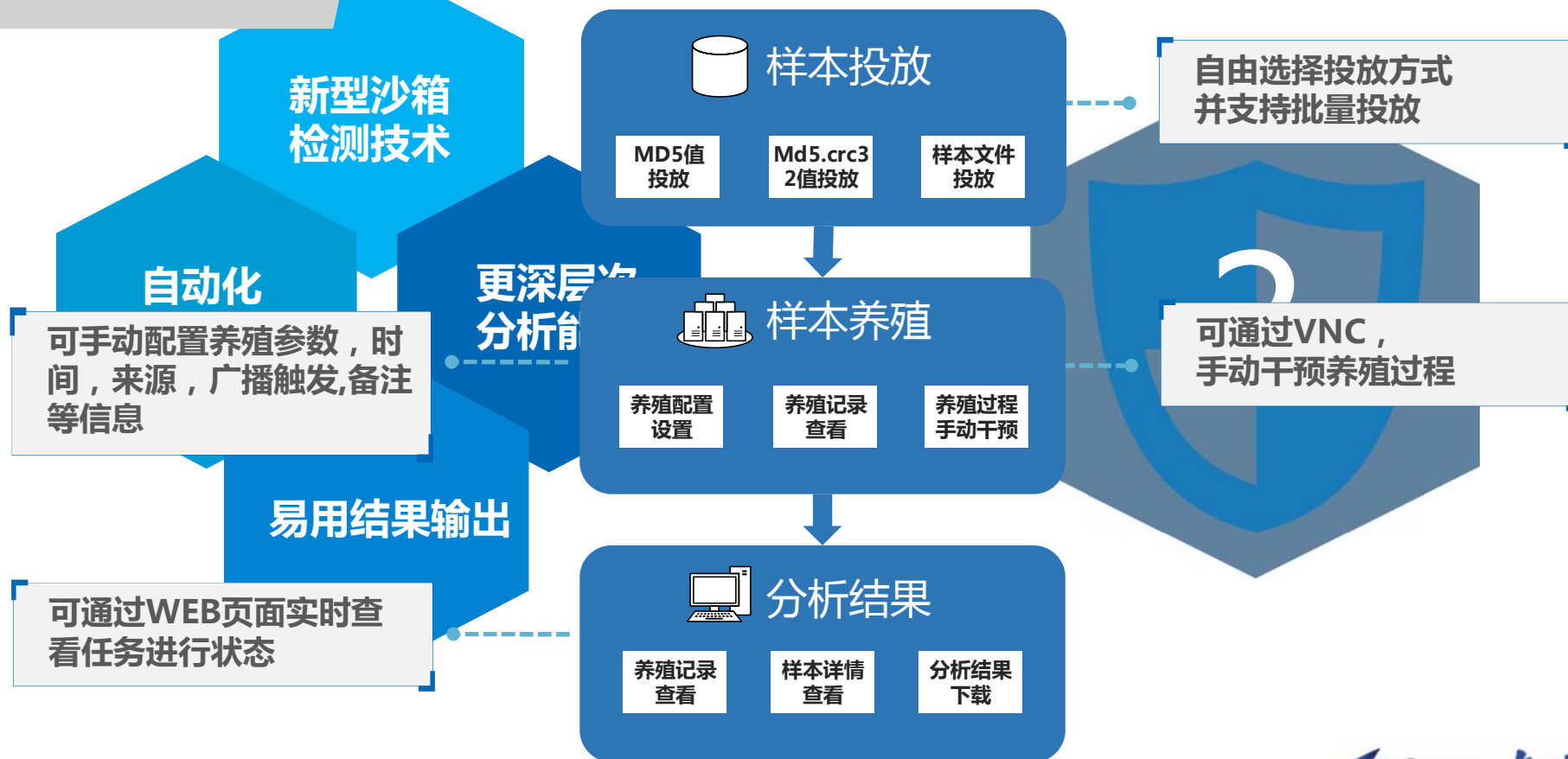


关键技术----定制化监控参数设置

13

2

定制化配置调节





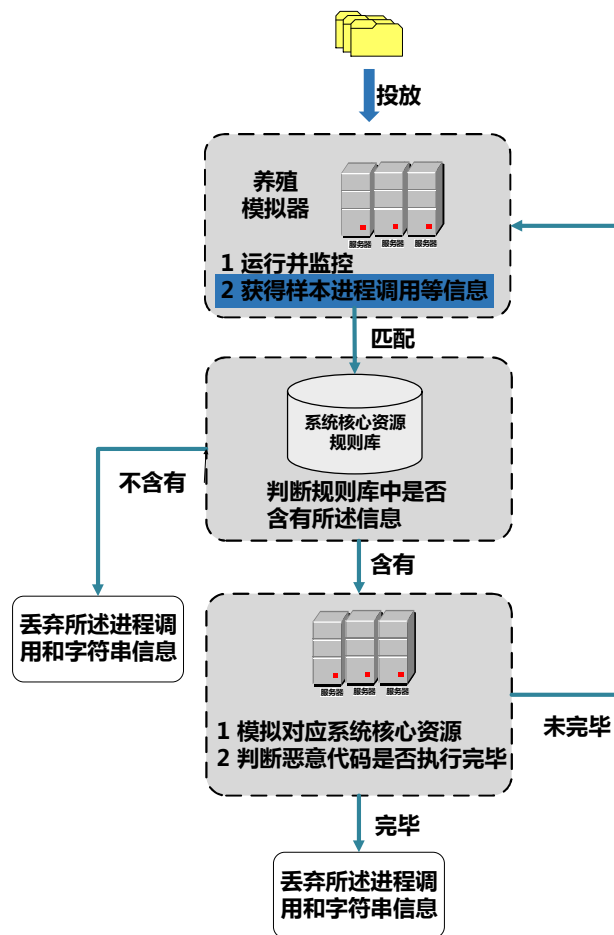
关键技术----丰富样本行为触发分析能力

14

1：动态模拟满足系统核心资源

1 当恶意代码未能执行完毕时，能够根据恶意代码行为需要，动态模拟满足系统核心资源，减少资源消耗，提升检测效率。

动态模拟流程





关键技术----ROM改造

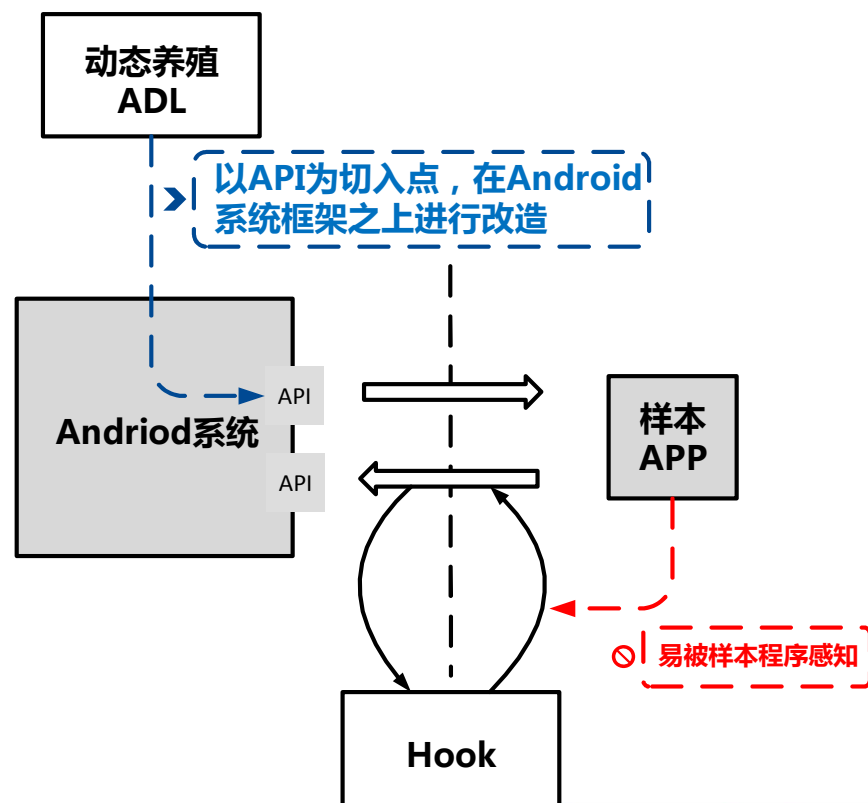
15

2

2 : 行为触发&针对反逃逸

行为捕获基于rom的改造，对于应用是透明不可见，提升反逃逸能力

基于rom的改造



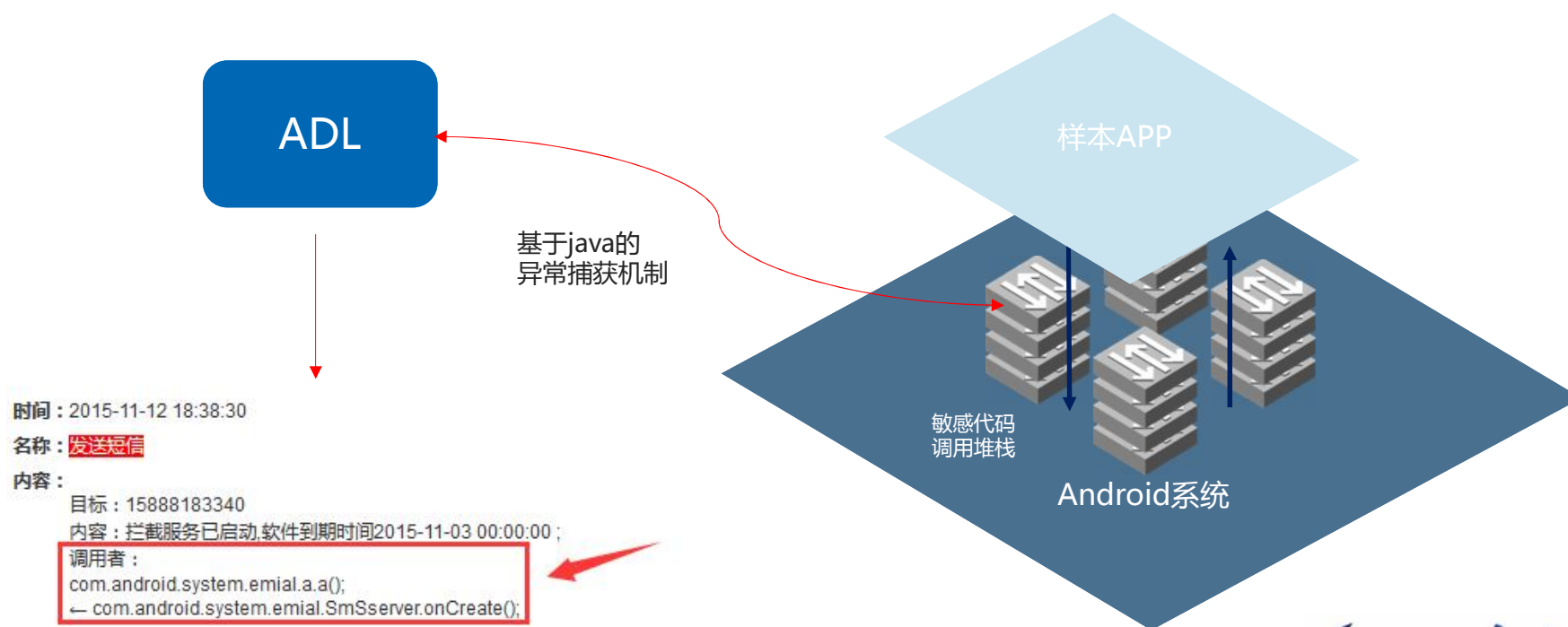


关键技术----恶意行为代码级函数调用追踪

16

1

对于样本行为有详细函数调用信息，便于进行代码级追踪分析。



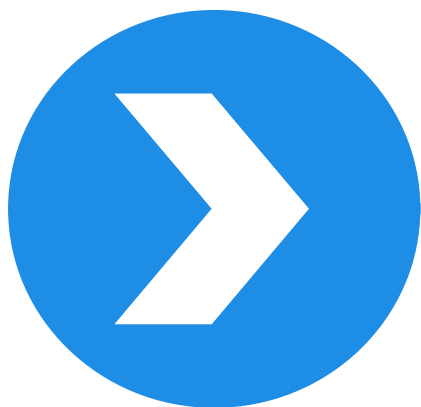


以时间为序的详细结果展示，以及行为捕获日志行形成标准化行为文档进行输出



本地行为(182)	网络行为(29)	敏感注册(30)	敏感行为(28)
地址：192.168.1.1代表无信任设备，192.168.1.100代表无信任设备			
时间：2015-11-11 11:38:10 名称：192.168.1.1 内容： 来源：来自：dataapi.mind.1141100053.inp 目标：dataapi.com.beyaa.com/x3d-1.asp		时间：2015-11-11 11:38:37 名称：192.168.1.1 内容： 来源：19063320 设备：cmd+1，设备传输IDU格式002003810163802J00J511111116J032J03033336J08L01	
