2015 年全国大学生信息安全竞赛

作品简介

作品名称：向 Android 系统内部数据传递过程的敏感信息保护系统

电子邮箱：1242750990@qq.com

提交日期：2015 年 6 月 7 日

填写说明

1. 所有参赛项目必须为一个基本完整的设计。参赛作品简介旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。

2. 参赛作品简介采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。

3. 参赛作品简介不超过6页A4纸。

4. 参赛作品简介模板里已经列的内容仅供参考，作者也可以多加内容。

5. 为保证网评的公平、公正，作品简介中应避免出现作者所在学校、院系和指导教师等泄露身份的信息。

目录

一. 概述

用户敏感数据在 Android 系统内部传递过程中极易被恶意软件窃取与篡改的安全威胁目前尚无相关研究，这给用户的隐私和财产安全无疑带来了巨大的威胁。

针对此类威胁，本作品创新性的展开研究，给出了实用性较强的解决方案和原型系统。作品首先详细分析了用户敏感数据在 Android 系统内部产生、传递和使用的整个流程：为了实现不同应用程序、不同进程间的资源共享、数据传输，Android 系统提供了基于 Binder 的进程间通信的机制。敏感数据在系统内以 Binder 为通道进行传输。但是敏感数据在 Binder 传输过程中是以明文形式进行的，这就使得恶意软件能够轻易地截获和篡改基于 Binder 通信的敏感数据，如短信内容、GPS 位置信息等。针对这一问题，在上述研究的基础上，**本作品创新性地提出并实现了针对 Android 系统中敏感数据从产生到使用整个生命周期的自适应性透明加密的保护方案**，有效防止了敏感数据被恶意第三方窃取与篡改的威胁，**保障了用户敏感数据在系统内部传输过程中的私密性及完整性**。此外，本系统还向用户提供了简单、灵活的操作体验，使得用户可独立自主地对指定应用程序中特定类型的敏感数据进行保护，增强了本系统的易用性和实用性。

本作品的成功完成，不仅创新性的为 Android 系统中基于 Binder 的敏感数据传递提供了一种可选的防护方法；同时也为企业、政府等安全性要求较高的机构构建了一种强制性的安全增强型策略，促进了 Android 系统在企业中的应用，推动了 Android 生态系统的健康良性的发展，具有较大意义。

二. 设计与实现

为了灵活地保护敏感信息在Android系统内部传递过程，本小组设计了一款应用软件，其核心思想是在敏感信息产生的地方进行加密，只允许拥有解密权限的应用解密，并且让用户能够自由选择要保护的敏感信息类型以及选择授予应用软件权限，以保证用户隐私。

2.1 基本模块

本作品程序负责监控Android手机的基本安全情况，根据用户输入开启相应的保护模块，并对整个系统的保护状态进行反馈与评估。整个系统分为四个模块：**数据呈现模块，基本信息获取模块，命令接收模块，敏感数据保护模块**。其中，敏感数据保护模块又分为**敏感数据加密模块，敏感数据解密模块**两个子模块。

2.2 业务流程

1) 程序启动，获取手机基本状态并显示当前保护信息。
2) 等待用户输入。
3) 若已经开启保护模式，判断当前需要保护的功能模块。
4) 当需要保护的功能模块确定后，判断需要保护的 APP。
5) 判断成功后，对手机的相应程序功能实施保护方案并反馈保护结果。

2.3 相关技术

经过对Android系统各层次能够完成的功能进行评估发现，整个系统的功能实现均能在JAVA层实现。在整个系统中，数据的保护是通过API Hook劫持方法调用实现的，其本质是对目标方法进行重定向，在执行目标方法前进行一些额外的操作。虽然Android并没有提供与Hook操作相关的方法，但是通过Xposed框架能共完成相同的功能。其中关键技术有：对相关函数的拦截技术采用API Hook技术；数据加密解密采用SMS4加解密算法；攻击测试方案中对Binder IPC数据拦截采用注入和HOOK技术。

