

Android 系统隐藏技术及检测方法

平 程,蔡皖东

(西北工业大学 计算机学院,陕西 西安 710129)

摘 要:Android 木马通过获取系统 root 权限,修改内核表项实现隐藏功能,进而躲避木马查杀软件的检测。因此研究 Android 系统隐藏技术对于发现隐藏木马、提高查杀软件的检测能力有重要意义。文中在传统 Linux 系统隐藏技术的基础上,对 Android 系统服务启动过程进行分析,探究出适用于 Android 系统的隐藏方法,并实现了一种 Android Rootkit 木马原型,用于测试现有木马检测软件对该类型木马的检测能力。文中提出了针对此类 Rootkit 型木马的检测方法,实验证明这些方法对检测此类木马有一定的作用。

关键词:Android 系统;隐藏技术;Rootkit;检测

中图分类号:TP316

文献标识码:A

文章编号:1673-629X(2014)05-0142-04

doi:10.3969/j.issn.1673-629X.2014.05.034

Hidden Technology and Method of Detection in Android System

PING Cheng, CAI Wan-dong

(College of Computer, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract: Getting the access to root privileges, the Android Trojan not only modifies important tables in the kernel to hide, but also leaves away from the detection of anti-virus software. So it's very important to discover hidden technology in Android system for finding hidden Trojan and improving detection capability. Based on the traditional hidden technology in Linux system, analyze the Android system service startup process, explore the hidden technology in Android system, and realize a prototype of Android Rootkit Trojan to test the detection capability of existing software in related to this type of Trojan. A detection technology is presented which focuses on finding Rootkit Trojan and the experiment shows the method plays a certain role in detecting.

Key words: Android system; concealing; Rootkit; detection

0 引 言

Google 公司于 2007 年 11 月正式发布基于 Linux 内核的开源手机操作系统 Android。经过五年多的发展,基于 Android 系统的手机日益普及,同时 Android 也被越来越多的重要行业所应用^[1]。Android 平台因其开源的特性带来不断发展创新的同时,也成为众多恶意软件的攻击目标。恶意程序利用 Android 平台未公布或未修复的安全漏洞,植入恶意代码,并开启后门监听程序来实现其恶意的目的,伺机窃取用户的隐私信息或者强制扣费等^[2]。因此,研究安卓系统隐藏隐蔽技术,对于改进木马防护软件的性能,提高其检测查杀能力有重要意义,而且有助于研究移动平台网络安全的防护。

文中针对 Android 系统,研究其安全机制和 Android 木马的隐藏隐蔽技术,设计并实现一种简单的

Android Rootkit 木马原型,通过修改内核表项实现服务和文件隐藏,并提出预防此类恶意程序的方法,为 Android 系统安全防护提供技术支持。

1 相关知识

1.1 Android 系统架构

Android 作为一个移动设备的平台,其软件层次结构自下而上分成四层:操作系统层、各种库和 Android 运行环境、应用程序框架、应用程序。其中操作系统层使用 Linux 内核,内核的主要任务是负责与计算机硬件进行交互,实现对引进的编程控制和接口操作,并调度对硬件资源的访问。此外,内核还为用户应用程序提供一个高级的执行环境和访问硬件的虚拟接口^[3]。

1.2 Android 安全机制

Android 是一个权限分离的系统,在创建初始就有

收稿日期:2013-07-15

修回日期:2013-10-19

网络出版时间:2014-02-11

基金项目:陕西省科学技术研究发展计划项目(2013K06-19)

作者简介:平 程(1988-),男,硕士研究生,研究方向为网络信息安全;蔡皖东,教授,博士生导师,研究方向为网络信息安全。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140211.1613.031.html>

很好的安全机制,通过继承和创新,对系统架构各个层次的安全性都进行有针对性的增强设计。Android 通过继承 Linux2.6 内核的安全机制实现系统安全,又通过创新设计出权限机制实现数据安全,在理论上有一定的安全防护能力^[4]。

1.3 Android 服务启动过程

Android 系统在启动时首先会启动 Linux 基础系统,然后引导加载 Linux Kernel 并启动初始化进程 Init,接着启动 Linux 守护进程 Daemons,与此同时,还需要启动 Zygote 进程。再接着,需要初始化 Runtime 进程,过程包括:初始化服务管理器和注册服务管理器。Runtime 进程初始化不久,将发送一个请求到 Zygote,开始启动系统服务,这时 Zygote 将为系统服务进程建立一个虚拟机实例,并启动系统服务^[5]。接下来,系统服务将启动原生系统服务,主要包括 Surface Flinger 和 Audio Flinger,这些本地服务将注册到服务管理器作为 IPC 服务的目标。紧接着,系统服务将启动 Android 管理服务,最后当系统加载完所有的服务之后会处于等待状态,等待程序运行,如图 1 所示^[6]。