时间：2015-11-11 11:38:38 名称：192.168.1.1 内容： data\data.com.beyaa.com/wj/mis/plugin/mado-adk-plugin.asp		时间：2015-11-11 11:38:38 名称：192.168.1.1 内容： 来源：data\data.com.beyaa.com/wj/cache/mis-adk-plugin1447513115340 目标：data\data.com.beyaa.com/wj/mis/plugin/mado-adk-plugin.asp	
时间：2015-11-11 11:38:38 名称：192.168.1.1 内容： 来源：19063320 设备：cmd+1，设备传输IDU格式002003810163802J00J511111116J032J03033336J08L01		时间：2015-11-11 11:38:41 名称：192.168.1.1 内容： 来源：19063320 设备：cmd+1，设备传输IDU格式002003810163802J00J511111116J032J03033336J08L01	
时间：2015-11-11 11:38:32			

以时间为序，对样本行为进行跟踪记录并展示



技术细节

- 定制ROM
- 定制化监控参数设置



1

技术细节：

1: 修改packages/apps/.....

2: the key of function打印log

---http post

---SMS_send

---Device_admin

3: 修改packages/apps/..... imsi&imei对抗 check monitor



2

技术细节：

4: add TimerTask 定时发送短信

5: Analog Click

---应对email

---行为触发

6: Ubuntu编译源码



技术细节----定制化监控参数设置

21

3

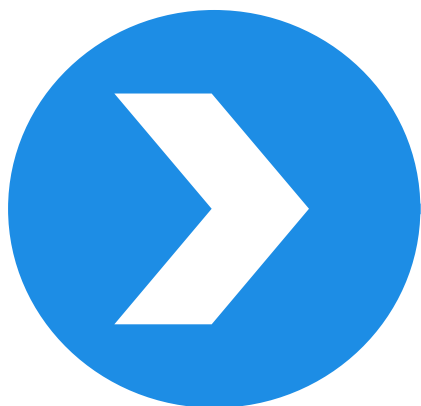
参数特点：

Application 层函数节点

例：sendDataMessage 简化SMS_SEND

类型	类型名称	类型说明	内容说明		类型	类型名称	类型说明	内容说明
短 信	SMSSF	往目标号码短信发送	目标号码地址;发送内容		安装卸载	INSTALLPKG	安装应用程序apk	ADB/NOR/pkg Name
	SMSINTERCEPT	拦截系统短信程序接收短信,收件箱中查看不到短信	拦截系统短信程序接收短信,收件箱中查看不到短信			DELETEPKG	卸载应用程序	packageName
底层操作	EXEC/FEEXEC	执行Linux底层命令,如su、mount等命令	执行命令 命令参数		录音	STARTRECSND	开始录音	API调用记录
网络连接相关	URLCREATE	包含url 的URL对象创建	url		拍照	TACKEPIC	开始拍照	PREV:OPEN/CLOSE (是否开启预览)
	HTTPPOST	Http Post操作的URL	url		外部加载	外部加载	加载dex文件	path

android.intent.action.SIG_STR	手机信号强度
android.intent.action.BATTERY_CHANGED	电量改变
android.intent.action.BOOT_COMPLETED	启动完成
android.intent.action.TIME_TICK	时钟广播
android.intent.action.SCREEN_ON	屏幕解锁
android.intent.action.SCREEN_OFF	锁屏广播
aandroid.intent.action.MEDIA_MOUNTED	SDCard挂载
android.intent.action.UMS_CONNECTED	USB进入U盘模式
android.intent.action.CALL	拨打电话
android.intent.action.DIAL	拨打电话
android.intent.action.CFF	语音电话的呼叫转移状态已经改变
android.intent.action.CONFIGURATION_CHANGED	设备的配置信息
android.intent.action.DATA_ACTIVITY	电话的数据活动(data activity)状态已经改变
android.intent.action.DATA_STATE	电话的数据连接状态已经改变
android.intent.action.DATE_CHANGED	日期被改变
android.server.checkin.FOTA_CANCEL	取消所有被挂起的 (pending) 更新下载
android.server.checkin.FOTA_INSTALL	更新已经被确认,马上就要开始安装
android.server.checkin.FOTA_READY	更新已经被下载,可以开始安装
android.server.checkin.FOTA_RESTART	恢复已经停止的更新下载
android.server.checkin.FOTA_UPDATE	通过 OTA 下载并安装操作系统更新
android.intent.action.PACKAGE_ADDED	设备上新安装了一个应用程序包
android.intent.action.PACKAGE_REMOVED	设备上删除了一个应用程序包