2.4 理论研究

2.4.1Binder 机制研究

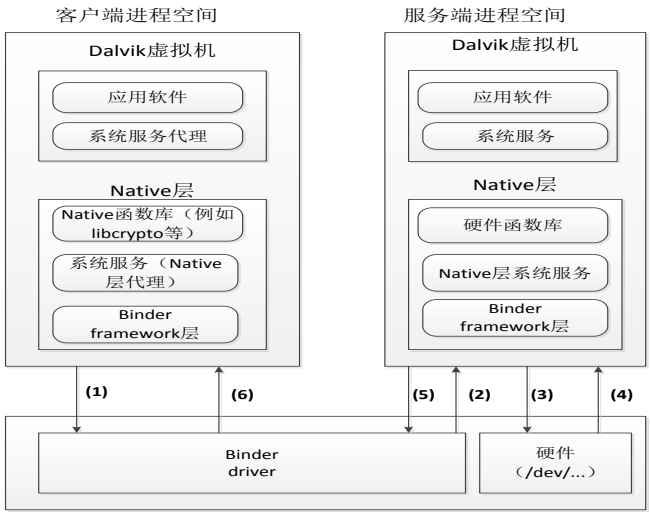

图 2-1 Binder 机制

从图 2-1 可以清晰地知道 Binder 机制的流程如下：

1) binder 框架层创建一个 ioctl syscall， 其中有文件描述作为参数， 以及相关数据传入内核 。
2) binder driver 寻找对应的 服务，把数据复制到 server 的空间，创立线程等待处理 。
3) server 响应服务，向相关 硬件提交请求 。
4) 硬件回应请求 。
5) 将回复传递给 driver。
6) 将回复传递给客户端进程。

2.4.2API Hook 研究

- 基于Xposed框架的HOOK方式

Xposed 框架是一款 Android 系统辅助工具，可以在不修改 APK 的情况下影响程序运行（修改系统）的框架服务，基于它可以制作出许多功能强大的模块，完成 Android 系统权

限管理、内存清理、电池控制、界面修改显示等功能，并且能够互不冲突地运作。

- 基于ptrace系统调用的SO库注入的HOOK方式

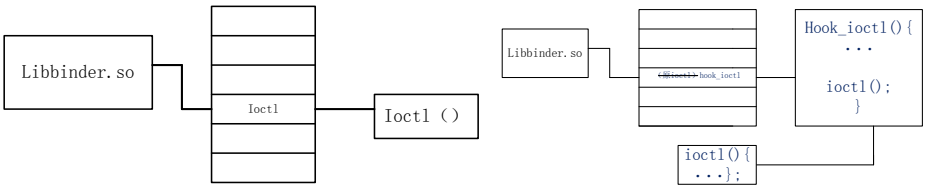

图2-2 ptrace系统调用的so库注入的Hook方式

从上图，可以知道调用流程是，Libbinder.so通过其GOT表中的Ioctl地址调用Ioctl方法，被Hook之后，GOT表中填入新的Ioctl地址，Libbinder.so会在ioctl()执行前执行hooked_ioctl方法。

2.5 短信方案设计

对于Android操作系统中，关于短信应用程序，从软件的功能角度来讲，短信分为**对话列表，消息列表，短信编辑，彩信编辑，短信显示**和**配置**。从实现的角度来看，它分为**GUI展示层，发送/接收，信息数据等**，这些分类对应着源码中的各种包。在Android操作系统中，与短信相关的源码放在com.android.mms包中。在系统发送端：需要重点关注的类是WorkingMessage，该类中封装了代表短信内容的属性变量和相关的处理方法，可以认为是Framework层上短信产生的源头；在系统接收端：需要重点关注的类是SmsSingleRecipientSender和SmsMessageSender，SmsSingleRecipientSender是短信发送的最后一站。综上因此，对于Android系统的应用来说，可以会利用该类来实现短信的发送，要想在Server端进行拦截，应该考虑HOOK该类本身或者与该类有关的类即可实现短信敏感信息保护。

2.6 GPS 方案设计

在 Android 系统中，系统通过位置提供者（LocationProvider）的回调方法，当 LocationProvider 接收到位置信息的变化时就会调用 LocationMangerService 的 reportLocation()，这个方法会促使 Server 端更新位置信息，并将其存放在一个 HashMap 中。所以我们通过该类进行 hook 保护即可达到 GPS 位置敏感信息的保护。