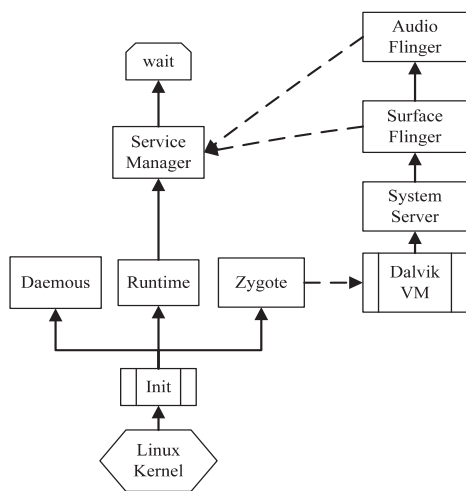


图 1 Android 系统初始化流程

2 传统 Linux 隐藏技术

2.1 模块编程技术

为了使内核保持较小的体积并能够方便的进行功能扩展,Linux 系统提供了模块机制。由于模块在插入后是作为 Linux 内核的一部分来运行的,所以模块编程实际上就是内核编程,因此可以在模块中使用一些由内核导出的资源,这样就可以根据该地址直接修改系统调用的入口,从而改变系统调用。

2.2 映像修改技术

/dev/kmem 是一个字符设备,是计算机主存的映像,通过它可以测试甚至修改系统,当内核不导出 sys_call_table 地址或者不允许插入模块时可以通过该映像修改系统调用,从而实现隐藏文件、进程或者模块的

目的^[7]。

2.3 虚拟文件技术

proc 文件系统是一个虚拟的文件系统,它通过文件系统的接口实现,用于输出系统运行状态。Linux 中不存在直接查询进程信息的系统调用,类似于 ps 这样查询进程信息的命令是通过查询 proc 文件系统来实现的,通过访问并修改 proc 文件,可以达到隐藏某特定服务的目的。

2.4 小结

Linux 隐藏技术有很多,大体上的思路是:通过各种方法找到系统服务调用表(sys_call_table)的地址,修改相应的 sys_call_table 表项,使其指向新的系统调用,调用完毕再恢复原来的系统调用,这样木马隐藏运行,不易被发现。Android 系统基于 Linux 内核,所以 Linux 系统的隐藏技术通过改进处理能够应用于 Android 系统。

3 Android 系统隐藏技术的实现

3.1 简单隐藏实现

针对一些对 Android 移动设备不熟悉的用户,只在 launcher 应用程序一栏中不显示。此方法比较简单,只要把应用程序项目中 androidManifest.xml 文件中<category android:name="android.intent.category.LAUNCHER"/>去掉即可实现。但是在 setting 里面的应用程序浏览中仍可以找到,隐蔽得不彻底。市场上一些主动隐藏的软件,例如网秦的“私密空间”,可以隐藏某些信息,但是同样,在 setting 里面的应用程序浏览中仍可以发现此应用在运行。这种主动隐藏的方法实际上是利用权限实现的隐藏,不具备权限和口令的非权限用户无法访问隐藏空间,但是较简单,root 和 system 用户容易查询其踪迹,一般不被木马采用。

3.2 Rootkit 技术实现

Rootkit 是能够持久或可靠地、无法检测地存在于计算机上的一组程序和代码^[8]。Rootkit 所采用的大部分技术和技巧都用于在系统上隐藏代码和数据,Android 系统木马利用 Rootkit 技术隐藏程序本身、进程、文件和网络通信等,隐蔽性极高,尤其对于内存小不适宜安装大型杀毒软件的移动 Android 系统更加难以检测。下面是 Rootkit 在 Android 系统下的隐藏技术实现过程。

(1) 破坏内核。

利用可加载模块的方式将代码植入内核,一旦拥有了在内核中运行的代码,就能够完全访问内核和系统进程的全部特权内存空间^[9]。通过内核级访问,可以修改系统上所有软件代码和数据结构。

(2) 修改系统重要参数。

① 获取 `sys_call_table` 偏移地址:Android 从早期版本就是使用了 Linux 2.6 以上的内核版本,获取 `sys_call_table` 偏移地址与在 Windows、Linux 系统上有很大不同,目前已研究的方法有主观 `system.map` 获取和通过 `swi` 中断来动态搜索系统调用表。但对于目前各种 Android 移动设备来说,平台已固化到芯片上,动态搜索更安全可靠^[10]。获取 `sys_call_table` 地址即可加入代码(挂钩技术)调用系统原始进程查询函数并过滤查询结果,从而隐藏木马进程。