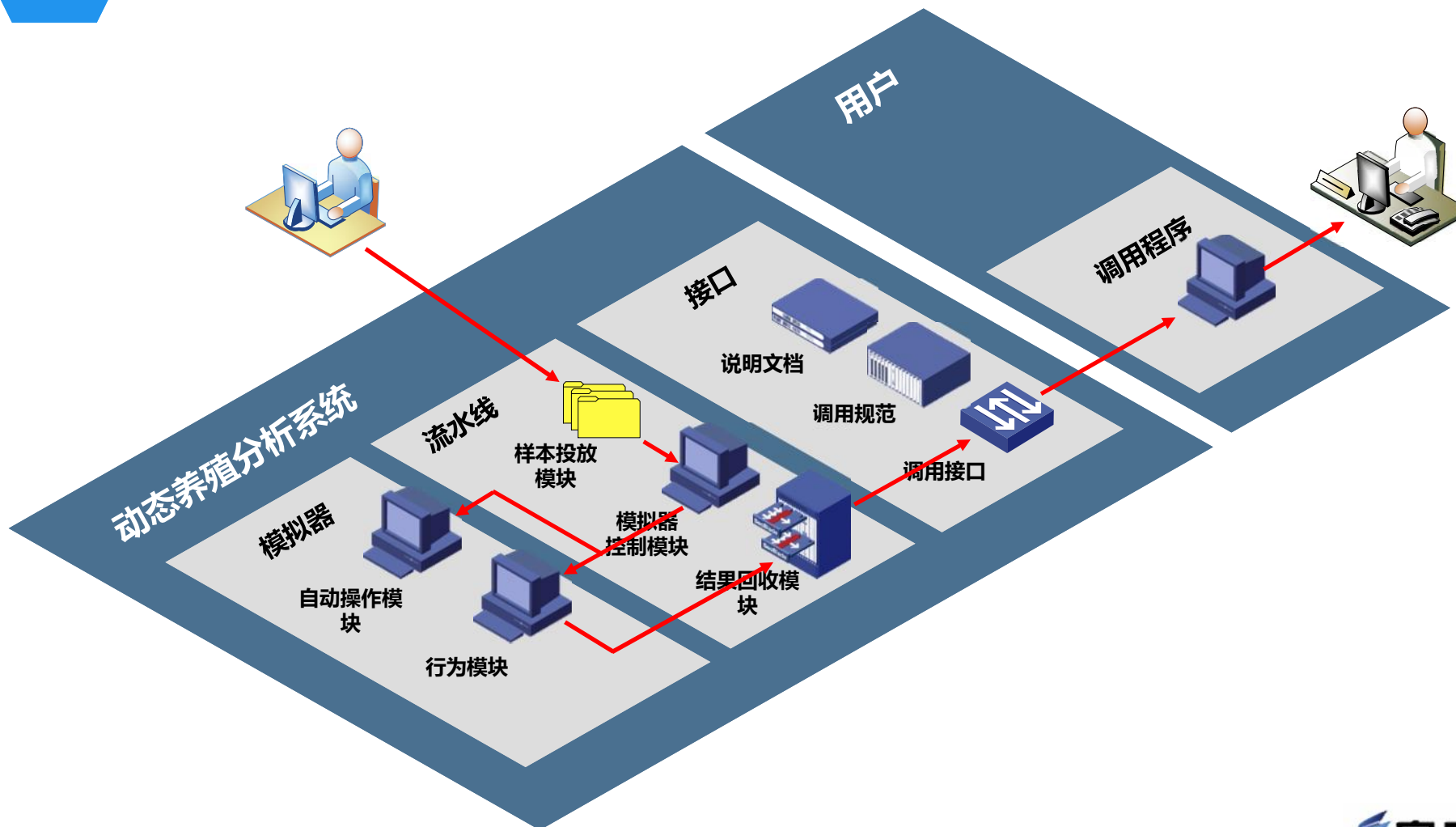
系统介绍

- 系统体系结构
- 成熟案例应用
- 产品实际案例



安天动态养殖分析系统----系统体系架构

23

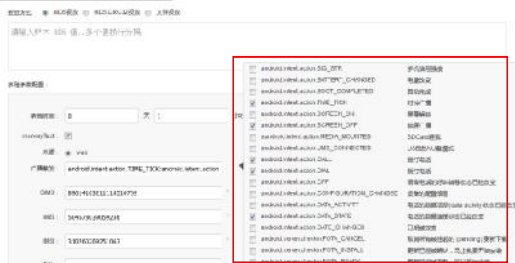




安天动态养殖分析系统----成熟案例应用

24

1 任务分发



2 任务列表

任务ID	任务名称	来源	操作人	截止时间	开始时间	结束时间	任务状态	任务优先级	备注
1701	C:\ProgramData\Antiy\Antiy\Antiy.exe	web	Antiy	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	进行中	高	
1702	C:\ProgramData\Antiy\Antiy\Antiy.exe	web	Antiy	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	进行中	高	
1703	C:\ProgramData\Antiy\Antiy\Antiy.exe	web	Antiy	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	进行中	高	

3 服务器状态

服务器ID	服务器名称	IP地址	操作系统	CPU使用率	内存使用率	磁盘使用率	网络流量	连接数	备注