三. 测试与分析

3.1 防护功能测试

-GPS防护功能测试

1) 测试方法

使用手机 A 运行本作品，开启 GPS 保护模块，显示保护效果并通过 Log 信息来反馈

HOOK 结果。

2) 测试目的

测试程序是否能完成注入，实现预期的 GPS 保护方案，从而保障了用户敏感数据在系统内部传输过程中的私密性及完整性。

3) 预期结果

程序成功 HOOK 了预期方法，完成了对 GPS 敏感数据在系统内部传输过程中的加密解密功能。另外，在开启 GPS 保护后，用户无需进行额外的操作即可正常使用 GPS 服务。

-短信防护功能测试：

1) 测试方法

使用手机 A 运行本作品，开启短信保护模块，发送短信给 B 手机，显示发送效果并通过 log 信息反馈 HOOK 结果。

使用手机 A 运行本作品，开启短信保护模块，从 B 接受短信，显示接受效果并通过 Log 信息反馈 HOOK 结果。

2) 测试目的

测试程序是否能完成注入，实现预期的短信保护方案，从而保障了用户敏感数据在系统内部传输过程中的私密性及完整性。

3) 预期结果

程序成功 HOOK 了预期方法，完成了对短信敏感数据在系统内部传输过程中的加密解密功能。另外，在开启短信保护后，用户无需进行额外的操作即可正常使用短信服务。

3.2 基本功能测试

-应用管理功能测试

一、 测试方法

使用手机 A 安装本程序，点击按钮进入应用管理功能，尝试将程序添加/删除信任，并获取系统基本信息。

二、 测试目的

测试程序能否正确实现应用管理功能，引导用户完成防护方案的基本配置。

三、 预期结果

程序能实现预期的应用管理功能。包括显示，筛选系统程序。显示程序基本信息，以及将程序添加或剔除信任。

-日志记录功能测试

一、 测试方法

使用手机 A 安装本程序，点击按钮进入日志记录功能，查看当前的保护记录。

二、 测试目的

测试程序是否能够正确实现日志记录功能，记录用户的历史保护方案和基本保护信息。

三、 预期结果

程序能实现预期的日志记录功能，包括记录用户的历史保护方案和基本保护信息。

-计划任务功能测试

一、 测试方法

使用手机 A 安装本程序，点击按钮进入计划任务功能，设置系统定时计划。并查看相应日志记录。

二、 测试目的

测试程序是否能够正确实现计划任务功能。

三、 预期结果

程序能实现预期的计划任务功能，包括设定定时保护的功能。

-安全配置功能测试

一、 测试方法

使用手机 A 安装本程序，点击按钮进入安全配置功能，设置系统定时计划。并查看相应日志记录。

二、 测试目的

测试程序是否能够争取实现安全配置功能，管理当前的防护方案。

三、 预期结果

程序能实现预期的计划任务功能，包括管理当前已防护程序。

四. 应用前景分析

用户敏感数据在Android系统内部传递过程中极易被恶意软件窃取与篡改的安全威胁目前尚无相关研究，这无疑带来了巨大的威胁。本作品目的就是解决这一难题，在应用方面十分广阔。不言而喻，用户隐私敏感数据对于每一个用户都至关重要，尤其是对Android市场占有率84.6%以上的手持移动智能操作系统，惠及的可以说是全球84.6%的用户，不可不说是多么广阔；同时从长远来看，安卓系统的进一步强化安全，可以使Android进入高安全要求性的部门与单位的应用领域，比如大中型企业和政府部门，进一步完善Anroid在各个领域的生态安全体系，应用前景现在，未来都十分广阔。

五. 创新点总结

本作品为面向Android系统内部数据传递过程的敏感信息保护软件，并且此软件以用户为主，根据用户的选择为用户敏感数据提供透明化的精确的保护，在立题初衷，实现技术，应用前景方面具有重大的创新性，同时在防护加固，应用范围，实际操作等方面具有重大的实用性。

本作品的创新性具体表现在以下几个方面：

1. 立题初衷

近年来 PC 转向移动化，智能化道路，从而迎来移动与智能的春天。移动智能手持设备已成为现今人们生活，交流，工作不可缺少的一部分。然而，移动智能操作系统开始处于一方独大的局面，Android 系统 2014 年第二季度全球市场份额已达 84.6%，并且不断上升。由于漏洞与操作系统本身关联极大，多种操作系统并存的一个好处是，一个操作系统的漏洞影响范围仅仅局限该操作系统。而 Android 的巨大市场占有率本身就预示着巨大的安全隐患，而今一个 Android 的漏洞波及范围是全球 85%以上用户，可想而知，对于 Android 系统的加