② 修改中断描述符表 IDT:利用钩子技术检测或阻止任何使用系统调用进程,主要用来躲避一些杀毒软件和检测软件的检测,使其为己所用,蒙蔽检测者。

③ 修改内存中文件操控函数:对于文件隐藏,原理同进程隐藏类似,修改内存中文件操控函数,在返回结果之前加入过滤函数,隐藏掉相关文件。

④ 网络相关信息欺骗:为了避免木马暴露,采用最小脚印策略,伪装 TCP/IP 协议和隐写术等技术都可被 Rootkit 所采用。例如 `enyelkm` 木马将木马所在系统设置成客户端而非服务器,使其外出的数据包穿过防火墙而不被检测,达到网络隐藏的目的。

3.3 木马原型实例

在传统 Linux 隐藏技术和 Rootkit 技术的基础上,分析 Android 系统服务启动过程,设计了一种 Android Rootkit 型木马原型 `Mykit`。`Mykit` 基于 Linux LKM 的原理,通过可加载模块的方式进驻内核,利用 Hook 技术劫持系统调用。其技术难点有:如何获得系统调用表(`sys_call_table`)的位置^[11],如何在 Android 的源码下进行编译以及如何调试这些代码。获取 `sys_call_table` 的方式采用 `swi` 中断动态搜索的方式。

`Mykit` 最主要的目的是隐藏自身的踪迹,主要是通过文件隐藏和进程隐藏等方式,文件隐藏的原理图如图 2 所示。

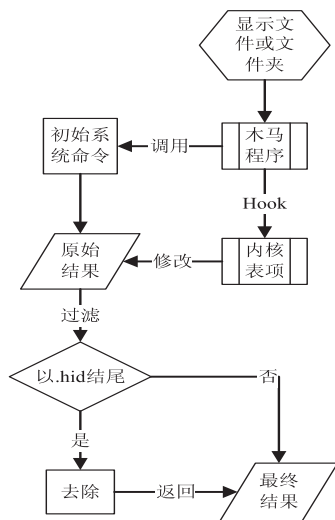


图 2 `Mykit` 文件隐藏原理图

木马将要隐藏的文件(或文件夹)命名以 `.hid` 结尾,在显示文件或文件夹时,先调用原来的系统调用,对得到的结果进行过滤去除与特定文件名相关的文件信息,从而实现文件隐藏。在 Linux 中不存在直接查询进程信息的系统调用,类似于 `ps` 这样查询进程信息的命令是通过查询 `proc` 文件系统来实现的,所以进程隐藏类似于文件隐藏。

木马控制系统窃取系统信息的关键是获取 `root` 权限,攻击者一旦窃取最高权限会进一步修改系统,所以避免木马第一次入侵至关重要。研究 Android 木马原型实例可以验证现有木马检测软件的检测能力,对改进检测算法有一定的促进意义。

4 Android Rootkit 检测方法

4.1 基于现有检测程序的检测

Android Rootkit 的攻击者和防御者之间自 Android 系统问世以来一直“战争”不断,先加载到内核中运行的一方占有巨大优势。众所周知 Rootkit 难以检测且较难清除,它们在内核中运行,拥有 `root` 权限,但存在即有痕迹,只要它们能运行起来,就一定存在内在调用机制,要进行信息窃取也必定留下“脚印”。优秀的 Rootkit 检测软件能够抓住这些“蛛丝马迹”,发现 Rootkit 恶意程序的存在。遗憾的是目前没有针对 Android 系统 Rootkit 木马的完整检测软件和清除工具,大部分停留在研究阶段,无相关下载,只能利用一些手机安全软件例如 `LBE 安全大师` 进行综合检测^[12]。

4.2 基于占有机制的检测

Rootkit 木马不同于其他普通类型的木马和病毒,它对于一个系统的占用有“先入为主”的特点,若系统中已经安装了一个 Rootkit,攻击者的程序在安装己方恶意程序时可能会因为出错而停止。这种系统服从单一类型木马或病毒控制的机制称为木马或病毒的占有机制。基于占有机制,可以设计守护 Rootkit 程序,植入安全监测模块到内核,但不破坏内核,针对恶意 Rootkit 可能改动的内核位置,设置改动标识,并通过内核模式与用户模式交互,反馈结果给用户。这种“自我控制”的方式虽然在一定程度上,可以减少其他 Rootkit 木马的入侵,但由于技术复杂性和系统差异性,并不适合普通用户开发。

4.3 基于扫描数据库的检测

Android 木马开机自启动需要获取 `android.permission.RECEIVE_BOOT_COMPLETED`,访问网络需要获取 `android.permission.INTERNET`,可见一般木马要进行恶意行为必须获取相应权限^[13]。所以当应用程序安装时,可以通过核查其功能与权限的对应程度,判断其是否为恶意木马。建立合法程序功能权限扫描数据

库,在程序安装之前进行安全检查;建立已知木马MD5样本扫描库,安装之前进行匹配排除。该方法需要建立大型数据库,并获取合法软件的相关信息,类似于云扫描。此种方法的缺陷是无法检测未知的木马,但却是目前较通用的方法。