1701	192.168.1.101	Windows 7	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	
1702	192.168.1.102	Windows 7	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	
1703	192.168.1.103	Windows 7	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	

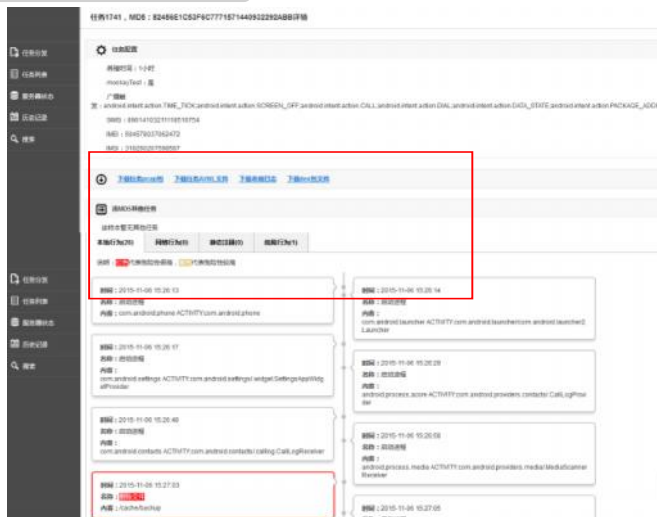
4 历史记录

任务ID	任务名称	来源	操作人	截止时间	开始时间	结束时间	任务状态	任务优先级	备注
1701	C:\ProgramData\Antiy\Antiy\Antiy.exe	web	Antiy	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	进行中	高	
1702	C:\ProgramData\Antiy\Antiy\Antiy.exe	web	Antiy	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	进行中	高	
1703	C:\ProgramData\Antiy\Antiy\Antiy.exe	web	Antiy	2015-11-17 22:00:00	2015-11-17 22:00:00	2015-11-17 22:00:00	进行中	高	