固与保护意义的重大性。Binder 的明文传输可以说是 Android 系统存在的一个薄弱点，前瞻性做出实际工作，防患于未然，保护 Binder，实现透明化加密解密，意义非凡。

2. 实现技术

保护 Binder 里面的敏感数据，从技术层次方面涉及的不是上层建筑，而是底层支持。在底层方面 Android 系统提出的基于 Binder 的 IPC 机制实现十分复杂，要实现这方面的保护，难度可想而知，也是十艰巨的。保护 Binder 里面的敏感数据，并实现加解密其中必要条件就是在深刻领悟源码的基础上融合加解密 SMS4 方面的知识。本小组团结合作，分工明确，在老师指导下，明确系统目标，各自攻读 Android 底层源码，同时经常交流，沟通，最终实现成果，在技术方面实现可以说是对系统层次安全性的加固，有很大的技术创新性。

3. 应用前景

Android 系统在市场上的高占有率，使得每一个针对 Android 系统的安全创新拥有广大的应用与市场，其中需要指出的是对企业，政府等安全性要求很高的部门，对安全性的追求更是永无止尽的。基于 Binder 的 IPC 机制下的敏感数据的保护在应用方面有很大的需求，本作品在技术实现的基础上，面向用户进行了界面包装美化，操作也简单易懂，不仅为广大用户提供了一种强有力保护敏感数据的解决方案，同时也为企业、政府等安全性要求较高的构建了一种新的安全增强型策略，可以促进 Android 系统在企业中的应用，推动了 Android 生态系统的健康良性的发展，具有应用创新意义。

本作品的实用性具体表现在以下几个方面：

1. 防护加固

信息时代，信息安全至关重要，尤其是手机中的大量敏感信息，因此对 Android 的加固防护必须引起十分的重视。Binder 机制中敏感数据的明文传输是一个亟待解决的安全问题。本课题组实现了针对 Binder 的加解密的解决方案，不仅有效增强了用户敏感数据的保护，而且在此基础上，增加了用户自主开启，关闭，配置，定制以及日志纪录等操作与功能，成功打造了有一个有效的，时刻保护用户敏感数据的生态体系，防护加固实用性十分强。

2. 应用范围

绝大多数的敏感数据是基于 Binder 机制的，应用的上层建筑——各种 App 尽管可能千奇百样，种类繁多，但是映射到底层，我们只需要专注于 Binder 的安全保护，在底层方面实现该机制的保护，不仅是在系统级打造了一扇安全门，截断了绝大多数的上层 App 随意获取敏感数据的通道，而且不言而喻应用范围也是极其广阔的。

3. 实际操作

本课题组在实现 Binder 机制敏感数据加解密的核心技术之上并不满足，在面向用户，尤其是普通用户，实际操作与体验实用性方面也是本课题组关注的一个地方，本课题组细致考虑，在 UI 方面尽可能友好，不仅仅提供实时动态的圆圈指示保护状态，还设身处地提供开启、关闭定制、日志记录，个性配置等操作功能，在实际操作方面十分实用。

六. 未来工作

在以 Android 手机为代表的移动智能终端迅猛发展的同时，针智能移动终端的恶意软件也层出不穷、花样繁多，这给保障用户隐私、财产等安全带来了极大挑战。攻击和防护作为“矛”和“盾”，则是一个长期博弈的过程，很难在一个较短的时间周期内战胜另一方。

受利益的趋势和信息安全中短板效应的决定，攻击者攻击系统的手段肯定会更加高明，不排除将来不法分子能够在 IPC 传输流程上找到一个比我们更加优先的攻击点，使得我们的安全防护失效。

除此之外，本次作品实现的防护方案还较为单一，无法满足各类用户的功能需求。

针对以上的问题，今后将继续开展如下工作：

1) 继续加强对 Android 系统的理解，尽量深入到系统的 Native C 层和内核层，彻底实现从敏感数据产生到使用整个生命周期的不完善。从最本质的部分学习 IPC 的工作过程，并研究攻击者可能的攻击入口，提供相应的保护方案并集合到我们已有的系统之上。
2) 完善程序的功能模块，提供各种行之有效的服务保护功能，争取针对用户满足其各类不同的要求。