对于不断发展的 Android Rootkit 技术来说,任何检测算法都会有纰漏。用户需要做的是规范获取合法代码程序的途径,增强维护信息安全的意识。生产厂商应提高硬件安全技术,切不可为了谋私利植入恶意木马。

5 测试及结果分析

5.1 实验环境

实验是在 Linux 的操作系统上安装 Android 驱动开发环境,在模拟器上运行试验,具体如表 1 所示。

表 1 测试环境说明

条目	要求
虚拟机	VMware Workstation6.5, 安装有内核 2.6.32 的 CentOS
Eclipse	JDK1.6 及以上版本,带有 ADT
Android	带有源码,Android SDK2.2 及以上
其他	eclipse 创建的模拟器安装安卓版本及 360 杀毒软件

5.2 检测方法分析

将 Mykit 编译加载运行在 Android 模拟器上,360 病毒查杀软件没有提示发现可疑木马,在模拟器 setting 里面的应用程序浏览中未发现 Mykit 踪迹,且系统其他功能仍工作正常。

目前 Linux 系统下 Rootkit 的检测工具有 Rootkit Hunter、chkrootkit 等,而 Android 系统下尚无针对 Rootkit 型木马的检测工具,故无法测试。将 Mykit 的程序源码名字及内容做一些改变,编译成可加载模块 His-kit.ko,在已安装 Mykit 的前提下,用 insmod 命令加载时出错,说明占有机制测试可行。基于扫描数据库的检测方法由于软件权限和庞大系统的特点实验下无法测试。表 2 是三种方法的比较。

表 2 三种检测方法比较

方法	应用范围	对系统影响	开发难度	实验测试
检测程序	单机	一般	一般	360 检测
占有机制	测试	较大	较小	可行
扫描数据库	大型系统	一般	困难	无

6 结束语

文中以传统 Linux 隐藏技术为基础,分析了 Android 服务启动过程,剖析出 Android 木马隐藏机制,并编程实现了一种 Android Rootkit 木马原型,此类木马获取系统 root 权限,修改 sys_call_table 等内核重要表项,隐蔽性高,难以检测。对 Android 木马隐藏技术的实现和对检测方法的测试,有助于加深对 Android Rootkit 型木马的认识,提高对 Android 手机全民安全防护意识,促进此类恶意程序检测工具的开发进程,维护用户信息安全。

参考文献:

[1] 符易阳,周丹平. Android 安全机制分析[C]//第 26 次全国计算机安全学术交流会. 出版地不详:出版者不详,2011: 23-25.

[2] 王 玮. 基于 Android 系统的恶意程序原理分析[J]. 信息网络安全,2012(10):71-76.

[3] 杨丰盛. Android 技术内幕·系统卷[M]. 北京:机械工业出版社,2011:5-9.

[4] King S T,Chen P M. SubVirt:Implementing malware with virtual machines[C]//Proc of IEEE symposium on security and privacy. Berkeley:[s. n.],2006:327-341.

[5] Shabtai A,Kanovov U,Elovici Y. Intrusion detection for mobile devices using the knowledge based,temporal abstraction method[J]. Journal of Systems and Software,2010,83(8): 1524-1537.

[6] 李 骏,陈小玉. Android 驱动开发与移植实战详解[M]. 北京:人民邮电出版社,2012:10-20.

[7] 袁 源,戴冠中. LKM 后门综述[J]. 计算机科学,2008,35(7):5-8.

[8] Hoglund G,Butler J. ROOTKITS-Windows 内核的安全防护[M]. 北京:清华大学出版社,2007:19-31.

[9] 李文新,王姜博,慕德俊,等. Android 系统 Rootkit 技术综述[J]. 微处理机,2011,32(2):68-72.

[10] 董 蕾,黄淑华,尹浩然,等. 基于 Android 平台的手机木马关键技术分析[J]. 信息网络安全,2012(11):63-65.

[11] 韩诗源,尹浩然,杨 晶. Android 系统 Rootkit 原理[J]. 电子制作,2012(10):98-99.


[12] Enck W,OngTang M,McDaniel P. Understanding Android security[J]. IEEE Security and Privacy,2009,7(1):50-57.

[13] 宋 杰,党李成,郭振朝,等. Android OS 手机平台的安全机制分析和应用研究[J]. 计算机技术与发展,2010,20(6):152-155.

Android系统隐藏技术及检测方法

作者：平程， 蔡皖东， PING Cheng, CAI Wan-dong

作者单位：西北工业大学 计算机学院, 陕西 西安, 710129

刊名：计算机技术与发展

英文刊名：Computer Technology and Development

年，卷(期)：2014(5)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjfz201405034.aspx