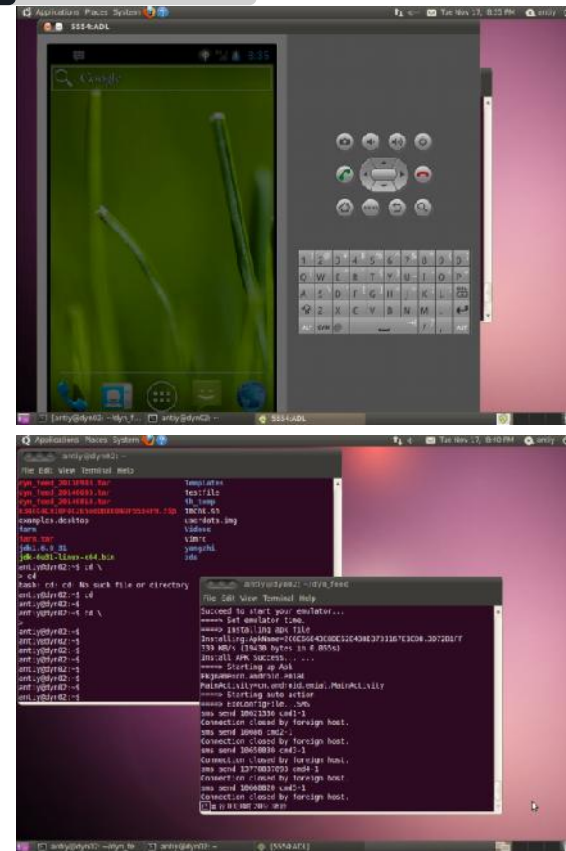
5 任务分发



6 样本养殖详情



7 查看VNC





安天动态养殖分析系统----产品实际案例

25

MD5: 70E62332ECD56EF3C73F316F944DAA3D

解密获取手机号码

时间: 2014-07-12 02:00:39 名称: 解密

内容: 输入: 2e3def3af08b4c57151e2ace7606400e(H);
输出: 1851069843;
调用者:
com.example.test.DesUtils.decrypt(DesUtils.java);
← com.example.test.DesUtils.decrypt(DesUtils.java);
← com.example.test.u.getData(u.java);
← com.example.test.n.getSendTo(n.java);
← com.example.test.n.chkIsFirstRun(n.java);
← com.example.test.u.chkFirstRun(u.java);
← com.example.test.la.onCreate(la.java);

解密获取数据

时间: 2014-07-12 02:00:39 名称: 解密

内容: 输入: af707591cd7a19633bc18e1940d7e59b04aa14b6d827d392(H);
输出: 2014-05-01 11:30;
调用者:
com.example.test.DesUtils.decrypt(DesUtils.java);
← com.example.test.DesUtils.decrypt(DesUtils.java);
← com.example.test.u.getData(u.java);
← com.example.test.n.getSendTo(n.java);
← com.example.test.n.chkIsFirstRun(n.java);
← com.example.test.u.chkFirstRun(u.java);
← com.example.test.la.onCreate(la.java);

时间: 2014-07-12 02:00:40 名称: 调用类方法

内容:
android.telephony.SmsManager.sendMultipartTextMessage("1851069843","null","null");
←;
调用者:
com.example.test.Tx.tx(Tx.java);
← com.example.test.u.sendMsg(u.java);
← com.example.test.n.chkIsFirstRun(n.java);
← com.example.test.u.chkFirstRun(u.java);
← com.example.test.la.onCreate(la.java);

激活设备管理器

时间: 2014-07-12 02:00:40 名称: 启动活动

内容:
com.android.settings.DeviceAdminAdd 行为: android.app.action.ADD_DEVICE_ADMIN;
调用者:
com.example.test.la.onCreate(la.java);

时间: 2014-07-12 02:00:40 名称: 发送短信

内容:
目标: 1851069843
内容: 安装成功啦;
调用者:
java.lang.reflect.Method.invokeNative(Native Method);
← java.lang.reflect.Method.invoke(Method.java:530);
← com.example.test.Tx.tx(Tx.java);
← com.example.test.u.sendMsg(u.java);
← com.example.test.n.chkIsFirstRun(n.java);
← com.example.test.u.chkFirstRun(u.java);
← com.example.test.la.onCreate(la.java);



对数据进行回溯

污点数据：

时间：2014-09-05 16:10:51 名称：**发送短信**

内容：

目标：13685841

内容：提示：设备已安装，等待激活；

调用者：

me.rt6t54rjhfrtry5tfhfg.rt4dfhfgj.tool.MeTianxiaOuChanUtil.se(MeTianxiaOuChanUtil.java:56);

← me.rt6t54rjhfrtry5tfhfg.rt4dfhfgj.activity.MeTianxiaOuChanActivity.onCreate(MeTianxiaOuChanActivity.java:21);

定位：



污点数据：

VIP.M@il 新浪邮箱

搜索邮件、通讯录、网盘文件等

邮箱首页 通讯录 邮箱网盘 应用 收件夹

写信 收信 删除 举报 移动 标记 回复 转发 更多

收件夹 5

星标邮件 ★

草稿夹

已发送

垃圾邮件

更多分类

已删除

所有邮件

网站通知

商讯信息

代收邮箱

m13724072328 8月3日 ★

全部信息 (A00000482D1434) - 95588您尾号3078卡6日12:33POS支出(跨行消费)6...

m13724072328 8月3日 ★

联系人 (A00000482D1434) - 房产科办公室—0276786 小付办公室—02782821...

m13724072328 8月3日 ★

联系人 (864938026433934) - 北京大娘—0106582 李纯—130018 朱...

m13724072328 8月3日 ★

全部信息 (864938026433934) - 106551 7尊敬的用户, 1863191 至7月...

m13724072328 8月3日 ★

联系人 (355666052109203) - 沈阳—+861370 赵凡—+861527267070...

m13724072328 8月3日 ★

全部信息 (355666052109203) - 95533张翔5月27日13时20分向您尾号4717的储蓄...



二 :

前言 Introduction

Question :

IOS非常安全？

没有越狱呢？

隐私泄露的代价？



IOS安全事件----未越狱设备

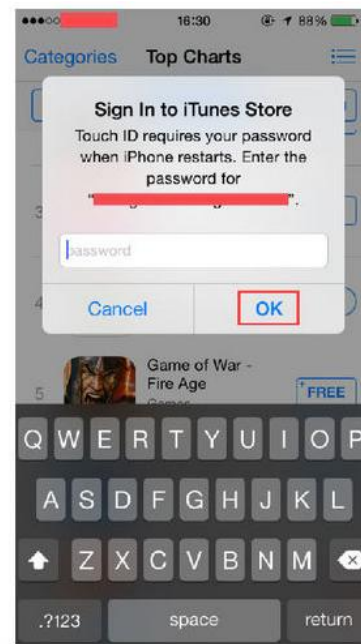
28

事件1: Hacking Team泄露样本无需越狱即可监控iOS用户
—2015年7月

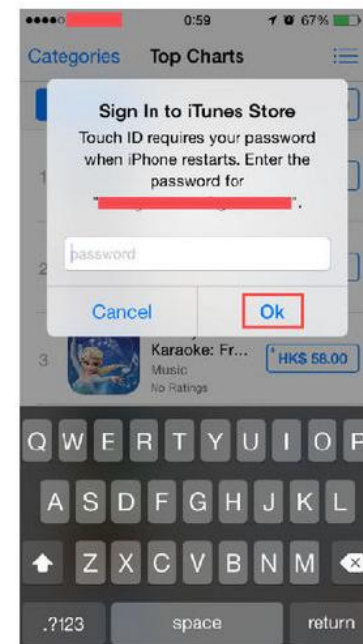


事件2: CVE-2015-5838 触发沙箱逃逸
— 2015年9月

Ios8.1未越狱设备，伪造app弹出虚假登录框盗取Apple ID



原版的AppStore登录窗口



伪造的AppStore登录窗口



IOS安全事件----越狱设备

29

事件3：KeyRaider盗取超过22.5万苹果账户
--2015年9月



[应用工具] 《3K抢红包王》

★★★★★ 6
2894 553



共享Appleid免费购买应用



```
if ( std::__1::basic_string<char,std::__1::char_traits<char>,std::__1::allocator<char>>::find(
    &v15,
    "itunesstored",
    0,
    12) != -1 )
{
    v18 = 3;
    std::__1::basic_string<char,std::__1::char_traits<char>,std::__1::allocator<char>>::init(
        &v15,
        "/System/Library/Frameworks/Security.framework/Security",
        54);
    v18 = 4;
    std::__1::basic_string<char,std::__1::char_traits<char>,std::__1::allocator<char>>::init(&v14, "SSLWrite", 8);
    v18 = 5;
    v11 = getAddress(&v15, &v14);
    v18 = 6;
    std::__1::basic_string<char,std::__1::char_traits<char>,std::__1::allocator<char>>::basic_string(&v14);
    v18 = 7;
    std::__1::basic_string<char,std::__1::char_traits<char>,std::__1::allocator<char>>::basic_string(&v15);
    if ( v11 )
    {
        v18 = 8;
        NSHookFunction(v11, replace_SSLWrite, &original_SSLWrite);
    }
}
```



获取appleid账号密码



```
query_label = objc_msgSend(_query, "objectForKey:", CFSTR("lab1"));
if ( _query_label )
{
    v69 = -1;
    NSLog(CFSTR("==lab1:%08"));
    v69 = -1;
    if ( !objc_msgSend(_query_label, "compare:", CFSTR("com.apple.lockdown.identity.activation")) )
    {
        *(DWORD *)_result = 0;
    }
}
```





IOS安全实际案例

30

国内丢失iphone你可能会出现以下这种情况：

----销赃的利益链

---- 2015年12月

iphone丢失 → 补办卡/登录邮箱



一个月内成功骗到一万三千多个iCloud账号和密码



追索

流入二手市场销售

手机解除Appleid绑定





- 1: Appleid账户密码很重要, 决定了你手机何时报废、包括你的ipad、macbook
- 2: 只从AppStore上下载应用, 勿信任任何企业证书
- 3: 勿越狱IOS设备、酌情更新漏洞补丁

谢谢大家

THANK YOU FOR YOUR ATTENTION

www.antiy.com

 安天 | 智者安天下